Mohammed, Thaha; Naas, Si-Ahmed; Sigg, Stephan; Francesco, Mario Di

Knowledge Sharing in AI Services: A Market-based Approach

Published in:
IEEE Internet of Things Journal

DOI:
10.1109/JIOT.2022.3206585

Published: 15/01/2023

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Mohammed, T., Naas, S.-A., Sigg, S., & Francesco, M. D. (2023). Knowledge Sharing in AI Services: A Market-based Approach. *IEEE Internet of Things Journal*, *10*(2), 1320-1331. Article 9891736.
https://doi.org/10.1109/JIOT.2022.3206585

Knowledge Sharing in AI Services: A Market-based Approach

Thaha Mohammed, Si-Ahmed Naas, Stephan Sigg, and Mario Di Francesco

***Abstract*—Today's deep neural networks (DNNs) are very accurate when trained on a large amount of data. However, suitable input might not be available or may require extensive data collection. Data sharing is one option to address these issues, but it is generally impractical because of privacy concerns or due to the problematic process of finding a sharing agreement. Instead, this work considers knowledge sharing by first exchanging the weights of pre-trained DNNs and then applying transfer learning. Specifically, it addresses the economics of knowledge sharing in AI services by taking a market-based approach. In detail, a model based on Fisher's market is devised for optimal knowledge sharing, defined as the gain in inference accuracy from exchanging DNN weights. The proposed approach is shown to reach a market equilibrium and to satisfy important economic properties, including Pareto optimality. A technique for weight fusion is also introduced to merge acquired knowledge with the existing one. Finally, an extensive evaluation is conducted in a distributed intelligence scenario. The obtained results show that the proposed solution is efficient and that weight fusion with transfer learning significantly increases inference accuracy compared to the original DNN, without the overhead of federated learning.**

***Index Terms*—Knowledge sharing, artificial intelligence, smart services, weight trading, Fisher market, game theory.**

I. Introduction

Machine learning has been increasingly adopted in a large number of applications leveraging Artificial Intelligence (AI) to impact people's daily life [1]. In fact, more and more providers of *AI-based services* include deep neural networks (DNN) as building blocks to realize intelligent applications. One example is represented by object detection, that can be applied to a variety of use cases, ranging from smart factory automation to autonomous driving [2].

Today's DNNs are effective in terms of accuracy, especially when trained on a large amount of data. This generally involves using large-scale datasets that are publicly available, as suitable input might not otherwise be accessible and (or) would require extensive data collection [3]. Unfortunately, these public datasets were primarily conceived as benchmarks, therefore they may not be appropriate for application-specific scenarios. One option to overcome these issues is *data sharing*: owners of valuable domain-specific datasets share their data with other parties – which can then train their own DNNs with these data – in exchange for a monetary compensation, for instance [4]. Despite its simplicity, data sharing is often impractical due to multiple reasons, including privacy concerns related to the source data [5] or the problematic process of finding an agreement between different parties [6].

A different alternative is *knowledge sharing*: rather than the sheer data, the knowledge built on these data is shared instead [7]. In practice, the weights of a pre-trained DNN constitute such knowledge. In this context, *transfer learning* is an effective technique to utilize the weights of a given DNN with a different one [8]. Accordingly, a model that has previously been trained on a similar task – namely, a base DNN – is used as a starting point or as a feature extractor for a target DNN. During training, a subset of the old layers is frozen and does not change. The remaining layers are then re-trained on a different task with a new training dataset [9]. A different approach is *federated learning*, in which different devices train a local model that is continuously shared with others through a central coordinator to gain global knowledge [10].

Knowledge sharing is convenient as it does not need access to the source data. It is also particularly beneficial when multiple AI-based service providers (SPs) are involved. In fact, the knowledge from DNNs trained for specific use cases could as well be employed in other applications. For instance, a smart manufacturing service may run different types of DNNs for object recognition, which might also be applied to assisted driving and video surveillance (Fig. 1a). However, this kind of knowledge sharing requires suitable economic incentives, otherwise SPs might not be willing to participate [11] – for instance, because of the effort already put in collecting data and training their own DNNs. The existing literature on DNNs has extensively addressed data sharing – especially in terms of partitioning [12, 13] – in addition to federated learning [14]. However, the economic aspects of knowledge sharing in the context of AI-based services have not been adequately studied. This work specifically addresses this gap by considering the economics of knowledge sharing in AI services through a market-based approach. Accordingly, it considers multiple service providers trading weights of DNNs pre-trained to specific use cases. These weights are acquired by buyers to increase their inference accuracy by means of transfer learning. This work is one of the first to address economic aspects of knowledge sharing in AI-based applications offered by multiple SPs.

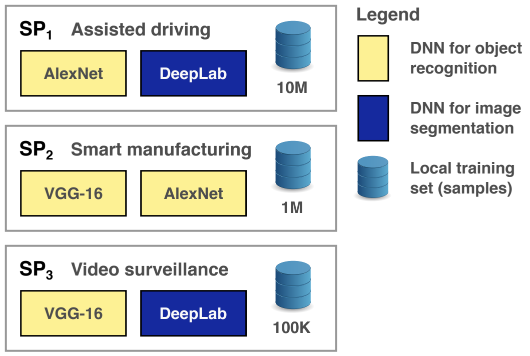
In detail, this work establishes the following contributions.

- It **introduces a novel game-theoretic scheme for knowledge sharing based on DNN weights**. In particular, it devises a model based on Fisher's market [15] for optimal knowledge sharing, defined as the gain in inference accuracy obtained by different SPs.

This work was partially supported by the Academy of Finland under grants number 326346 and 332307, and by the European Commission through the REDI project (Marie Skłodowska-Curie grant number 101034328).

T. Mohammed and M. Di Francesco are with the Department of Computer Science, Aalto University, Espoo, Finland (e-mail: {thaha.mohammed, mario.di.francesco}@aalto.fi). S.-A. Naas and S. Sigg are with the Department of Communications and Networking, Aalto University, Espoo, Finland (e-mail: {si-ahmed.naas,stephan.sigg}@aalto.fi).

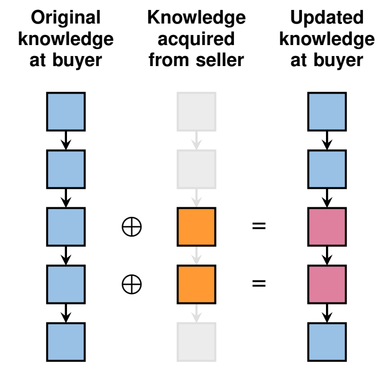
 DOI: 10.1109/JIOT.2022.3206585



(a)

SP	DNN type	Selling price ($)	Buying price ($)
1	AlexNet	15	0
1	DeepLab	8	1
2	VGG-16	5	2
2	AlexNet	4	7
3	VGG-16	0	10
3	DeepLab	3	9

(b)



(c)

Fig. 1: (a) Reference scenario: multiple providers offer AI-based services, relying on different DNN architectures. (b) Service providers sell and buy the weights associated with the DNNs they use as a form of market-based knowledge sharing based on transfer learning. The actual price depends on an estimate about the benefit in buying or selling the related weights, for instance, due to the size of the local training set. (c) Acquired weights are used by buyers to update (part of) the knowledge in their DNNs so as to increase their inference accuracy.

- A **weight market mechanism is devised to trade and fuse knowledge of different SPs**. The mechanism satisfies several important economic properties: Pareto optimality, envy freeness, sharing incentives, and proportional allocation.
- An **extensive evaluation is conducted in a distributed intelligence scenario**, in terms of both economic properties and the impact of weight fusion on learning. The obtained results show that the proposed solution is efficient in clearing the market and that weight fusion with transfer learning significantly increases inference accuracy, without the overhead of federated learning.

The rest of the article is organized as follows. Section II introduces the reference scenario and the considered market model, then formulates the problem of optimal knowledge sharing. Section III describes a weight market mechanism and characterizes its economic properties. Section IV evaluates the proposed scheme in terms of the resulting market equilibrium and prediction accuracy against the state of the art. Section V overviews the related work. Finally, Section VI provides a summary as well as directions for future research.

II. A Weight Market for Knowledge Sharing

This section introduces the reference scenario and the main features related to the considered economic model (Fig. 1), in addition to the formulation of the optimal knowledge sharing problem. Table I summarizes the key notation used throughout the article.

A. Reference Scenario

The reference scenario is a network offering different AI-based services to end users (Fig. 1a). These services involve diverse applications, requiring different types of inference that is customized to a specific use case. For instance, a service supporting intelligent transportation systems includes different AI-based components such as text recognition of license plates, object detection for collision avoidance, and forecasting of traffic conditions to find time-efficient routes. Each of these components is a deep neural network (DNN) that has been trained for the specific use case. However, the knowledge of a certain DNN could also help other services relying on the same functions. As an example, object detection for intelligent transportation could also be leveraged in a smart construction site. Therefore, the existing knowledge on object detection could be employed in the other use case too.

In this context, the concept of *transfer learning* has emerged as an efficient alternative to training DNN models from scratch [8]. Accordingly, a DNN is first pre-trained on a large dataset and then the related knowledge (i.e., the weights of that DNN) is leveraged to customize a different DNN. This work specifically addresses knowledge sharing through transfer learning, wherein multiple service providers (SPs) offer the weights of their DNNs for sale to other SPs. More formally, the SPs are denoted by the set $\mathcal{S}$ and divided into buyers $s{\in}\mathcal{S}_B$ and sellers $\hat{s}{\in}\mathcal{S}_S$, such that $\mathcal{S}{=}\mathcal{S}_B \cup \mathcal{S}_S$. Each seller offers the weights of R different DNN architectures for sale, where $r \in \mathcal{R}{=}\{1, 2, \dots, R\}$ is the actual type of DNN, for instance, AlexNet, ResNet32, or VGG16 [16]. Transfer learning is coordinated by an orchestrator, operating as an inference serving system [9]. In particular, the orchestrator collects information from SPs about their DNN architectures as well as the related features before mediating the process involving buyers and sellers. Specifically, the orchestrator determines the prices of the weights by targeting a market equilibrium, as detailed next.

B. Market Model

The process of knowledge sharing is characterized as a Fisher's market[1] [15] wherein goods (i.e., resources) are the weights of a specific DNN architecture, sold and bought by SPs (Fig. 1b). Weights have a certain demand that, in turn, determines their price. Moreover, buyers have a certain budget for the weights as well as an indication of the benefit obtained by acquiring these weights. Such a benefit is described in terms of a *utility function*. Buyers are rational, namely, they

[1]The key idea behind such an approach is to clear the market to maximize knowledge sharing. In contrast, other options such as auctions may result in monopoly or oligopoly, therefore consolidating knowledge into one or a few providers instead of sharing it across all of them [17].

TABLE I
SUMMARY OF KEY NOTATION.

Symbol	Description
$\mathcal{S}=\mathcal{S}_B \cup \mathcal{S}_S$	Set of SPs (buyers and sellers, respectively)
$s, \hat{s}$	A given buyer and seller, respectively
$\mathcal{R}, r$	Set of DNN types and a given DNN type, respectively
$\boldsymbol{w}_r$	Vector denoting weights of different layers
$c_{\hat{s}}^r$	The layer count for DNN of type r at $\hat{s}$
$\boldsymbol{c}_{\hat{s}}^r$	Capacity vector denoting $c_{\hat{s}}^r$ for DNN of type r
$a_s^{\hat{s},r}$	Weights for DNN type r allocated to SP s from SP $\hat{s}$
$\boldsymbol{a}_s^{\hat{s}}$	Allocation vector denoting $a_s^{\hat{s},r}$ for DNN of type r
$d_s^{\hat{s},r}$	Demand of SP s of DNN type r from SP $\hat{s}$
$\boldsymbol{d}_s^{\hat{s}}$	Demand vector denoting $d_s^{\hat{s},r}$ for DNN of type r
$\boldsymbol{A}_s$	Matrix of weights received by a buyer s from all sellers
D_s	Maximum demand of service s
$\mathrm{u}_s(\boldsymbol{A}_s)$	Utility of the allocation $\boldsymbol{A}_s$ for buyer s
B_s	Budget of each buyer s
$p^{\hat{s},r}$	Price for the DNN weights of type r at SP $\hat{s}$
$\boldsymbol{p}_{\hat{s}}$	Price vector denoting $p^{\hat{s},r}$ for DNN of type r
$\tilde{\boldsymbol{p}}$	Vector denoting equilibrium prices for weights
$\tilde{\boldsymbol{A}}$	Vector denoting equilibrium allocation of weights
$b_s^{\hat{s}}$	Price of demand vector for buyer s from seller $\hat{s}$

aim at maximizing their utility, which expresses the gain in the accuracy from using the acquired weights (Fig. 1c). Sales take place over certain time periods in rounds, i.e., sale can occur multiple times (for instance, periodically). The roles of the SPs may differ at each round: an SP may act as a buyer in one round and as a seller in another one. However, an SP is either buying or selling weights within a single round of sale. Moreover, weights sold to one seller in a round cannot be sold to another seller in the same round. Still, the same weights can be resold to different buyers in separate rounds.

Weights of DNN type r are denoted by a vector $\boldsymbol{w}_r$, corresponding to the weights of the different layers therein according to a certain order[2]. The weights are considered divisible resources, in the sense that SPs may trade a subset of them: one or more layers, or portions thereof. Consequently, the size of a given DNN architecture is expressed in terms of its number of layers. Following the standard terminology in the literature [18], such a maximum size is called *capacity*. Accordingly, the capacity of the weights owned by an SP $\hat{s}$ is given by the vector $\boldsymbol{c}_{\hat{s}}^r=(c_{\hat{s}}^1, c_{\hat{s}}^2, \dots, c_{\hat{s}}^R)$, where $c_{\hat{s}}^r$ is the maximum number of layers of a given weight type $r\in\mathcal{R}$ owned by SP $\hat{s}$. The base demand of a given class of weights for the buyer is defined as the minimum amount of these weights to improve the accuracy derived from the local knowledge only. Specifically, the base demand of SP s for the weights of $\hat{s}$ is indicated as the vector $\boldsymbol{d}_s^{\hat{s}}=\{d_s^{\hat{s},1}, d_s^{\hat{s},2}, \dots, d_s^{\hat{s},R}\}$ where $d_s^{\hat{s},r}>0, \forall\hat{s}\in\mathcal{S}_S, r\in\mathcal{R}, s\in\mathcal{S}_B$ is the demand for weight of type r for SP s. The base demand vector depends on the actual weights of the sellers, which may vary across different SPs. Moreover, such a vector is determined based on the expected accuracy[3] from acquiring the weights, which has been shown to be predictable [21, 22]. Each buying SP s is allocated weights from selling SP $\hat{s}$, indicated through the vector $\boldsymbol{a}_s^{\hat{s}}=(a_s^{\hat{s},1}, a_s^{\hat{s},2}, \dots, a_s^{\hat{s},R})$, where $a_s^{\hat{s},r}$ represents the amount of weights for DNN type r allocated to SP s from SP $\hat{s}$. An allocation $\boldsymbol{a}_s^{\hat{s}}$ is feasible if the seller $\hat{s}$ has the weights of the DNN type (r) required by the buyer. The weights a buyer receives from multiple sellers are called a weight *bundle*. In particular, the weight bundle the SP s receives from all the other selling SPs is described by the matrix, $\boldsymbol{A}_s=(\boldsymbol{a}_s^1, \boldsymbol{a}_s^2, \dots \boldsymbol{a}_s^{\hat{s}}, \boldsymbol{a}_s^{|\mathcal{S}_S|})$, where $\boldsymbol{A}_s \in \mathbb{R}^{|\mathcal{S}_S|\times R}$. In other words, the rows of the matrix represent the allocation vector from each SP $\hat{s}$.

Recall that the different weights of DNN type r belong to a given service, thus, they should be considered as a whole. For instance, these weights may be associated with two distinct DNNs – one for object recognition and another for image segmentation – as for the assisted driving and video surveillance scenarios illustrated in Fig. 1a. As a consequence, certain weights might become a bottleneck by affecting the accuracy of the service as a whole. Such an issue is explicitly accounted for in the allocation carried out by the broker, as detailed next.

Given an allocation $\boldsymbol{a}_s^{\hat{s}}$ for an SP s, the minimum gain in accuracy $x_s^{\hat{s}}$ of s from the weights of $\hat{s}$ among all r is:

$$\mathrm{x}_s^{\hat{s}}(\boldsymbol{a}_s^{\hat{s}}) = \min\left\{\frac{a_s^{\hat{s},1}}{d_s^{\hat{s},1}}, \frac{a_s^{\hat{s},2}}{, d_s^{\hat{s},2}}, \dots, \frac{a_s^{\hat{s},R}}{d_s^{\hat{s},R}}\right\} = \min_r \frac{a_s^{\hat{s},r}}{d_s^{\hat{s},r}}, \forall\hat{s}, s \tag{1}$$

implying that there should not be any knowledge transfer that reduces accuracy.

Moreover, given a weight bundle $\boldsymbol{a}_s^{\hat{s}}$, the total weight gain x_s, which SP s can obtain from all sellers $\hat{s}$ with a finite demand is the function:

$$\mathrm{x}_s(\boldsymbol{a}_s^{\hat{s}}) = \min\left\{\sum_{\hat{s}\in\mathcal{S}_S} \min_{r\in\mathcal{R}} \frac{a_s^{\hat{s},r}}{d_s^{\hat{s},r}}, D_s\right\}, \forall s \tag{2}$$

where D_s is the maximum demand of service s. Here, the summation across the sellers denotes the minimum weight gain that can be obtained from all sellers for a given buyer s.

Finally, the utility of services $\mathrm{u}_s(\boldsymbol{A}_s)$ is a function describing the buyer's appraisal for a bundle, i.e., $\mathrm{u}_s(\boldsymbol{A}_s) : \mathbb{R}^{|S_S|\times|R|} \to \mathbb{R}$. In particular, the utility is such that $\mathrm{u}_s(\boldsymbol{A}_s) = \mathrm{x}_s(\boldsymbol{a}_s^{\hat{s}})$, namely:

$$\mathrm{u}_s(\boldsymbol{A}_s) = \mathrm{x}_s(\boldsymbol{a}_s^{\hat{s}}) = \min\left\{\sum_{\hat{s}\in\mathcal{S}_S} \min_{r\in\mathcal{R}} \frac{a_s^{\hat{s},r}}{d_s^{\hat{s},r}}, D_s\right\}, \forall s \tag{3}$$

A higher utility indicates a larger number of weight requests accommodated at a bottleneck, thus, a higher knowledge transfer for a given service in the network.

C. Problem Formulation

The considered trading scenario can be formulated as a utility maximization problem in a Fisher's market where

[2] Using a "linear" representation of the weights allows to characterize DNNs with different structures such as, for instance, those considered in Section IV.

[3] Section IV will show how to derive such an accuracy for the case of distributed intelligence based on combined local / global knowledge [19, 20].

some of the SPs are the buyers, some of the SPs are the sellers, and the weights are the goods in a given round. The prices of the weights are expressed through the vector $\boldsymbol{p}_{\hat{s}} = (p^{\hat{s},1}, p^{\hat{s},2}, \ldots, p^{\hat{s},r}, p^{\hat{s},R})$, wherein the price $p^{\hat{s},r}$ refers to the weights for DNN type r at SP $\hat{s}$. The objective of the market is to determine the optimal price and allocation of the weights to the buyers. The corresponding problem can be formulated as follows.

Problem 1 (Optimal Knowledge Sharing Problem)**.** The optimal resource allocation for SP s at price p maximizes the utility of the services as:

$$\max_{\boldsymbol{A}_s} \quad \mathrm{u}_s(\boldsymbol{A}_s) \tag{4}$$

$$\text{s.t} \quad \sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}} a_s^{\hat{s},r} p^{\hat{s},r} \le B_s \tag{4a}$$

$$a_s^{\hat{s},r} \ge 0, \ \forall \hat{s}\in\mathcal{S}_S, r\in\mathcal{R} \tag{4b}$$

where the constraint in Eq. (4a) signifies that the allocation should be within the budget B_s of each service s, while the constraint in Eq. (4b) indicates that the allocation consists of a positive amount of weights.

It is worth noting that the capacity constraints on goods do not affect the prices of the resources. Therefore, the problem can be reformulated by explicitly writing the utility from Eq. (3) as:

$$\max_{\boldsymbol{A}_s, u_s} \quad u_s \tag{5}$$

$$\text{s.t} \quad u_s = \min\left\{\sum_{\hat{s}\in\mathcal{S}_S} u_s^{\hat{s}}, D_s\right\}, \forall s \tag{5a}$$

$$u_s^{\hat{s}} = \min_{r\in\mathcal{R}} \frac{a_s^{\hat{s},r}}{d_s^{\hat{s},r}}, \forall \hat{s} \tag{5b}$$

$$\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}} a_s^{\hat{s},r} p^{\hat{s},r} \le B_s \tag{5c}$$

$$a_s^{\hat{s},r} \ge 0, \ \forall \hat{s}\in\mathcal{S}_S, r\in\mathcal{R} \tag{5d}$$

The goal is then to find a market equilibrium as detailed next.

III. Weight Allocation and Update

This section first addresses how to reach a market equilibrium, which solves the problem introduced earlier through an optimal allocation of weights to buyers. A weight market mechanism for trading and fusing knowledge is then proposed, followed by an evaluation of the economic properties achieved at the market equilibrium.

A. Market Equilibrium

Solving Problem 1 corresponds to finding a solution that maximizes the utility of each buyer s subject to budget constraints and clears[4] the market. Such an outcome is denoted as the tuple $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{A}})$, where the vector $\tilde{\boldsymbol{p}} = (\tilde{\boldsymbol{p}}_1, \ldots, \tilde{\boldsymbol{p}}_{\hat{s}})$ indicates the equilibrium prices and $\tilde{\boldsymbol{A}} = (\tilde{\boldsymbol{A}}_1, \ldots, \tilde{\boldsymbol{A}}_s)$ represents an equilibrium allocation of weights. The corresponding condition is called *market equilibrium* [15, 23].

[4]Market clearance implies that the traded supply is equal to the demand.

Definition 1 (Market equilibrium)**.** *A solution $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{A}})$, is a market equilibrium if and only if the following conditions are met.*

1) ***Utility maximization**: for all buying SP $s \in \mathcal{S}_B$, receives the weight bundle $\tilde{\boldsymbol{A}}_s$ that maximizes the utility u_s, given a non-negative price vector $\tilde{\boldsymbol{p}}$ under budget constraint. This implies that $\tilde{\boldsymbol{A}}_s$ is an optimal solution to Problem 1 with $\boldsymbol{p} = \tilde{\boldsymbol{p}}$.*
2) ***Market clearance**: each weight is either completely sold or has zero price, i.e., $\left(\sum_s \tilde{a}_s^{\hat{s},r} - c_{\hat{s}}^r\right)\tilde{p}^{\hat{s},r}=0$ and $\sum_s \tilde{a}_s^{\hat{s},r} \le c_{\hat{s}}^r$, $\forall s, \hat{s}, r$.*

The zero pricing policy enables unsold weights to be allocated to SPs without violating the budget constraints. Note that such an assignment does not affect the utility of any SP, as $\tilde{\boldsymbol{A}}_s$ is already the best bundle among all affordable ones. Hence, the market clears with the above-mentioned conditions. For simplicity and following a common practice in the relevant literature [18], the capacities of the weights ($c_{\hat{s}}^r$ in $\boldsymbol{c}_{\hat{s}}^r$) are normalized in the range $[0, 1]$ and the prices (as well as demands) are scaled accordingly in the rest of the article.

The following establishes a correspondence between the market equilibrium and an equivalent formulation of the knowledge sharing problem (i.e., Problem 1) that can be solved via convex optimization [23, Chapter 5 and 6]. Specifically, Problem 1 can be rewritten into the *Extended Eisenberg-Gale Program* [24].

Problem 2 (Extended Eisenberg-Gale Program)**.** The extended Eisenberg-Gale program is defined as:

$$\max_{\boldsymbol{A}, u} \quad \sum_{s\in\mathcal{S}_B} B_s \ln \sum_{\hat{s}\in\mathcal{S}_S} u_s^{\hat{s}} \tag{6a}$$

$$\text{s.t} \quad u_s^{\hat{s}} \cdot d_s^{\hat{s},r} = a_s^{\hat{s},r}, \ \forall s, \hat{s}, r \tag{6b}$$

$$\sum_{\hat{s}\in\mathcal{S}_S} u_s^{\hat{s}} \le D_s, \forall \hat{s} \in \mathcal{S}_S \tag{6c}$$

$$\sum_{s} a_s^{\hat{s},r} \le 1, \ \forall r \in \mathcal{R}, \hat{s} \in \mathcal{S}_S \tag{6d}$$

$$a_s^{\hat{s},r} \ge 0, \ \forall s, \hat{s}, r \tag{6e}$$

The *extended* Eisenberg-Gale (extended EG or EEG) program is so called as it is a more general version of the *standard* EG program: it allows to enforce that the utility of buyers does not exceed the maximum demand D_s while still resulting in a Pareto-optimal, non-wasteful, and economic market. This is accomplished by the constraints in Eq. (6b) and (6c), which are not part of the standard EG program.

The actual correspondence between a solution of Problem 2 and the market equilibrium in Definition 1 is characterized next by leveraging the convex nature of the EEG program.

Proposition 1. *An optimal solution to Problem 2 is an exact non-wasteful and economic market equilibrium for a Fisher's market where buyers either spend all their budget or reach their utility limit.*

Proof. Assume that an optimal solution exists. Moreover, let $\lambda_s^{\hat{s},r}$, μ_s, $p^{\hat{s},r}$, and $\gamma_s^{\hat{s},r}$ be the dual variables associated with the constraints in Eqs. (6b)–(6e), respectively. From the Karush-Kuhn-Tucker (KKT) conditions, it is:

$$\forall s, \hat{s}: \qquad \frac{B_s}{u_s} - \sum_{r\in\mathcal{R}} \lambda_s^{\hat{s},r} \cdot d_s^{\hat{s},r} - \mu_s = 0 \tag{7}$$

$$\forall s, \hat{s}, r: \qquad \lambda_s^{\hat{s},r} \leq p^{\hat{s},r} \tag{8}$$

$$\forall s, \hat{s}, r: \qquad \text{if } a_s^{\hat{s},r} > 0 \Rightarrow \lambda_s^{\hat{s},r} = p^{\hat{s},r} \tag{9}$$

$$\forall \hat{s}, r: \qquad \text{if } \textstyle\sum_s a_s^{\hat{s},r} < 1 \Rightarrow p^{\hat{s},r} = 0 \tag{10}$$

$$\forall \hat{s}, r: \qquad \text{if } p^{\hat{s},r} > 0 \Rightarrow \textstyle\sum_s a_s^{\hat{s},r} = 1 \tag{11}$$

$$\forall \hat{s}, r: \qquad p^{\hat{s},r} \cdot (\textstyle\sum_s a_s^{\hat{s},r} - 1) = 0 \tag{12}$$

$$\forall s: \qquad \text{if } \mu_s > 0 \Rightarrow u_s = \textstyle\sum_{\hat{s}} u_s^{\hat{s}} = D_s \tag{13}$$

In detail, Eq. (7) expresses the relation between the dual variables (i.e., $\lambda_s^{\hat{s},r}$, μ_s) and the utility. Eqs. (8) and (9) indicate the connection between the price $p^{\hat{s},r}$ and the dual variable $\lambda_s^{\hat{s},r}$. Eq. (10) specifies that all unallocated weights have zero price. Eq. (11) signifies that all the weights with positive prices are completely allocated. Eq. (12) is the market clearance condition according to Definition 1.

Let the price of the demand vector $\boldsymbol{d}_s^{\hat{s}}$ for buyer s from seller $\hat{s}$ be $\boldsymbol{d}_s^{\hat{s}}$ be $b_s^{\hat{s}} = \sum_r p^{\hat{s},r} \cdot d_s^{\hat{s},r}$, $\forall s, \hat{s}$. Eq. (6b) implies that $a_s^{\hat{s},r} = 0$ if and if only if $d_s^{\hat{s},r} = 0$. Hence, $d_s^{\hat{s},r} > 0$ yields $\lambda_s^{\hat{s},r} \leq p^{\hat{s},r}$ as in Eq. (8). Through Eqs. (7) and (9), the following holds:

$$\forall s, \hat{s}: \qquad b_s^{\hat{s}} \geq \textstyle\sum_r \lambda_s^{\hat{s},r} \cdot d_s^{\hat{s},r} = (B_s/u_s) - \mu_s \tag{14}$$

$$\forall s, \hat{s}: \qquad \text{if } a_s^{\hat{s},r} > 0 \Rightarrow b_s^{\hat{s}} = (B_s/u_s) - \mu_s \tag{15}$$

Let now the minimum price of the demand vector for s be $b_s^{\min} = (B_s/u_s) - \mu_s$. Eqs. (14) and (15) establish that $b_s^{\hat{s}} \geq b_s^{\min}$ and if allocation $a_s^{\hat{s},r} > 0$ then $\hat{s}$ becomes $b_s^{\min}$. Consequently, the buyers always buy weights from the cheapest sellers.

The KKT conditions above also allow to write:

$$B_s - \mu_s u_s = u_s \textstyle\sum_r \lambda_s^{\hat{s},r} d_s^{\hat{s},r} = \sum_{\hat{s}} \sum_r p^{\hat{s},r} d_s^{\hat{s},r}$$

from which $\mu_s \cdot u_s = B_s - \sum_{\hat{s}} b_s^{\hat{s}}$ holds $\forall s$. Here, $\mu_s \cdot u_s$ is the remaining budget of a buyer s after buying the weights. By complementary slackness [25], if $u_s < D_s$ then $\mu_s = 0$, implying that buyer s spends the entire budget; when $\mu_s > 0$ [Eq. (13)] the utility is equal to the demand. Therefore, buyer s either spends all their budget at equilibrium or reaches their utility limit with surplus budget $\mu_s u_s$. As a consequence, the optimal solution to Problem 2 maximizes the utility of each buyer under budget constraints and results in a non-wasteful and economic market equilibrium. □

Proposition 1 has established that an optimal solution to the EEG program in Problem 2 is a market equilibrium. It remains to show that such a solution exists and is unique.

Proposition 2. *The equilibrium prices always exist and the utilities are unique for Problem 2.*

Proof. An equilibrium price exists if and only if the feasible region of the convex program is not empty [24]. According to Eq. (6e), an allocation $a_s^{\hat{s},r} = 0, \forall s, \hat{s}$ is also a feasible allocation. Hence, the equilibrium exists as the feasible region is not empty. Moreover, the objective function in Problem 2 has the form $\sum \hat{s} \log u_s$, thus, it is strictly concave. Therefore, there exists a unique utility vector u_s that maximizes the objective function. □

It has been now shown that solving the EEG program (i.e., Problem 2) achieves a market equilibrium. The last part in the derivation establishes the actual correspondence between the EEG program and the optimal knowledge sharing problem (i.e., Problem 1).

Proposition 3. *All non-wasteful and economic market equilibria captured by Problem 1 are an optimal solution of Problem 2.*

Proof. Consider $(\boldsymbol{p}, \boldsymbol{A})$ as a non-wasteful and economic market equilibrium captured by Problem 1. Since $\boldsymbol{A}$ is non-wasteful, no weight allocation exceeds the capacity, i.e., $\sum_s a_s^{\hat{s},r} \leq 1$. Moreover, the buyer is not allocated any additional weights not required by it, i.e, $\sum_{\hat{s}} \min_r (a_s^{\hat{s},r}/d_s^{\hat{s},r}) \leq D_s, \forall s$. Accordingly, the weights are allocated proportionally to the demands of the SP. Therefore, the utility can be defined as $u_s^{\hat{s}} = \min_r (a_s^{\hat{s},r}/d_s^{\hat{s},r}) = (a_s^{\hat{s},r}/d_s^{\hat{s},r}), \forall s, \hat{s}$. Hence, $\sum_{\hat{s}} u_s^{\hat{s}} \leq D_s$. Thus, $\boldsymbol{A}$ satisfies all constraints of Problem 2 in Eqs. (6b)–(6e).

The total amount spent by a buyer s for weights does not exceed the budget according to Eq. (4a). Moreover, the buyer obtains weights from the cheapest seller. Hence, the allocation $a_s^{\hat{s},r} > 0$ only takes place when $b_s^{\hat{s}} = b_s^{\min}$. The allocation $a_s^{\hat{s}}$ and the utility $u_s^{\hat{s}}$ are zero when $b_s^{\hat{s}} > b_s^{\min}$. Therefore,

$$B_s \geq \sum_{\hat{s}} u_s^{\hat{s}} b_s^{\hat{s}} = \sum_{\hat{s}} u_s^{\hat{s}} b_s^{\min} = u_s b_s^{\min} = \sum_{\hat{s}} \sum_r p^{\hat{s},r} a_s^{\hat{s},r}$$

Let $\mu_s = (B_s/u_s) - b_s^{\min}, \mu_s \geq 0$. It is $b_s^{\hat{s}} \geq b_s^{\min} = (B_s/u_s) - \mu_s, \forall s, \hat{s}$. Clearly, $b_s^{\hat{s}} = b_s^{\min} = (B_s/u_s) - \mu_s$ when $a_s^{\hat{s}} > 0$. Hence, the constraints in Eqs. (14)–(15) are satisfied. Consequently, the constraints in Eqs. (7)–(9) hold with an appropriate value of $\lambda_s^{\hat{s},r}$. As per Definition 1, the market clearing prices are either positive (when all weights at a seller are sold) or zero. This follows the KKT conditions for constraints in Eqs. (10)–(12). Furthermore, let us assume $u_s < D_s$ for a buyer s. The spent amount $\sum_{\hat{s}} \sum_r p^{\hat{s},r} a_s^{\hat{s},r}$ is within the budget when $\mu_s > 0$ as shown above. Assume now that another affordable bundle with allocation $\delta_s a_s$ where $\delta_s = (B_s / \sum_{\hat{s}} \sum_r b_s^{\hat{s}} a_s^{\hat{s},r}) > 1$ exists and improves the utility $u_s(\delta_s a_s)$ when bought. This implies $u_s(\delta_s a_s) > u_s(a_s)$. Thus, $\boldsymbol{A}$ does not satisfy the utility maximization condition under budget constraints in Definition 1, thereby contradicting the assumption that $\boldsymbol{A}$ is a market equilibrium. Therefore, $u_s < D_s$ when $\mu_s > 0$, which follows the condition in Eq. (13). As a consequence, $\boldsymbol{A}$ satisfies the all the KKT conditions in Eqs. (7)–(13) and the proposition holds. □

In summary, it is enough to solve Problem 2 to find an allocation of weights that maximizes the utility defined in Problem 1 and corresponds to the market equilibrium.

B. Trading and Weight Fusion

The knowledge sharing process as a whole is the weight market mechanism described in Algorithm 1. The proposed

Algorithm 1: Weight Market Mechanism

Input : A set of buying SPs $s \in \mathcal{S}_B$, selling SPs $\hat{s} \in \mathcal{S}_S$, broker, and weights for sale $\boldsymbol{w}_r$
Output : Equilibrium prices $\tilde{\boldsymbol{p}}$ and allocation $\tilde{\boldsymbol{A}}$
// Parameter derivation
1 **foreach** buyer s and DNN type r **do**
2 $\quad d_s^{\hat{s},r} \leftarrow$ Estimate demand through an API provided by the broker and model in [22] for DNN of type r
3 $\quad B_s \leftarrow$ Calculate the budget for based on $d_s^{\hat{s},r}$
4 $\quad$ Send $d_s^{\hat{s},r}$ and B_s to broker for sale
5 **foreach** seller $\hat{s}$ and DNN type r **do**
6 $\quad p^{\hat{s},r} \leftarrow$ Initial prices of weights for DNN of type r
7 $\quad$ Send $p^{\hat{s},r}$ to the broker for sale
// Market clearance
8 Compute the utilities from received data
9 $\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{A}} \leftarrow$ Compute the equilibrium prices and allocations for buyers and sellers ▷ Eqs. (6a) – (6e)
10 Send the prices and allocations back to all s and $\hat{s}$
11 All s pay $p^{\hat{s},r}$ to get $\tilde{a}_s^{\hat{s},r}$ from $\hat{s}$
// Weight fusion
12 $\boldsymbol{w_i} \leftarrow \tilde{\boldsymbol{a}}_s^{\hat{s}}$
13 Buyer s fuses $\boldsymbol{w_i}$ into current weight $\boldsymbol{w}$ as $\widetilde{\boldsymbol{w}}$ ▷ Eq. (16)

solution relies on a broker, which estimates the expected accuracy from acquiring weights to buyers and derives the weight allocation that clears the market. The algorithm itself consists of two main phases.

The first phase is the actual trading (lines 1–11) which involves buyers, sellers and a broker. The buyers derive their demand based on an estimate provided by the broker – through an application programming interface (API) – for the given DNN types (line 2), update their budget accordingly (line 3), and send both of them to the broker (line 4). At the same time, the sellers set the initial price for the weights of the given DNN types (line 6) and send them to the broker (line 7). Once all SPs have contacted it, the broker derives the actual utilities (line 8), solves Problem 2 (line 9) and sends the prices as well as the weight allocation back to the SPs (line 10). Buyers and sellers finally complete the sale (line 11).

The second phase of the mechanism involves updating the DNNs of the buyers based on the weights acquired from the sellers (lines 12–13). In this respect, let $\boldsymbol{w}$ be the local weights of a buyer and $\boldsymbol{w}_i$ the weights acquired from seller i, $\forall i$ such that $0 \leq i < n$. The new weights $\widetilde{\boldsymbol{w}}$ are then calculated as follows:

$$\widetilde{\boldsymbol{w}} = \boldsymbol{w} + \frac{1}{n} \sum_{i=1}^{n} \lambda_i \left(\boldsymbol{w_i} - \boldsymbol{w} \right) \tag{16}$$

where λ_i are tradeoff parameters such that $0 \leq \lambda_i \leq 1, \forall i \in \{1, \ldots, n\}$. Specifically, these parameters characterize the impact of the new weights on the prior knowledge. There are two special cases: $\lambda_i = 0$, which completely ignores the weights from seller i, by keeping the pre-existing knowledge as it is; and $\lambda_i = 1$, which entirely replaces the local knowledge with that of seller i.

C. Analysis of Economic Properties

The rest of the section characterizes several economic properties of the weight market mechanism at the market equilibrium.

- *Pareto optimality* (PO): individual SPs cannot improve their own condition without making that of any other SP worse.
- *Envy freeness* (EF): an SP should not prefer (i.e., envy) the allocation vector and prices of any other SP to its own, i.e., $\mathrm{u}_s(\boldsymbol{A}_s) \geq \mathrm{u}_{s'}(\boldsymbol{A}_{s'}), \forall s, s'$ and $\mathrm{u}_s(\boldsymbol{A}_s) \geq \mathrm{u}_{s'}(\frac{B_s}{B_{s'}}\boldsymbol{A}_{s'}), \forall s, s'$ for equal and unequal budgets, respectively.
- *Sharing incentive* (SI): each SP should have a utility higher than that corresponding to an allocation of the resources proportional to the budgets. Accordingly, $\mathrm{u}_s(\boldsymbol{A}_s) \geq \mathrm{u}_s(\hat{\boldsymbol{A}}_s), \forall s$ where $\hat{\boldsymbol{A}}_s$ is the proportional allocation.
- *Proportionality* (PR): the utility of every SP is proportional to its budget. Equivalently, an allocation $\boldsymbol{A}$ satisfies proportionality if $\mathrm{u}_s(\boldsymbol{A}_s) \geq \frac{B_s}{\sum_{s' \in \mathcal{S}_B} B_{s'}} \mathrm{u}_s(\boldsymbol{W}), \forall s, \boldsymbol{W} \in \mathbb{R}^{|\mathcal{S}_S| \times R}$, where $\mathrm{u}_s(\boldsymbol{W})$ is the maximum utility achieved by an SP s when it is allocated every single weight.

Recall that Algorithm 1 obtains a weight allocation by solving Problem 2. Therefore, it reaches market equilibrium with an optimal solution to the knowledge sharing problem (i.e., Problem 1). Each of the above-mentioned economic properties is proven next.

Lemma 4. *The market equilibrium obtained by Algorithm 1 is Pareto-optimal.*

Proof. Let $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{A}})$ be the optimal solution of Problem 2. Assume that there is another allocation $\boldsymbol{A}^*$ such that $\boldsymbol{A}^*$ is strictly preferred over $\tilde{\boldsymbol{A}}$. Therefore, $\mathrm{u}_s(\boldsymbol{A}_s^*) \geq \mathrm{u}_s(\tilde{\boldsymbol{A}}_s), \forall s$. Hence, for some $s' \in s$, the utility is strictly greater, i.e., $\mathrm{u}_s(\boldsymbol{A}_s^*) > \mathrm{u}_s(\tilde{\boldsymbol{A}}_s)$. If SP s' attains the utility limit with allocation $\tilde{\boldsymbol{A}}$, s' cannot strictly improve its utility. This implies that s' has not reached the utility limit. From the market clearing conditions, SP s' completely spends its budget with $\tilde{\boldsymbol{A}}$ at a market equilibrium. Moreover, every SP buys the weights from the cheapest sellers providing them. This implies that s' cannot improve its utility $\mathrm{u}_{s'}(\tilde{\boldsymbol{A}}_{s'})$ at the given prices $\tilde{\boldsymbol{p}}$, thereby establishing the Pareto optimality of $\tilde{\boldsymbol{A}}$. □

Theorem 5. *The market equilibrium obtained by Algorithm 1 is envy-free.*

Proof. Consider a market equilibrium with prices $\tilde{\boldsymbol{p}}$ and an allocation $\tilde{\boldsymbol{A}}$. At the prices $\tilde{\boldsymbol{p}}$, a buying SP s' can afford the weight bundle $\tilde{\boldsymbol{A}}_{s'}$ because $\tilde{\boldsymbol{A}}_{s'}$ is its optimal weight bundle at the market equilibrium. Hence, the total price paid by s' is less than its budget, i.e., $\sum_{\hat{s} \in \mathcal{S}_S} \sum_{r \in \mathcal{R}} \tilde{p}^{\hat{s},r} \tilde{a}_{s'}^{\hat{s},r} \leq B_{s'}$. This implies that $\sum_{\hat{s} \in \mathcal{S}_S} \sum_{r \in \mathcal{R}} \tilde{p}^{\hat{s},r} (\frac{B_s}{B_{s'}} \tilde{a}_{s'}^{\hat{s},r}) \leq B_s$. Hence, at the prices $\tilde{\boldsymbol{p}}$, the weight bundle $(\frac{B_s}{B_{s'}} \tilde{\boldsymbol{A}}_s)$ is under the budget constraint for service s. However, at the market equilibrium the preferred bundle of s is $\tilde{\boldsymbol{A}}_s$. Therefore, $\mathrm{u}_s(\tilde{\boldsymbol{A}}_s) \geq \mathrm{u}_s(\frac{B_s}{B_{s'}} \tilde{\boldsymbol{A}}_s)$ holds $\forall s, s'$. Thus, $\tilde{\boldsymbol{A}}$ and $\tilde{\boldsymbol{p}}$ are envy free. □

Theorem 6. *The market equilibrium obtained by Algorithm 1 satisfies the sharing incentive property.*

Proof. The budget constraints are satisfied by each buying SP at the market equilibrium, i.e., $\sum_{\hat{s} \in \mathcal{S}_S} \sum_{r \in \mathcal{R}} \tilde{p}^{\hat{s},r} \tilde{a}_s^{\hat{s},r} \leq B_s, \forall s$.

Therefore, the total expenditure of all the buying SPs is less than the cumulative budget of the SPs:

$$\sum_{s\in\mathcal{S}_B}\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\tilde{p}^{\hat{s},r}\tilde{a}_s^{\hat{s},r}=\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\tilde{p}^{\hat{s},r}\sum_{s\in\mathcal{S}_B}\tilde{a}_s^{\hat{s},r}\leq\sum_{s\in\mathcal{S}_B}B_s$$

According to the market clearing conditions, all unsold weights have zero prices, (i.e., if $\sum_{s\in\mathcal{S}_B}\tilde{a}_s^{\hat{s},r}<1$ then $p^{\hat{s},r}{=}0$) and all the weights with positive prices are allocated (i.e., if $p^{\hat{s},r}{>}0$ then $\sum_{s\in\mathcal{S}_B}\tilde{a}_s^{\hat{s},r}{=}1$). As a consequence, it is:

$$\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\tilde{p}^{\hat{s},r}=\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\tilde{p}^{\hat{s},r}\sum_{s\in\mathcal{S}_B}\tilde{a}_s^{\hat{s},r}\leq\sum_{s\in\mathcal{S}_B}B_s$$

Consider an allocation $\boldsymbol{A}^*$ where the buyers receive weights from the sellers proportionally to their budget (i.e., $a_s^{*\hat{s},r}=B_s/\sum_{s'\in\mathcal{S}_B}B_{s'},\forall s,\hat{s}$) with equilibrium prices $\tilde{\boldsymbol{p}}$. The cost of $\boldsymbol{A}^*$ for SP s is:

$$\begin{aligned}\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}a_s^{*\hat{s},r}\tilde{\boldsymbol{p}}_{\hat{s},r}&=\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\frac{B_s}{\sum\limits_{s\in\mathcal{S}_B}B_s}\tilde{p}^{\hat{s},r}\\&=\frac{B_s}{\sum\limits_{s\in\mathcal{S}_B}B_s}\sum_{\hat{s}\in\mathcal{S}_S}\sum_{r\in\mathcal{R}}\tilde{p}^{\hat{s},r}\\&\leq\frac{B_s}{\sum\limits_{s\in\mathcal{S}_B}B_s}\sum_{s\in\mathcal{S}_B}B_s\\&\leq B_s\end{aligned}$$

This implies that $\boldsymbol{A}^*$ is obtained at prices $\tilde{\boldsymbol{p}}$ by s within its budget. However, SP s always prefers the utility-maximizing weight bundle out of all the feasible allocations. Therefore, $\mathrm{u}_s(\tilde{\boldsymbol{A}}_s){\geq}\,\mathrm{u}_s(\boldsymbol{A}_s^*)$ which satisfies the sharing incentive property. □

Theorem 7. *The market equilibrium obtained by Algorithm 1 satisfies the proportional property.*

Proof. By definition of proportionality, it is:

$$\begin{aligned}\frac{B_s}{\sum\limits_{s'\in\mathcal{S}_B}B_{s'}}\mathrm{u}_s(\boldsymbol{W})&=\frac{B_s}{\sum\limits_{s'\in\mathcal{S}_B}B_{s'}}\min\left\{\sum_{\hat{s}\in\mathcal{S}_S}\min_r\frac{1}{d_s^{\hat{s},r}},D_s\right\}\\&=\min\left\{\sum_{\hat{s}\in\mathcal{S}_S}\min_r\frac{\frac{B_s}{\sum_{s'\in\mathcal{S}_B}B_{s'}}}{d_s^{\hat{s},r}},\frac{B_s}{\sum\limits_{s'\in\mathcal{S}_B}B_{s'}}D_s\right\}\end{aligned}\tag{17}$$

where $\boldsymbol{W}$ is the matrix denoting the weights for sale across all sellers. On the other hand, the sharing incentive property implies that:

$$\begin{aligned}u_s(\tilde{\boldsymbol{A}}_s)&\geq\mathrm{u}_s\left(\frac{B_s}{\sum\limits_{s'\in\mathcal{S}_B}B_{s'}}\boldsymbol{W}\right)\\&=\min\left\{\sum_{\hat{s}\in\mathcal{S}_S}\min_r\frac{\frac{B_s}{\sum_{s'\in\mathcal{S}_B}B_{s'}}}{d_s^{\hat{s},r}},D_s\right\}\\&\geq\min\left\{\sum_{\hat{s}\in\mathcal{S}_S}\min_r\frac{\frac{B_s}{\sum_{s'\in\mathcal{S}_B}B_{s'}}}{d_s^{\hat{s},r}},\frac{B_s}{\sum\limits_{s'\in\mathcal{S}_B}B_{s'}}D_s\right\}\end{aligned}\tag{18}$$

From Eqs. (17) and (18) it follows that $\mathrm{u}_s(\tilde{\boldsymbol{A}}_s)\geq B_s/\sum_{s'\in\mathcal{S}_B}B_{s'}\,\mathrm{u}_s(\boldsymbol{W})$. Hence, the market equilibrium satisfies the proportional property. □

Proposition 8. *Algorithm 1 finds an optimal solution to Problem 2 in polynomial time.*

Proof. Problem 2 is a convex program, therefore it can be solved by using interior-point methods such as the barrier method or the primal-dual method [26]. In both cases, the optimal solution is found in polynomial time as discussed in [25, 26]. In the barrier method, the desired number of centering steps required to reach an accuracy level of ϵ (from the optimal solution) is $\lceil\log\left(m/t^{(0)}\epsilon\right)/\log\mu\rceil$, where m is the number of inequality constraints, while μ and $t^{(0)}$ are constants. The polynomial time complexity of a barrier method (using Newton iterations) is $\mathcal{O}(\sqrt{m}\log(\frac{m/t^{(0)}}{\epsilon}))$ [25, 26]. □

An empirical evaluation of the running time will be discussed in Section IV-A.

IV. Evaluation

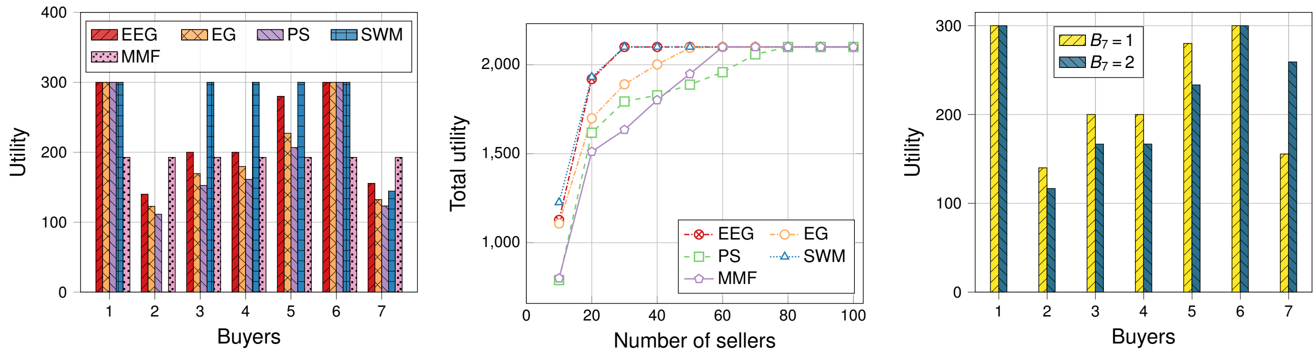
This section evaluates the proposed solution for knowledge sharing by extensive data-driven simulations. In particular, the market properties at the equilibrium are characterized first, followed by the impact of weight fusion on accuracy.

A. Market Equilibrium

The evaluation is carried out with a custom python simulator by considering the following settings. The reference scenario includes 14 sellers (i.e., $|\mathcal{S}_S|=14$) and 7 buyers (i.e., $|\mathcal{S}_B|=7$). All the SPs have the same budget (namely, $B_s=1,\forall s$) and maximum demand (utility) limit of 300 ($u_s^m=300$). Three well-known CNN architectures for object recognition are considered: AlexNet (with 12 layers), VGG-16 (with 16 layers), and NiN (with 8 layers). The Stanford Cars dataset [27] is considered for both training and inference. Computations were carried out on a machine equipped with an Intel i7-7700HQ CPU at 2.80 GHz, 32 GB of RAM, and an NVIDIA GTX 1050 Ti GPU.

In addition to the optimal solution to the EEG program (i.e., Problem 2), the following schemes are also considered.

- Eisenberg-Gale (EG), which allocates resources according to the standard EG program, namely, as the optimal solution of Problem 2 without the constraints in Eqs. (6b) and (6c).



(a) (b) (c)

Fig. 2: Utility of (a) individual buyers and (c) impact of budget changes in the EEG scheme; (b) total utility as a function of the number of sellers.

- Proportional sharing (PS) [28], which allocates a portion of every resource proportionally to the buyer's budget.
- Social welfare maximization (SWM) [29], which maximizes the total utility of all the buyers subject to the available supply, without considering the budget constraint.
- Max-min fairness (MMF), which maximizes the utility of the buyer with the lowest utility subject to the available supply.

The EG and EEG programs are solved by employing the CVXPY modeling language [30] and the MOSEK solver [31] after normalizing the capacity (i.e., the maximum number of layers) of the considered DNN architectures. The base demand vector is determined for a distributed intelligence scenario [19, 20], in which the orchestrator predicts the model accuracy expected from the weights by using a gradient boosting machine with regression trees [22].

Simulations are carried out by using the independent replication method, in which randomly selected subsets of the data are assigned to different SPs. The values reported in the figures represent the average from 25 iterations, along with the corresponding standard deviations as error bars, where noticeable. Two sets of experiments are performed: the first focuses on the achieved utility, while the other characterizes the economic properties of the different schemes.

1) Utility: Fig. 2a shows the utility of individual buyers for the different schemes. The highest values of utility are obviously obtained by SWM, however, the related allocations are very uneven: the utility of SP 7 is about the half of the highest, while SP 2 is not allocated any resources. The utility of EEG is equal to or higher than that of EG, as the latter cannot handle finite demand. The utility of EEG is higher than PR in all cases, and higher than MMF in most of the cases. It is worth noting that MMF provides a fair sale in terms of utility, but an unbalanced sale in terms of the allocated weights, as some buying SPs receive more weights than the required target to make up for the low utility. Hence, MMF results in sharing less knowledge to many of the involved buying SPs. Moreover, SWM achieves the utility limit for some of the buying SPs, whereas it obtains a poor utility for the remaining SPs. Consequently, less knowledge is shared to some SPs compared to the rest.

Furthermore, Fig. 2b shows the total utility of the buyers as a function of the number of sellers for the considered schemes. The earlier observations are also confirmed here. As the demand is limited, the total utility of the buyers saturates in all cases as the number of sellers increases, since the higher availability of weights for sale helps buyers reach their utility target. The figure highlights how EEG and SWM are faster in reaching the maximum value of about 2,100, with only a slight difference in favor of SWM when there are at most 20 sellers. The total utility obtained by EG is lower when there are between 20 and 40 sellers; PS and MMF perform worse, with the latter reaching the saturation point earlier. As previously discussed, MMF has the lowest total utility due to fair but unbalanced sale of weights, which results in a reduction of knowledge sharing among the SPs. SWM exhibits a total utility which is comparable to that of EEG; however, it causes a higher disparity due to unfair allocation. Instead, EEG obtains a total utility with a fair allocation that maximizes the utility of each individual buying SP. This result empirically confirms that the sharing incentive property in Theorem 6 holds.

Fig. 2c illustrates the impact of the buyer's budget on the utilities of sellers for EEG. For this purpose, the budget of all the first six SPs is set to one (as before), while the budget of the last SP is doubled (i.e., $B_{s_7} = 2$). The results show that the utility of SP 7 almost doubles, whereas that of the other SPs only slightly decreases. This demonstrates that, under the market equilibrium obtained with EEG, an SP can be prioritized by providing an extra budget without any significant impact on the utilities of the other SPs.

2) Economic Properties: This set of experiments characterizes the economic properties achieved at the market equilibrium. To do so, it considers the following metrics: the proportional fairness (PF) index, as the ratio between the original utility of a buyer and the maximum attainable utility (i.e., when all the weights are assigned to the SPs); the envy freeness (EF) index of a weight allocation $\boldsymbol{A}$, calculated as $\min_{s,s'} \frac{u_s(\boldsymbol{A}_s)}{u_s(B_s/B_{s'}\boldsymbol{A}_{s'})}, \forall s, s' \in \mathcal{S}_B$; and the computation time (the lower, the better) for the computational efficiency.

Fig. 3a shows the PF index of the EEG scheme as a function of the number of sellers for the individual buyers, as an

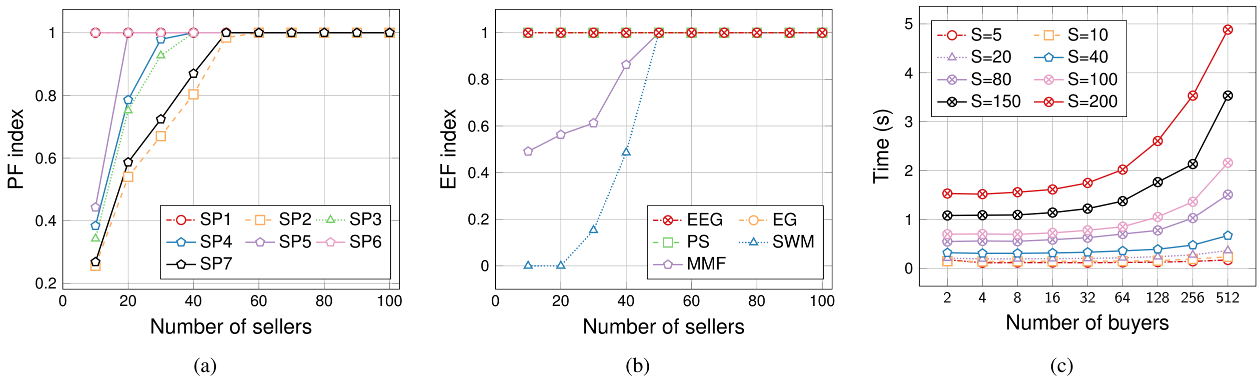


Fig. 3: (a) Proportional fairness index for EEG and (b) envy freeness index of the considered schemes as a function of the number of sellers. (c) Time needed to compute the market equilibrium for EEG as a function of the number of buyers, for different amounts of sellers.

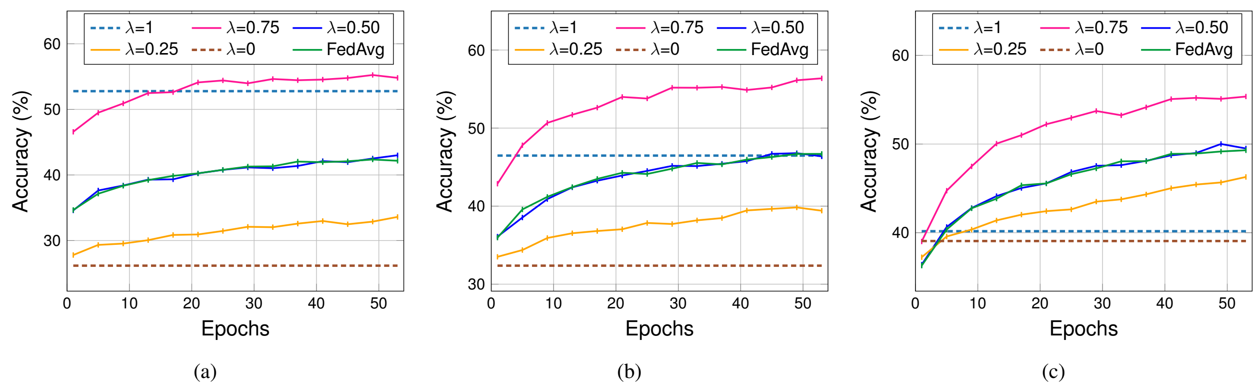


Fig. 4: Accuracy as a function of the training epoch for knowledge fusion involving one buyer and one seller for different distributions of training data between the buyer and the seller: (a) (30%, 70%), (b) (40%, 60%), and (c) (50%, 50%).

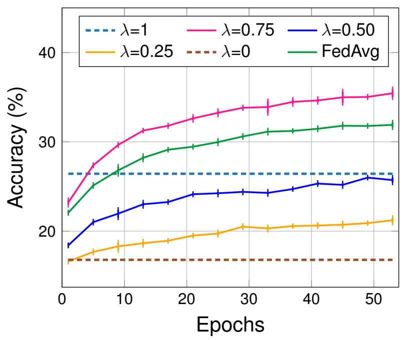
indication on how the proportionality property is fulfilled (the higher, the better). The PR property is satisfied by an allocation involving $|\mathcal{S}_B|$ buyers when the PF of each buyer is greater than or equal to $1/|\mathcal{S}_B|$; in the considered case, it should be greater than or equal to $1/7 = 0.1429$. Indeed, the PF index is always above 0.2559 even when there are only 10 sellers; it reaches the value of one for most buyers when there are at least 40 sellers. This confirms that the market equilibrium obtained through EEG satisfies the proportional property.

Fig. 3b illustrates the EF index as function of the number of sellers for the considered schemes, as an indication of how the envy-freeness property is fulfilled (the higher the better, as a high value of EF indicates less envy). Clearly, both EG and EEG obtain a value of one as they satisfy the EF property, thereby confirming the analysis in Section III-C; the same applies to PF, as a proportional allocation is envy-free by definition. SWM and MMF are not envy free unless the number of sellers is very large, i.e., at least 50. Overall, the results confirm that the proposed EEG scheme for selling weights implies that no SPs prefer (envy) the allocation and prices of any other SP.

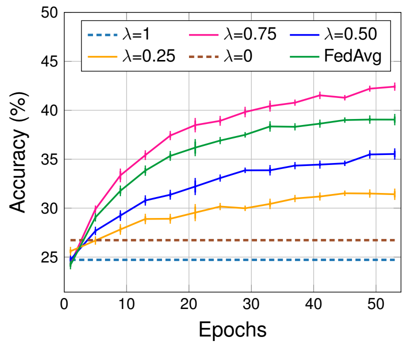
Finally, Fig. 3c shows the execution time for solving Problem 2 (i.e., for EEG) as a function of the number of buyers, for different amounts of sellers. Solving the problem takes less than 5 seconds even for the most demanding scenario with 512 buyers and 200 sellers, and less than 2 seconds in most cases. When the number of sellers ranges between 5 and 40, the execution time is less than one second even for 512 buyers. These results clearly demonstrate that the proposed solution for selling weights is efficient and scalable.

B. Weight Fusion

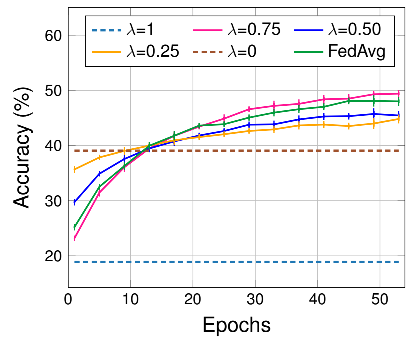
The rest of the section characterizes the gain in accuracy resulting from knowledge sharing, particularly, through the weight fusion technique introduced in Section III-B. For this purpose, it considers buyers that acquired weights from a single or multiple sellers in the allocation obtained by EEG. The results report the inference accuracy as a function of the training epoch for different values of the λ parameter. Recall that $\lambda = 0$ keeps the weights of the buyer as they are, while $\lambda = 1$ replaces the local weights with those of the seller; the source DNNs are not retrained for these two special cases as baselines. In contrast, the weights are first fused and the DNNs are fine tuned [32] with the data available at the buyer for all other values of λ. Moreover, different scenarios are considered in terms of how training data are distributed between the buyer and the sellers. For instance, a value of (30%, 70%) means that the buyer has 30% of the total training data (i.e., of both the buyer and the seller), while the seller holds 70% of such data (i.e., more than twice as many training data). The accuracy of the proposed solution is compared against that of federated learning [14], particularly, the widely-used FedAvg scheme [10] for weight fusion in that context.



(a)



(b)



(c)

Fig. 5: Accuracy as a function of the training epoch for knowledge fusion involving one buyer and two sellers for different distributions of training data between the buyer and the sellers: (a) (15%, 35%, 50%), (b) (33%, 33%, 33%), and (c) (50%, 30%, 20%).

Fig. 4 shows the accuracy as a function of the training epoch, for the case of a buyer that received weights from a single seller, under different distributions of training data. In all cases it is clear how weight fusion increases accuracy only within a few epochs, irrespective from the value of λ, especially when the data available at the buyer are much less than those at the seller. The accuracy more than doubles when the seller has 30% of the total training data (Fig. 4a). The λ parameter has a significant impact on the achievable accuracy and allows better performance by simply using the acquired weights in all cases. However, higher values of λ have a more significant impact on the accuracy when the difference between the training data at the buyer and at the seller is smaller (Figs. 4b and 4c). It is also worth noting that the difference in the accuracy between the two special cases (of $\lambda = 0$ and $\lambda = 1$) decreases as the difference in the amount of training data at the buyer and at the seller decreases too. FedAvg obtains an accuracy similar to using weight fusion with $\lambda = 0.5$, as it averages all the weights involved in the distributed training process (i.e., of both sellers and buyers). In contrast, the weight fusion approach employed here does not require exchanging weights multiple times during training, but leverages the allocation of weights found by the broker.

Similarly, Fig. 5 shows the accuracy as a function of the training epoch, for the case of a buyer that received weights from two sellers, under different distributions of training data. For simplicity, the same value of $\lambda = \lambda_1 = \lambda_2$ is considered to fuse weights from the two sellers. The results confirm the previous findings that a lower improvement in accuracy corresponds to a reduction in the amount of data available at the seller (as apparent from Figs. 5a and. 5b). The results for the case where the buyer holds the majority of training data are particularly important (Fig. 5c). They show that the buyer would benefit more from keeping the existing weights, instead of replacing them with the fused weights obtained from the two sellers. However, the advantage of joint weight fusion and transfer learning becomes clear, by exceeding the original accuracy after only 15 epochs. Different from the previous cases, the actual value of λ does not have a significant impact on the final accuracy after 50 epochs. The accuracy of FedAvg is higher than weight fusion with $\lambda = 0.5$ in presence of multiple sellers, especially when they possess more data than the buyer (i.e., in Figs. 5a and 5c). In these cases, the accuracy of FedAvg is significantly lower than weight fusion with $\lambda = 0.75$. This highlights the importance of trading off new and prior knowledge, which can be achieved by an appropriate setting of the λ parameter.

V. Related Work

The state of the art related to knowledge sharing can be broadly classified into economic models for resource or data allocation, transfer learning and data sharing, as well as distributed and federated learning. Relevant works in these categories are described next.

Economic models for resource allocation. Several works have considered competition among SPs in different scenarios. Huang et al. [33] propose a resource allocation model based on game theory that considers the competition between cloud and network SPs. In particular, the authors characterize the relationship between the market shares, service rates, and connectivity rates. Feng et al. [17] target a market with multiple competing IaaS cloud providers. The authors study monopoly, duopoly, and oligopoly to find optimal pricing for cloud services. In addition, iterative algorithms to find the equilibrium prices in duopoly and oligopoly are proposed. Similarly, other works focus on determining the optimal prices and resource allocation in cloud, edge and fog scenarios. Prasad et al. [34] propose an online Fisher's market that dynamically determines the price and the integral allocation of diverse resource types based on the supply and demand in a certain period of time. Li et al. [35] discuss a price-incentive reverse auction mechanism wherein cloud resources are sold to users by trading off the interests of both entities. In particular, such a mechanism maximizes the number of users while ensuring at least a given profit rate for the cloud SPs. Peng et al. [36] introduce a reverse auction mechanism for resource allocation in vehicular fog networks by considering multiple attributes (including location, reputation, and computing power) in addition to price. Li et al. [37] address effective allocation of crowdsourced edge resources to competing users. The authors devise an online privacy-preserving and a truthful double auction mechanism for dynamic resource pooling at the edge. Wang et al. [38] target edge intelligence realized through unmanned aerial vehicles by minimizing the average age of information. In this context, they leverage a potential game and a stochastic learning scheme that achieves a Nash equilibrium in a distributed manner. In contrast, this work devises a market model of knowledge sharing for the specific scenario in which different SPs provide AI-based services.

Transfer learning and data sharing. Transfer learning (TL) is based on leveraging knowledge acquired for one (base) task to solve a related task in a different (target) domain [8]. Yosinski et al. [32] show that the transferability of features in a DNN depends on the distance between the base and the target tasks, particularly, using features from a distant task is better than random selection. Wei et al. [39] discuss a novel TL framework that utilizes prior experience to automatically determine what to transfer and how. Moreover, recent research employs feature augmentation techniques such as replication, reduction, clustering, and alignment [40]. In contrast, this work does not develop a new TL architecture, but rather applies TL to combine the weights of pre-trained DNNs obtained as part of a knowledge sharing process. Rasouli and Jordan [4] characterize markets in which agents exchange data to carry out machine learning tasks. The authors consider trading data for money and characterize such a process as a network formation game. Instead, this work leverages a Fisher's market formulation for optimal knowledge sharing rather than deriving optimal prices.

Distributed and federated learning. Recently, federated learning (FL) has established itself as an emerging technology to solve the issues in conventional centralized solutions [10]. In FL, several devices collaborate to construct a global model while maintaining the privacy of their own data. Devices update the model locally and then send the model parameters to a central server that aggregates and re-distributes the updated model. Accordingly, recent works have focused on learning and improving a single global model driven by a central orchestrator. Zhang et al. [40] discuss a FL-based scheme that selects the orchestrator in a fog network by means of a voting mechanism. Such a voting scheme takes into account how fog nodes are close to each other and the computational resources required for the training process. Hosseinalipour et al. [41] propose a hierarchical-hybrid scheme, also based on FL, considering a layered network with multiple clusters of devices. In their solution, devices collaborate at the different network layers to achieve local consensus on model parameters. Local models are also combined with multi-stage parameter relaying between layers in a tree-like structure. Lee et al. [42] introduce the concept of opportunistic FL in which individual devices belonging to different users learn models personalized to the user's experience. These devices then incorporate new knowledge as they encounter each other by exchanging model parameters and gradients. Despite its advantages, FL has a communication overhead that is generally substantial, since the server sends the model to all clients repeatedly [43, 44]. Instead, this work relies on exchanging and combining the weights of pre-trained DNNs along with transfer learning. Consequently, it significantly reduces the communication overhead while also improving the accuracy of local predictions.

VI. Conclusion

This work addressed knowledge sharing between multiple providers of AI services. In particular, the related process exchanges the weights of pre-trained deep neural networks and then applies transfer learning. An economic model was introduced based on Fisher's market, in which service providers buy and sell weights, before incorporating the acquired knowledge into their own by means of weight fusion. An approach based on an extended Eisenberg-Gale program was also introduced to handle the realistic scenario where the demand for weights is finite. The proposed approach was proven to achieve a market equilibrium that satisfies different important economic properties: Pareto optimality, envy freeness, sharing incentives, proportional allocation, and computational efficiency. Moreover, weight fusion was shown to be effective in improving local accuracy when weights are obtained from other service providers, under different distributions of training data at the buyer and the sellers.

This work has characterized knowledge sharing by focusing on a single instance of the weight trading process. It would also be interesting to explore the dynamics that result from carrying out trading periodically over time. Considering alternative formulations of the knowledge sharing problem – for instance, by means of auction design – is another promising direction for future work.

References

[1] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, 2017.

[2] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[3] N. Mittal, D. Kuder, and S. Hans, "AI-fueled organizations." `https://bit.ly/2RBPmmP`, Jan. 2019. Deloitte Insights.

[4] M. Rasouli and M. I. Jordan, "Data sharing markets." arXiv:2107.08630 [econ], July 2021.

[5] M. Kop, "Machine learning and EU data-sharing practices: Legal aspects of machine learning training datasets for AI systems," in *Research Handbook on Big Data Law*, ch. 22, pp. 432–453, Edward Elgar Publishing, 2021.

[6] A. Campion, M.-G. Hernandez, S. M. Jankin, and M. Esteve, "Managing artificial intelligence deployment in the public sector," *Computer*, vol. 53, no. 10, pp. 28–37, 2020.

[7] Z.-H. Zhou, "Learnware: on the future of machine learning," *Frontiers of Computer Science*, vol. 10, no. 4, pp. 589–590, 2016.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[9] A. H. Jiang, D. L.-K. Wong, C. Canel, L. Tang, I. Misra, M. Kaminsky, M. A. Kozuch, P. Pillai, D. G. Andersen, and G. R. Ganger, "Mainstream: Dynamic stem-sharing for multi-tenant video processing," in *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, (Boston, MA), pp. 29–42, USENIX Association, July 2018.

[10] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, vol. 54, pp. 1273–1282, Apr. 2017.

[11] B. D. Deebak, F. H. Memon, S. A. Khowaja, K. Dev, W. Wang, N. M. F. Qureshi, and C. Su, "Lightweight blockchain based remote mutual authentication for AI-empowered IoT sustainable computing systems," *IEEE Internet of Things Journal*, pp. 1–1, 2022.

[12] T. Rekatsinas, A. Deshpande, and A. Machanavajjhala, "SPARSI: Partitioning sensitive data amongst multiple adversaries," *Proc. VLDB Endow.*, vol. 6, pp. 1594—-1605, Aug. 2013.

[13] I. Ceballos, V. Sharma, E. Mugica, A. Singh, A. Roman, P. Vepakomma, and R. Raskar, "SplitNN-driven vertical partitioning." arXiv:2008.04137 [cs], Aug. 2020.

[14] P. Kairouz *et al.*, "Advances and open problems in federated learning." arXiv:1912.04977 [cs], Dec. 2019.

[15] S. Brânzei, Y. Chen, X. Deng, A. Filos-Ratsikas, S. K. S. Frederiksen, and J. Zhang, "The Fisher market game: Equilibrium and welfare," in *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, p. 587–593, AAAI Press, 2014.

[16] T. Mohammed, C. Joe-Wong, R. Babbar, and M. Di Francesco, "Distributed inference acceleration with adaptive DNN partitioning and offloading," in *The 39th IEEE International Conference on Computer Communications (INFOCOM 2020)*, pp. 854—863, July 2020.

[17] Y. Feng, B. Li, and B. Li, "Price competition in an oligopoly market with multiple IaaS cloud providers," *IEEE Transactions on Computers*, vol. 63, no. 1, pp. 59–73, 2014.

[18] N. R. Devanur, C. H. Papadimitriou, A. Saberi, and V. V. Vazirani, "Market equilibrium via a primal–dual algorithm for a convex program," *J. ACM*, vol. 55, Nov. 2008.

[19] E. G. Jones, "Neuroanatomy: Cajal and after Cajal," *Brain Res. Rev.s*, vol. 55, no. 2, pp. 248–255, 2007.

[20] S.-A. Naas, T. Mohammed, and S. Sigg, "A global brain fuelled by local intelligence: Optimizing mobile services and networks with AI," in *2020 16th International Conference on Mobility, Sensing and Networking (MSN)*, pp. 23–32, 2020.

[21] M. Denil, B. Shakibi, L. Dinh, M. Ranzato, and N. de Freitas, "Predicting parameters in deep learning," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, pp. 2148–2156, 2013.

[22] T. Unterthiner, D. Keysers, S. Gelly, O. Bousquet, and I. Tolstikhin, "Predicting neural network accuracy from weights." arXiv:2002.11448 [stat], Feb. 2020.

[23] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. Cambridge University Press, 2007.

[24] N. R. Devanur, K. Jain, T. Mai, V. V. Vazirani, and S. Yazdanbod, "New convex programs for Fisher's market model and its generalizations." arXiv:1603.01257 [cs], Mar. 2016.

[25] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[26] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 281–302, 2000.

[27] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," in *2013 IEEE International Conference on Computer Vision Workshops*, IEEE, 2013.

[28] S. Brânzei, N. Devanur, and Y. Rabani, "Proportional dynamics in exchange economies," in *Proceedings of the 22nd ACM Conference on Economics and Computation*, p. 180–201, ACM, 2021.

[29] Y. Jiao, P. Wang, D. Niyato, and Z. Xiong, "Social welfare maximization auction in edge computing resource allocation for mobile blockchain," in *2018 IEEE International Conference on Communications (ICC)*, IEEE, 2018.

[30] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2909–2913, 2016.

[31] MOSEK ApS, "The MOSEK optimization suite." `https://www.mosek.com/`. Accessed July 26, 2021.

[32] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.

[33] J. Huang, J. Zou, and C.-C. Xing, "Competitions among service providers in cloud computing: A new economic model," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 866–877, 2018.

[34] A. S. Prasad, M. Arumaithurai, D. Koll, Y. Jiang, and X. Fu, "OFM: An online Fisher market for cloud computing," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, IEEE, 2019.

[35] S. Li, J. Huang, and B. Cheng, "A price-incentive resource auction mechanism balancing the interests between users and cloud service provider," *IEEE Transactions on Network Service Management*, vol. 18, no. 2, pp. 2030–2045, 2021.

[36] X. Peng, K. Ota, and M. Dong, "Multiattribute-based double auction toward resource allocation in vehicular fog computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3094–3103, 2020.

[37] Y. Li, H. C. Ng, L. Zhang, and B. Li, "Online cooperative resource allocation at the edge: A privacy-preserving approach," in *2020 IEEE 28th International Conference on Network Protocols (ICNP)*, IEEE, 2020.

[38] W. Wang, G. Srivastava, J. C. Lin, Y. Yang, M. Alazab, and T. R. Gadekallu, "Data freshness optimization under CAA in the UAV-aided MECN: A potential game perspective," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–10, 2022.

[39] Y. Wei, Y. Zhang, J. Huang, and Q. Yang, "Transfer learning via learning to transfer," in *Proceedings of the 35th International Conference on Machine Learning* (J. Dy and A. Krause, eds.), vol. 80 of *Proceedings of Machine Learning Research*, pp. 5085–5094, PMLR, July 2018.

[40] T. Zhang, Z. Shen, J. Jin, X. Zheng, A. Tagami, and X. Cao, "Achieving democracy in edge intelligence: A fog-based collaborative learning scheme," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2751–2761, 2021.

[41] S. Hosseinalipour, S. S. Azam, C. G. Brinton, N. Michelusi, V. Aggarwal, D. J. Love, and H. Dai, "Multi-stage hybrid federated learning over large-scale d2d-enabled fog networks," *IEEE/ACM Transactions on Networking*, pp. 1–16, 2022.

[42] S. Lee, X. Zheng, J. Hua, H. Vikalo, and C. Julien, "Opportunistic federated learning: An exploration of egocentric collaboration for pervasive computing applications," in *2021 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, IEEE, 2021.

[43] M. Asad, A. Moustafa, T. Ito, and M. Aslam, "Evaluating the communication efficiency in federated learning algorithms." arXiv:2004.02738 [cs], Apr. 2020.

[44] F. Sattler, S. Wiedemann, K.-R. Muller, and W. Samek, "Robust and communication-efficient federated learning from non-i.i.d. data," *IEEE Transactions on Neural Networks Learning Sys.*, vol. 31, no. 9, pp. 3400–3413, 2020.