Collaboratively Building a Machine Learning Dataset on Layer 2 Blockchain

Yeon Joon Jung

StarkNet House Research Fellow

Abstract

Machine learning has grown rapidly over the past few years, expanding its application across various industries for its ability to predict outcomes with higher accuracy and efficiency. However, datasets we use for our machine learning models tend to lack security, response to data loss, and transparency due to their highly centralized nature. With blockchain offering us a decentralized, secured, and trustless system that resolves these problems, there had been various studies on integrating blockchain in building machine learning datasets. Layer 2 enables us to solve the scalability issue that revolved around many blockchain-based applications when we cannot give up both security and decentralization, and it creates machine learning datasets in a more scalable, cost-friendly manner. In this paper, we introduce a machine learning dataset that can be built on a Layer 2 solution, allowing decentralized, collaborative features where anyone can contribute to building the dataset.

Introduction

In recent years, the concept of artificial intelligence and machine learning has grown much more familiarity among the public, and its applications have integrated into our lives without notice. It isn't difficult to witness autonomous vehicles that can sense environments and drive safely with little or no human involvement. We often use virtual assistants embedded in phones and personal computers that use speech recognition engines. With machine learning expanding its significance over a wide variety of applications, the significance of datasets that we use for these algorithms has reached a level of significance that cannot be disregarded.

Data takes a crucial role in the machine learning process. Collected sets of data are the primary source used to train machine learning algorithms we build, making the algorithms' accuracy and reliability heavily dependent on them. However, even though many are aware of the importance of these data, the way we currently create and store them doesn't catch up to their importance, containing many flaws regarding their legitimacy and vulnerability.

In this paper, we will discuss the current state of the art in how datasets for machine learning are created and stored, along with examining several concerns these present-day practices hold. We

will then explore a possible suggestion to the identified concerns using a layer 2 blockchain network over Ethereum. Specifically, we will use StarkNet—a permissionless decentralized zero-knowledge rollup—in our example.

State of the Art

Storing datasets in centralized databases involves risk. Like any other examples of centralized management models, this approach requires full trust in the people/organization who manages the storage of the data. Hence, there exists a possibility of data being corrupted or tampered with in some sort by someone. Centralizing databases may also lead to the risk of data loss.

If so, would forbidding any modifications or changes to the data solve the issue? Unfortunately, no. Even if we exclude any malicious action that tries to corrupt the data, the data's quality cannot be guaranteed. In fact, poor-quality data is an issue that can't be ignored. IBM's estimate in 2016 on the yearly cost of poor-quality data, in the U.S. alone, was $3.1 trillion [1], showing that the impact data has is enormous. Datasets with proven high quality—in other words, for our sample to accurately represent the entire picture of the population—are essentials for developing machine learning models. For instance, outliers within the training dataset may result in instability and/or non-convergence, while incomplete, inconsistent, and missing data may cause a performance drop in prediction [2].

In 2008, the bitcoin system, the very first decentralized blockchain, was first conceptualized. In 2015, Ethereum, a blockchain with added functionality to support smart contracts, emerged. Since then, research on blockchain technology has surged. In the time frame from 2010 to 2018, there were a total of 4629 global publications associated with blockchain technology, recording an average annual growth rate of 150.24% [3]. With the blockchain system offering a trustless system that offers security and an easy-to-implement reward system that draws participants in, there had been studies and developments on utilizing the technology on building decentralized machine learning models.

In frameworks like 0xDeCA10B [4], a decentralized AI framework on the blockchain developed in Microsoft Research, anyone who wishes to participate can contribute to building a dataset and use smart contracts to host a constantly updating ML model that is shared on Ethereum. The framework encourages participants to provide new data through gamified, self-assessment incentive mechanisms. There also has been a suggestion of a system that establishes a marketplace where it enables exchanges of trained machine learning models on top of the Ethereum blockchain (DanKu) [5]. Nonetheless, one disadvantage of blockchain limits its growth: the issue of scalability. With Layer 1 protocols like Ethereum having an increased load of transactions and nodes count in the network, the cost associated has risen significantly as well.

Protocol

StarkNet attempts to solve the scalability issue while preserving the composability and security of Layer 1 Ethereum using a permissionless decentralized zero-knowledge rollup, leading to a reduction of transaction costs by 100 times.

In this paper, we explore a simple protocol that can be built on StarkNet: a StarkNet smart contract containing a machine learning dataset that is decentralized and collaborative. We will use a labeled dataset of housing prices for supervised learning as an example. The contract enables anyone who wishes to contribute to building a dataset to append data, offering a continuous update to the machine learning model when needed.

Below are several terminologies that we will use often for our dataset contract:

> A "contributor" is any participant who provides new data for the machine learning dataset.
>
> A "validator" is any participant who validates the contributed data, checking if there are mistakes, errors, or biases within them.
>
> The "dataset under review" is a collection of data that has not been verified by validators.
>
> The "validated dataset" is a collection of data that has been verified and can be used for machine learning models by anyone.

Let's start with a single piece of data. In our example, it would the price of a single house. In a housing model that predicts housing prices, the features could include house size in m^2, bedrooms count, bathrooms count, location (ZIP code), year built, number of schools, and crime rate. Our label will be its price. There can be several ways to store singular data in a dataset. Here we will define a struct that contains features and price labels for a house.

```
struct HousePrice:
    member house_size : felt
    member bedrooms : felt
    member bathrooms : felt
    member location : felt
    member year_built : felt
    member bedrooms : felt
    member school_count : felt
    member crime_rate : felt
    member price : felt
end
```

Figure 1. HousePrice Struct written in Cairo

These individual data pieces contributed by contributors must go through a verification process that checks their validity, to deal with data quality. The reasons, as mentioned previously, include detecting data containing missing or poorly written fields, finding noisy data that has conflicting or misleading information, or even detecting and preventing any "bot-attacks" that flood false or biased data.

The basic 3 steps of how a data piece gets added to the larger dataset are as followed:

1. A contributor submits data containing required features, as well as a label if it's supervised learning. All the data pieces contributed by users get appended to the collection of data we name "dataset under review".
2. The data in the "dataset under review" goes through a validation process by validators. This validation process will be collaborative where participants themselves validate each other's contributions. The process will also be democratic, where the decision will be done through a voting process. Any data that is considered to have errors/mistakes from the voting result gets removed from the dataset.
3. Once the data in the "dataset under review" is checked and validated, the data gets appended to the "validated dataset" containing clean, quality data that can be used by anyone for their machine learning model.

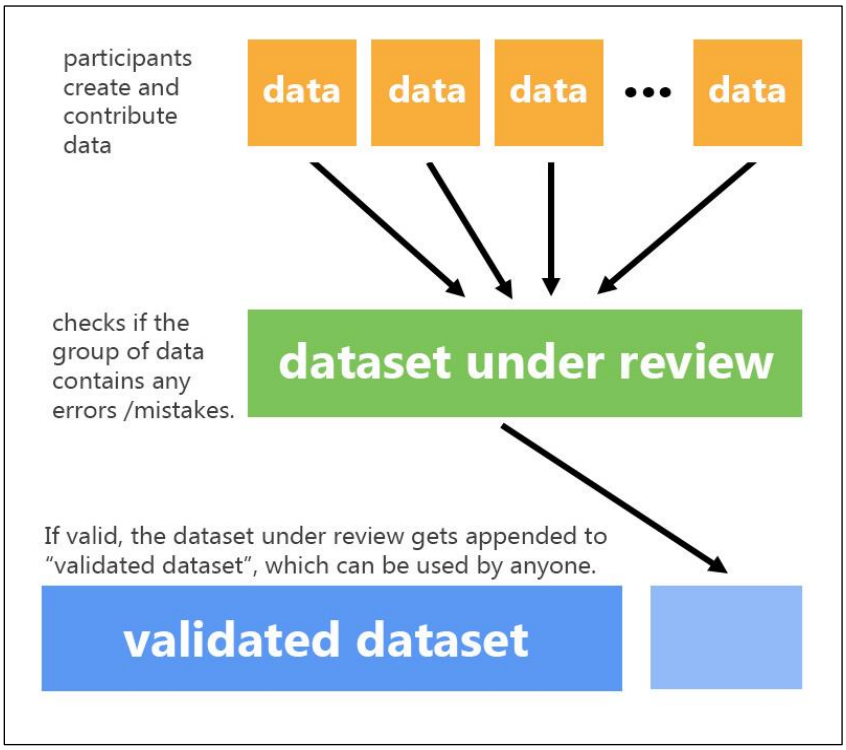

Figure 2. Data Validation Progress

There's a possibility of having a malicious user who uses "bot-attacks" that contributes tremendous amounts of false or biased data through automatically calling contribute function repeatedly, to disturb our ML dataset StarkNet contract. We can in fact prevent this case of a single user contributing too much data in a given time right when a user submits their contribution, before going through a validation process. We will use several Cairo features available when building our StarkNet contract to do this.

The unit of time we will use won't be measured in seconds, minutes, or days like we normally use in real life, but we will use block time, a time taken for the network to generate one extra block. Currently in StarkNet Testnet (Goerli)'s block time is around 1-2 minutes [6]. Thus, in our StarkNet contract, we could implement a storage variable that maps users to the number of requests they made in the current block time.

```
from starkware.starknet.common.syscalls import (
    get_block_number,
    get_block_timestamp,
    get_contract_address,
)

# res contains a tuple of two elements: current block time and # of
# requests, in that order.
# *  res : (block_time : felt, contribution_count : felt)
@storage_var
func user_contribute_count (user : felt) -> (res : (felt, felt)):
end

# function for users to contribute single data
@external
func contribute_single_data{
    syscall_ptr : felt*,
    pedersen_ptr : HashBuiltin*,
    range_check_ptr,
}(data : HousePrice):
    let (user) = get_caller_address()
    let (curr_block) = get_block_number()
    let (past_contributions) = user_contribute_count.read(user)
    let (last_request_time) = past_contributions[0]

    let (count) = past_contributions[1]
    user_contribute_count.write(user, (curr_block, 1))
    if curr_block == last_request_time:
```

```
        # doesn't allow more than 50 contributions per block
        assert count < 50
        user_contribute_count.write(user, (curr_block, count + 1))

    # ... more code...
end
```

Figure 3. Cairo code snippet for detecting bot-attacks

After the data goes through the "anti-bot filter" and gets appended to the dataset under review, we now go through the validation process (step 2 of the above timeline). To maintain a collaborative and decentralized behavior of our contract, we give participants themselves the authority to decide whether each data under review deserves to be added to the validated dataset. One way can be a voting system where each motivated participant can vote whether they approve the data or not, and its fate gets decided by the voting result. If the voting result indicates that the data is valid, the data gets appended to the validated dataset. Elsewise it gets discarded and is no longer under consideration.

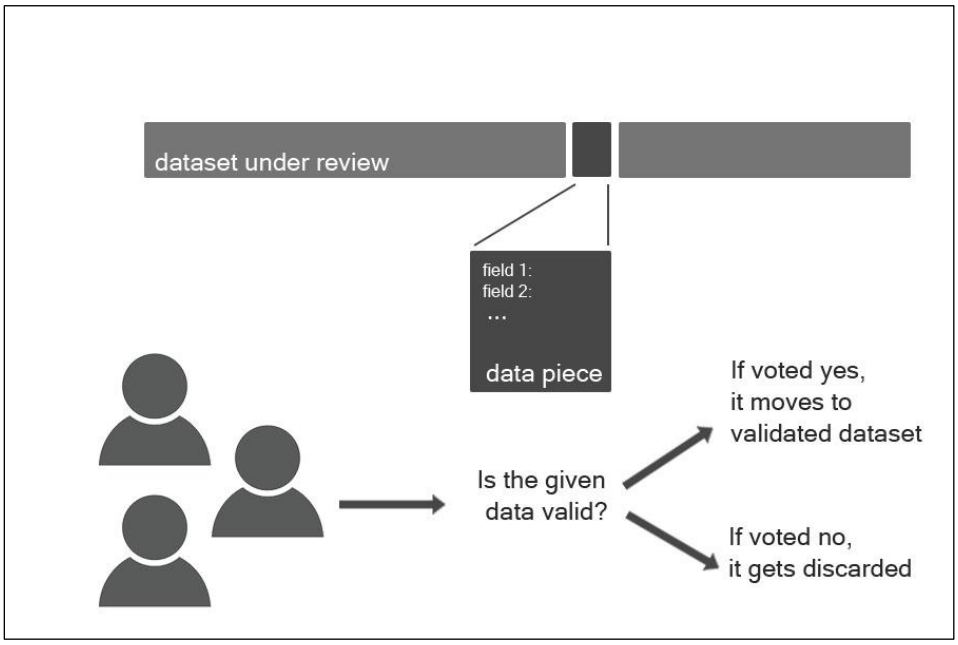

Figure 4. Simplified image of the data validation process

In our housing price dataset StarkNet contract, for instance, we may choose the voting system to include 10 different participants and must have at least 8 out of 10 votes that tell the data is valid.

The exact details of the voting system can vary, and it depends on the thoughts of the original creator of the dataset contract on what they believe is fair and faultless.

Below is an example of designing a simple collaborative machine learning dataset on StarkNet in a form of a block diagram. There can be various ways of designing it, depending on how we wish to build it.

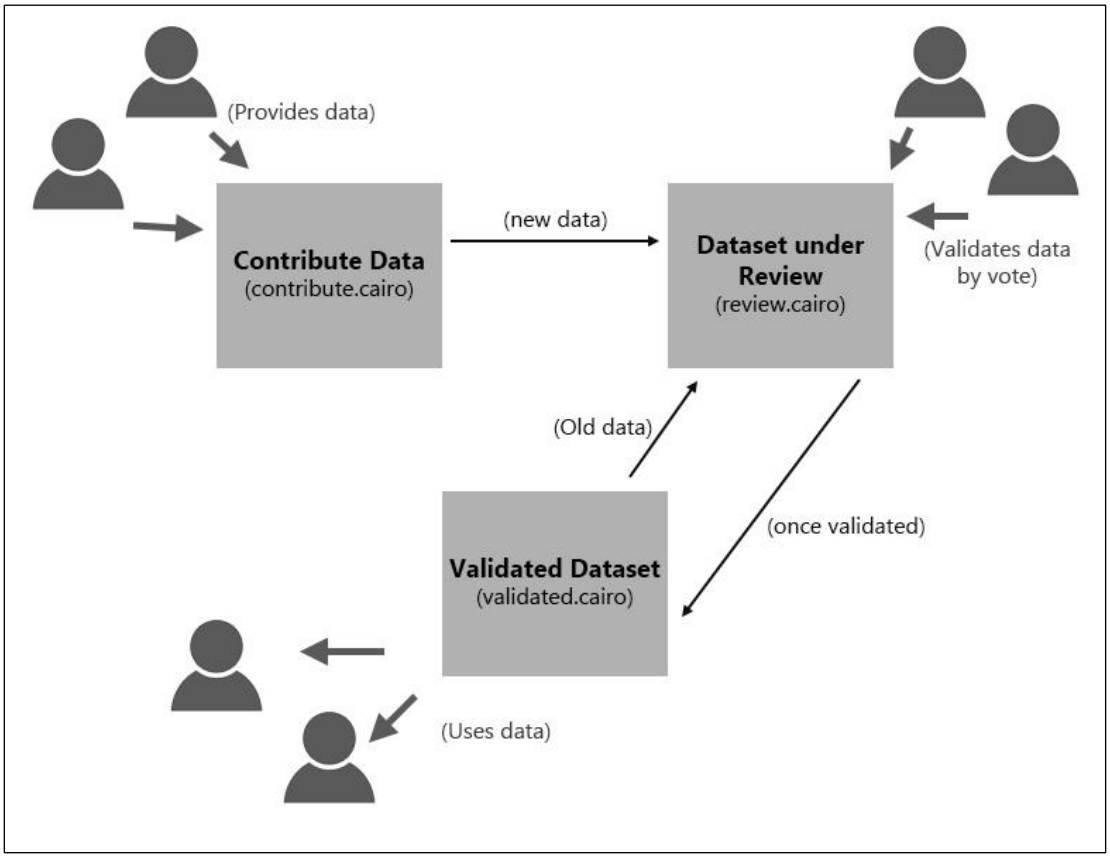

Figure 5. Example software structure for collaborative machine learning dataset

Incentive Mechanisms for Participants

Without any incentive mechanism, the collaborative community will soon die out. There can be several strategies to encourage users to contribute data and validate them. One way may be the idea of gamifying the entire dataset contract system that makes the data collection enjoyable for the users, including features such as participants earning points and badges when their contributed data gets validated [4], or giving cryptocurrency sent to their wallet as a reward for their contribution.

Conclusion

Layer 2 scaling solutions build another layer on top of an already existing layer 1 blockchain system, proceeding with transactions and flow of data off of layer 1 without tampering with its security and decentralization. Resolving the transaction speed, high-cost issues, and scaling difficulties that blockchain networks like Ethereum face, it allows more functionalities in blockchain applications.

As machine learning datasets involve huge amounts of data, we attempted to resolve their scalability issue when we implement them on the blockchain. In this paper, we presented an idea of storing a dataset in a form of a smart contract using StarkWare's StarkNet platform, a layer 2 scaling solution that uses permissionless decentralized zero-knowledge rollup. Adding collaborative features which enable participation for anyone who wishes to contribute, we hope this will spur on open-source machine learning communities.

References

[1] Redman, Thomas C. "Bad Data Costs the U.S. $3 Trillion per Year." *Harvard Business Review*, 4 Oct. 2017, hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year.

[2] Gudivada, Venkat, Amy Apon, and Junhua Ding. "Data quality considerations for big data and machine learning: Going beyond data cleaning and transformations." International Journal on Advances in Software 10.1 (2017): 1-20.

[3] Gupta, B. M., and S. M. Dhawan. "Blockchain Research a Scientometric Assessment of Global Literature during 2010 to 2018." *DESIDOC Journal of Library & Information Technology*, vol. 40, no. 01, 2020, pp. 397–405., doi:10.14429/djlit.40.01.14721.

[4] Harris, Justin D., and Bo Waggoner. "Decentralized and Collaborative AI on Blockchain." *2019 IEEE International Conference on Blockchain (Blockchain)*, 2019, doi:10.1109/blockchain.2019.00057.

[5] Kurtulmus, A. Besir, and Kenny Daniel. "Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain." arXiv preprint arXiv:1802.10185 (2018).

[6] *StarkNet - Alpha Block Explorer*, goerli.voyager.online/blocks.