DataBright: A Data Curation Platform for Machine Learning based on Markets and Trusted Computation

David Dao[1], **Dan Alistarh**[2], **Claudiu Musat**[3], **Ce Zhang**[1]
[1] ETH Zurich
[2] IST Austria
[3] Swisscom
david.dao@inf.ethz.ch, dan.alistarh@ist.ac.at, claudiu.musat@swisscom.com, ce.zhang@inf.ethz.ch

Abstract

Developing and deploying machine learning systems requires large amounts of data and computational resources. Thus, it is hard for an individual to access and customize this technology without relying on the infrastructure provided by large tech companies. Even if users invest into computational resources, it is often impossible to collect large amounts of data by themselves. Traditional crowdsourcing systems offer solutions to circumvent the data bottleneck but could suffer from short-term incentives: without careful quality control, it could limit the quality and diversity of the data.

We present DATABRIGHT, an end-to-end data curation platform for machine learning based on novel incentives, building on decentralized data ownership and smart contracts. DATABRIGHT rewards successful data contributors with *shares* in the overall data set instead of immediate small payments. This turns crowdsourcing into an investment in the future value of the created data. At the logical level, DATABRIGHT is a “database” whose content is curated by data contributors in a decentralized fashion — data contributors propose new data points to be added to a relation and will become a shareholder of the corresponding relation once peer shareholders accept their proposal by voting. The DATABRIGHT data market is complemented by a *trusted computational market*, which allows users to train models over DATABRIGHT datasets in an efficient and trustworthy manner.

1 Introduction

Learning systems are starting to play a crucial role in today’s society. The predictive power of learning systems is evident in applications such as self-driving cars, targeted advertising, and advances in the natural sciences. Unfortunately, most of today’s learning systems require a significant “concentration” of data and computation, usually only accessible to tech giants such as Google, Facebook and Amazon. This oligopoly in terms of data ownership and AI infrastructure makes it hard for independent researchers and scientists to create high-performance AI algorithms and systems without relying on such infrastructure. Thus, for the time being, predictive analytics, also known as prediction-as-a-service, are bound to be dominated by a few such large players.

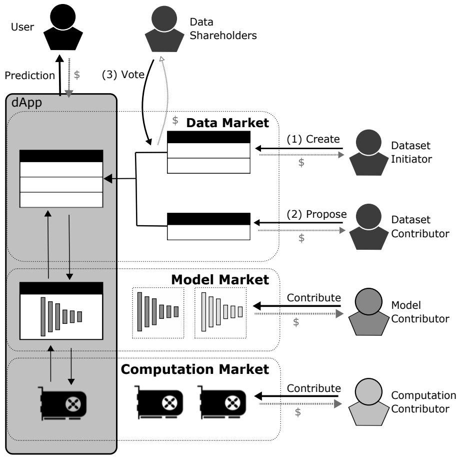

Figure 1: Architecture overview of DATABRIGHT

We are motivated by the following question: *Is it possible to leverage the power of crowds to build a democratic machine learning ecosystem which distributes payments fairly to data and computation contributors?* We introduce DATABRIGHT, a crowd-based data curation market for machine learning which approaches decentralized prediction-as-a-service through an economic perspective. In DATABRIGHT, we see a prediction as the end product of three main components: the *data*, the *machine learning model* trained on the data, and the *computation* used to train the model. The goal of DATABRIGHT is to distribute each of components to contributors from the crowd.

In our view, a key factor in enabling a successful crowd-based system is to combine incentives, ownership and accessibility:

1. How can we provide long-term incentives for our contributors? Can we tailor these incentives to ensure domain-relevant goals such as high data quality and fast data acquisition?

2. How can we lower the barrier for people to contribute and engage in this ecosystem?

3. How can we ensure trust and transparency into the system?

Quality control for crowdsourcing has been studied intensively to address these concerns [Allahbakhsh *et al.*, 2013]. The current version of DataBright does not fully answer these questions, but we believe that it provides a fresh perspective to quality control for crowdsourcing. In the meantime, we also believe that DataBright can enable an interesting discussion among the research community. DataBright exhibits a set of mechanisms which incentivize the crowd to contribute to the overall ecosystem, both through quality data, and computational power. See Figure 1 for a high-level system representation. Specifically, we developed a decentralized application (dApp) on top of *Ethereum* as a fault-tolerant logic that combines the markets to run and self-enforce contracts, without the need of a trusted third party. For the computation market, we design a hybrid platform based on Intel Software Guard Extensions (SGX), which provide trusted computation, and untrusted GPUs to balance between security, trust, and system performance when training machine learning models.

2 Decentralized Data Ownership

In crowdsourcing platforms such as Amazon Mechanical Turk, data contributors have no stake in the trained model, and thus little incentive, besides the immediate financial gain, to provide quality data. The emergence of distributed ledger technology allows us to consider alternatives. In DataBright, we designed the data market to allow every data contributor to become a data *investor*.

Interaction Model Users interact with the data market as follows. A **data initiator** creates a data request. **Data contributors** create a data proposal linking to data that are stored outside the blockchain. In a second step, data curators holding computational tokens see and vote on data proposals. This ensures the integrity of the created data. The process can be further extended with automated quality assessment of the data contributions, for instance using pre-trained models (such as image classifiers). If a data proposal reaches a vote threshold it gets accepted and stored as immutable entry into a data registry located on the chosen blockchain.

Implementation and Techniques The data market implements the following mechanisms to ensure long-term incentives:

1. Immutable data ledger. To correctly identify and distribute payments, it is required to track each individual data points to their contributors. Blockchain data ledgers are immutable, and thus can correctly and consistently assign credit to contributors.

2. Decentralized voting. Data contributors can earn credits for each model trained using their contributed data. At the same time, we give them a chance to decide on the *inclusion* of future data points in the dataset. This serves as a quality control mechanism, making it more attractive for end users to use this data.

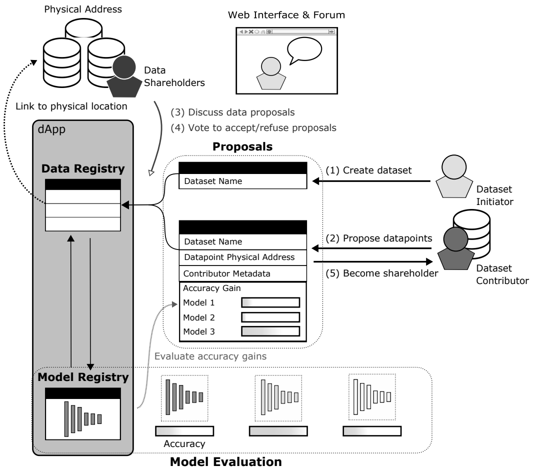

Figure 2: Overview of the DataBright data market

Estimating importance of data. Different strategies exist to evaluate the importance of certain data points with regards to the overall accuracy of a model. We implement "diminishing return" incentives which rewards early contributors to a data set by measuring the quality difference between the model trained on all previously contributed data to that trained on all previously contributed data plus a given data point.

Figure 2 gives us an overview of the data market. The data market is controlled by a smart contract inside the dApp, which is deployed on top of the *Ethereum* blockchain and owned by the community of data shareholders.

2.1 Decentralized data ledger and voting

The data registry is a ledger which links accepted data points to the (wallet) address of their respective contributors. It is implemented as a smart contract on top of the *Ethereum* blockchain, thus providing decentralized and immutable guarantees. The ledger includes a hash table which links the metadata associated to the data points, which is stored on the ledger, with the physical data file, which is stored on a cloud or decentralized storage service [Benet, 2014].

The procedure to add a new datapoint to an existing dataset is as follows. The contributor creates a proposal, containing metadata, physical address information, and the dataset to which the sample should be added. This proposal gets submitted, and then voted on through a decentralized voting scheme, which we implement as a smart contract on top of Ethereum. If accepted, then the new data gets added to the registry for this dataset, and the contributor gets assigned *shares* in the dataset, corresponding to its contribution, and will be allowed to vote on future proposals. Shares are implemented as decentralized digital tokens within the smart contract. For each data proposal, additional information can be requested, such as metadata about the contributor, or automated accuracy evaluation of the data proposal (for more details on this, see Section 2.3). Further, a web interface with a discussion forum is provided to discuss each submitted data proposal. Datasets are created in a similar manner, and submitted directly to the data registry.

Traditional crowd-sourcing systems such as Amazon Me-

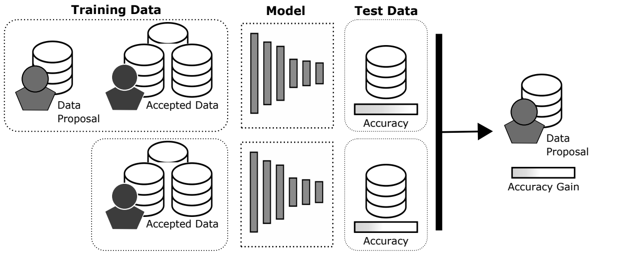

Figure 3: Model accuracy gain evaluation for data proposal using leave-one-out strategy

chanical Turk perform quality control from the data requester. Furthermore, it is common, in order to preserve a good reputation for future labeling requests, for the requester to just accept the data without additional checks. By contrast, our thesis is that successful data contributors should be seen as domain experts for their respective data set, with a stake in its quality. Thus, their vote is a valuable estimate on data quality.

2.2 Model registry and markets

A user can either train their own model on data *within the market*, or perform inference over a pretrained model. Thus, DATABRIGHT introduces a model market and respective model registry, where algorithm developers can submit and register their models, while training them on data sets linked to the data registry. Distributed training is run within a trusted computation market, discussed in Section 3. The model registration and submission processes are similar to the data submission process described in the previous section. The model registry, like the data registry, is part of a smart contract within the dApp and stores all available public models within DATABRIGHT with the hash and the physical address of its best performing model checkpoint. Models are owned by model contributors (e.g. developers) who can earn cryptocurrencies for each prediction requested from their model by a user.

2.3 Automated data proposal evaluation

The model market provides new data incentives and quality control mechanisms. Ensuring high data quality and right decision making during voting can be a challenging task, especially if the data shareholder community is small and the amounts of data proposals is unfeasible to check manually. Thus, we want to incentivize quick growth of good data contributions and data shareholders early on while on the same time allowing automated evaluations of data proposals. Fine-tuning models provided in the model market with submitted data proposals allows us to evaluate accuracy gains respectively for each proposal (by comparing to the stored best performing checkpoint, see Figure 3). We can thus reward data proposals with high accuracy gains (or provide it as additional information during voting). This encourages early contributions, as diminishing returns make it more difficult to achieve significant accuracy gains over time when the accepted data set becomes large. However, since we tie the "value" of a data proposal to model prediction and training, we require reliable training, which is performed through the trusted computation platform.

	SGX overhead	Forward Pass	Comm. Time	Epoch Time
Standard, 1 GPU	N/A	92 ms	0 ms	193 min
Standard, 2 GPUs	N/A	52 ms	0 ms	118 min
No splitting, 1 GPU	176.66 ms/run	92 ms	0 ms	193 min
No splitting, 2 GPUs		46 ms	2622 ms	1749 min
2-Way Splitting, 2 GPUs		93 ms	36 ms	225 min

Figure 4: Trusted computation runtime

3 Trusted Computation Market

The DATABRIGHT computation market consists of a pool of devices provided by computation contributors. As before, contributors are stored in a ledger inside the dApp. Whenever a contributor's device is used for training a model or predictions, the contributor gets paid through cryptocurrency. (We allow computation contributors to set a minimum fee above which their device should be used.) The key challenge is providing *trusted computation*: How can we ensure that the contributor's device(s) are actually returning the right model or prediction? For example, if the user is paying for a prediction from a certain trained model, how can we ensure that DATABRIGHT does not just return a random list of numbers, or a partially trained model?

To deal with these concerns, DATABRIGHT's computation market uses a hybrid model, which assumes a small set of trusted devices, implementing trusted hardware with Intel SGX, and a larger pool of untrusted workers, e.g. GPUs. All scheduling and verification steps happen on trusted Intel SGX devices, which provide a proof to the user of the exact program gets executed. While untrusted devices will be used for the bulk of the work. As shwon in Figure 5, we implement the following protections:

1. Triple modular redundancy (TMR). Users can choose to have untrusted computation be protected with standard TMR [Lyons and Vanderkulk, 1962]. In a nutshell, trusted devices randomly sample untrusted devices and form three redundancy groups, each of which conducts the same computation. Trusted devices will only return the result to a user when all these redundancy groups return the same result.

2. Periodical Reallocation. One possible attack is that an untrusted worker records the data it receives, and resells it. To prevent this, the trusted workers will limit the amount of data an untrusted worker can see to at most 5% of the whole training set.

3. Model Splitting. Users may wish to avoid a worker having access to the trained model. DATABRIGHT provides an *optional* way to avoid this by splitting the models into pieces, and put each piece on a different randomly sampled untrusted worker. The untrusted workers then communicate the activations and gradients of a single layer; no worker has access to the full model.

We illustrate the performance overhead introduced by our trusted computation design, we trained a VGG-16 network on ImageNet with three workers – one trusted worker with Intel SGX support, and two untrusted GPU machines. Machines are connected by a (slow) 1Gbps network. For *model splitting*, we split into two pieces, at the fully-connected layer and use batch size 32. As shown in Figure 4, executing

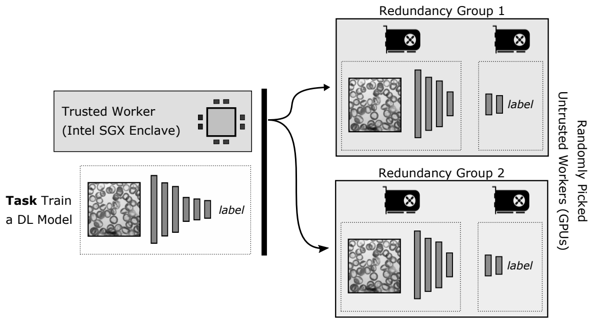

Figure 5: Overview of the trusted computation market

Figure 6: User interface for DataBright. The model registry (left) shows all pretrained models. The shareholder dashboard (right) allows to comment and vote on data proposals.

the scheduling and transmission via Intel SGX/TLS introduces almost negligible overhead. Parallelizing across two nodes (with no splitting) introduces significant communication overhead, due to the slow network. There is a vast literature on reducing these costs, e.g. [Seide and others, 2014] [Alistarh *et al.*, 2017] [Zhang and others, 2017b]. Model splitting introduces extra overheads compared with using just one GPU. Some of these overheads are not fundamental, and can be fixed through better system optimization.

4 Possible Applications

In the following we envision diverse applications that DataBright could enable.

1. (Scientific) data curation, peer-review and impact measurement. In the first application, a scientist collected valuable lab data such as microscopy images of cellular tissues, in order to help build a predictive model for drug discovery. DataBright provides a *peer-review layer* through the submission process, allowing for data feedback from collaborators, through the discussion associated with the data proposal. Upon acceptance of the data proposal, DataBright will allow to track the impact of the contributed scientific data set by displaying which other scientists are using the data to train predictive models and how much it improves accuracy in comparison to the existing data sets.

2. Prediction-as-a-service and micropayments. In the second application, a data scientist wants to build a machine learning application and browses through the data registry of DataBright. She selects a crowd-sourced dataset, which is curated in the first application scenarios, from the registry and requests to train a predictive model, which can be then submitted into the model registry without the need to leave DataBright. Using the model registry, she can then monetize the model through the dApp.

3. An accessible computation market. DataBright can be a low-cost alternative to cloud-based computation providers; it can support popular classification datasets such as ImageNet, CIFAR, and OpenNMT by default, and allow users to submit jobs directly to the computational market on these freely available datasets. The goal here is to decrease the relatively high current cost of fast computational devices, such as GPUs.

5 Related Work

There is a diverse literature on data acquisition and quality assurance, that is complementary to the described system. The leading crowdsourcing platform for the academic community is Amazon Mechanical Turk (AMT). AMT is a microtask crowdsourcing platform where the crowd can be accessed in a on-demand fashion and data contributors earn a (small) financial reward for each of their data contributions. CrowdDB [Franklin *et al.*, 2011] builds a relational database engine on top of an infrastructure such as AMT. It introduces an extension to SQL, CrowdSQL, that allows to query the crowd for new data and ask for subjective comparisons. For each new query, CrowdDB creates a custom user interface and uploads the task on AMT. DataBright is different from CrowdDB in its decentralized fashion, which we hope would be an alternative to AMT altogether. Quality control [Allahbakhsh *et al.*, 2013] and incentives [Chai and others, 2018] have been studied intensively, and we believe that DataBright provides an orthogonal, and fresh perspective to this topic. The combination of blockchain and SGX has been explored by Zhang et al. [Zhang and others, 2017a]. DataBright takes advantage of both Intel SGX and untrusted GPUs achieve effective, secure training of deep neural networks.

References

[Alistarh *et al.*, 2017] Dan Alistarh, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Randomized quantization for communication-optimal stochastic gradient descent. *NIPS*, 2017.

[Allahbakhsh *et al.*, 2013] M. Allahbakhsh, B. Benatallah, A. Ignjatovic, H. R. Motahari-Nezhad, E. Bertino, and S. Dustdar. Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, pages 76–81, 2013.

[Benet, 2014] Juan Benet. Ipfs-content addressed, versioned, p2p file system. *arXiv*, 2014.

[Chai and others, 2018] Chengliang Chai et al. Incentive-based entity collection using crowdsourcing. In *ICDE*, 2018.

[Franklin *et al.*, 2011] Michael J. Franklin, Donald Kossmann, Tim Kraska, Sukriti Ramesh, and Reynold Xin. Crowddb: Answering queries with crowdsourcing. In *SIGMOD*, 2011.

[Lyons and Vanderkulk, 1962] R. E. Lyons and W. Vanderkulk. The use of triple-modular redundancy to improve computer reliability. *IBM J. Res. Dev.*, 1962.

[Seide and others, 2014] Frank Seide et al. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, 2014.

[Zhang and others, 2017a] Fan Zhang et al. REM: Resource-efficient mining for blockchains. In *USENIX Security*, pages 1427–1444, 2017.

[Zhang and others, 2017b] Hantian Zhang et al. ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning. In *ICML*, 2017.