A Decentralized Privacy-Preserving Predictive Modeling Framework on Private Blockcahin Networks

Chunhou Liu

Abstract

Cross-institutional predictive modeling can accelerate research and facilitate quality improvement initiatives. While privacy-protecting methods to build predictive models exist, most are based on a centralized architecture, which presents security and robustness vulnerabilities such as single-point-of-failure (and single-point-of-breach) and accidental or malicious modification of records. In this article, we improve the model in [2] which integrate privacypreserving online machine learning with a private Blockchain network, apply transaction metadata to disseminate partial models, and design a new proof-of-information algorithm to determine the order of the online learning process.

Introduction

In the model of proof-of-information algorithm [2], we find some weaknesses of the model: First, **The site in the private blockchain networks is not freely to exit.** To make it intuitive, We give an example. Suppose we have 4 sites in this private blockchain network, they works follows the model of [2]:

Assume Mt_s = model at time t on site s, Et_s = error at time t on site s. In the initialization stage (t = 0), each site trains their own model using their local patient data, and the model with lowest error (Site 1 with $E0_1 = 0.2$ in our example) is selected as the initial model.Conceptually, we regard $M0_1$ is transferred from Site 1 to Site 1 itself. Then, the selected model ($M0_1$) is submitted to Site 2, 3 and 4.

Next (t = 1), each site evaluates the model $M1_1$ (which is the same as $M0_1$) using their local data. Suppose Site 2 has the highest error ($E1_2 = 0.7$).

Site 2 wins the information bid, and the model $M1_1$ is now transferred to Site 2 within the block B1 (with amount = 0 and transaction fee = 0)

Then (t = 2), Site 2 updates the online machine learning model as $M2_2$ Again, Site 2 send $M2_2$ to all other sites, and the site with highest error (or richest information) wins the information bid to update the model locally (Site 3 in our example). This process repeats until a site updates the model and finds that itself has the highest error than all other sites.

Second, **The waiting time for collecting errors from others has to be manually.** In the proof-of-information algorithm of [2], the number of sites is determined to be N. As there could be an undetermined number of sites join and exit the private blockchain network, it's better to determined automatically.

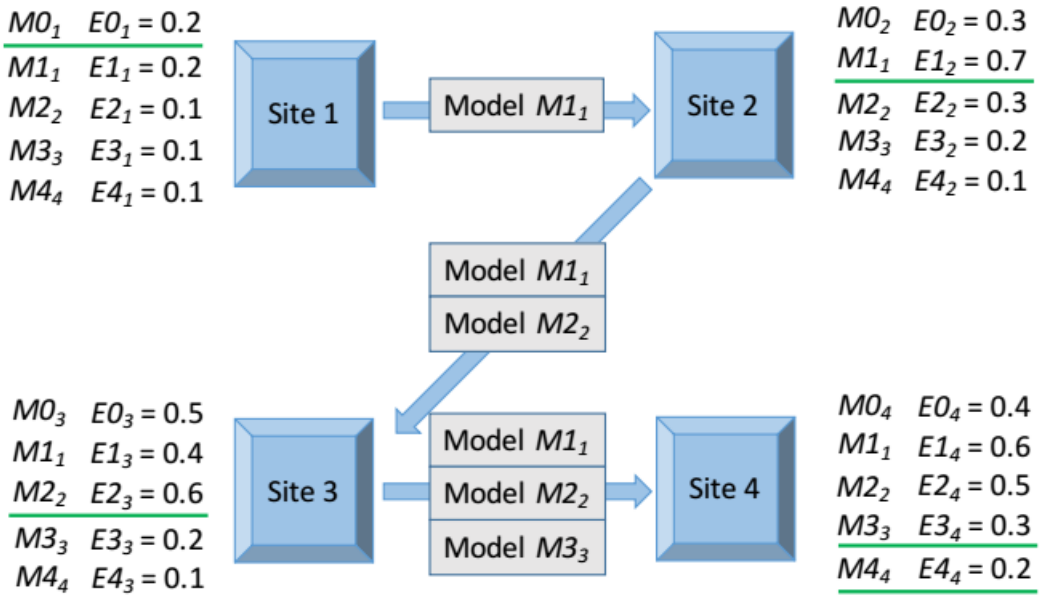

Figure 1: An running example of proof-of-information algorithm

Now we consider this situation that Site 2 exits after making the transaction at t=1 without updating the model, other sites will not be able to update the model because they have no right to update. The image of the blockchain will be stationary if there is no new site entering with largest evaluate error. The image of the blockchain states are showed:

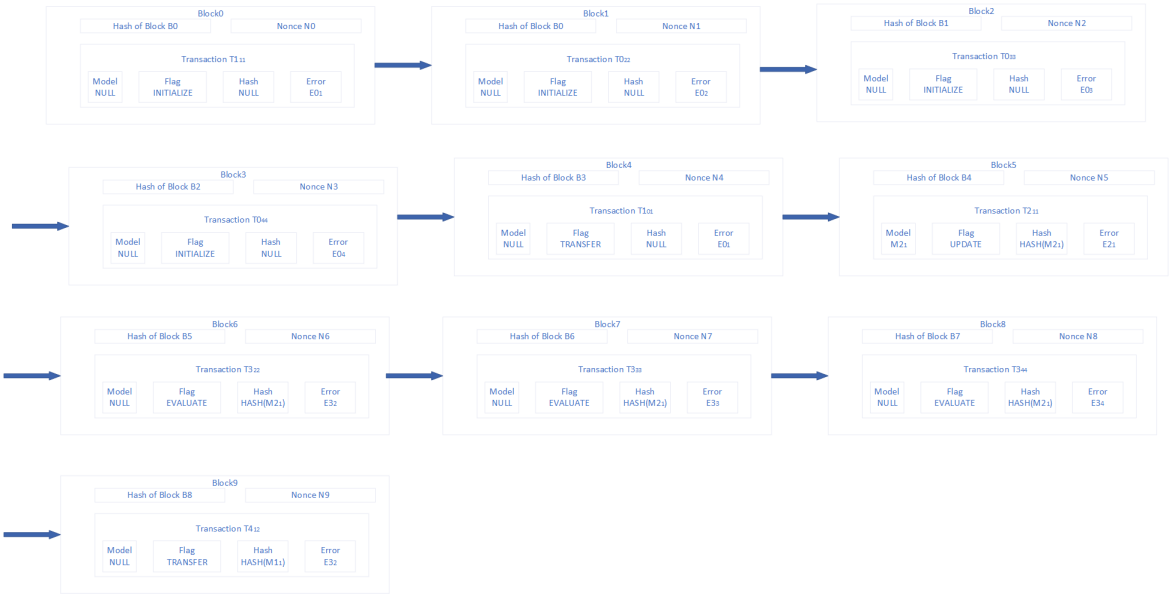

Figure 2: Blockchain States chains: if sites 2 exits, the blockchain will stationary until a new site entering.

Improved proof-of-information algorithm

We could add some enter information and exit information to the blockchain, when a site enter the private blockchain network, it evaluates the latest model to get error E, it makes a transaction to itself and write it to the blockchain, when the site wants to exit the private blockchain networks, it makes a transaction to itself with flag=EXIT and model=null, error=null.

The sites in the private blockchain networks inspect the blockchain in period. Each site maintains the information about the number of sites in this private blockchain, once a block with flag=INITIALIZE is received by the site, it increases the number and once a block with flag=EXIT is received it decrease the number, if the site exited is the site which gains the right to update the model and the current site has the largest error in the left sites, it will makes a transaction to itself. The algorithm for the new sites entering the private blockchain is showed in following:

Algorithm 1: Proof-Of-Information-New. This is the main algorithm for new participating site

Input : this site S, polling time period Δ
Output: The latest online machine learning model M

Learn Model M_S on the data in S and compute the error E;
Create a transaction from S to S itself with flag = INITIALIZE, model = NULL, hash = HASH (M_S), and error = E_S;
Retrive the latest Model M_C(generated by site C) and the largest current error E_C, set M = M_C;
Evaluate M_C on the data in S and compute the error E;
if $E > E_C$ **then**
    Create a transaction from C to S with flag = TRANSFER, model = NULL, hash = HASH (M_C), and error = E;
end
Set M = Proof-of-Information-Iteration(Δ);

Compare with the previous algorithm, We add an informing code which informs other sites that a new site entering.

Algorithm 2: Proof-Of-Information-Iteration.This is the core algorithm to determine the order of decentralized privacy-preserving online machine learning.

Input : this site S, polling time period Δ
Output: The latest online machine learning model M

Retrive the number of sites n from the blockchain according the INITIALIZE and EXIT flag block.;
for *every time period* Δ *check the blockchain* **do**
  if *There are new blocks with flag=INITIALIZE* **then**
    add the sites number n;
  end
  if *There are new blocks with flag=EXIT and site C* **then**
    decrease the site number n;
    if *C gained the right to update the model and S has the largest error in the left sites* **then**
      Create a transaction from C to S with flag = TRANSFER, model = NULL, hash = HASH (M_C), and error = E_S;
    end
  end
  if *There are new blocks with flag=UPDATE* **then**
    Retrieve the latest model M_C (generated by site C) and current largest error E_C from the block chain;
    Set M = M_C;
    Evaluate M_C on the data in S and compute the error E;
    Create a transaction from S to S itself with flag = EVALUATE, model = NULL, hash = HASH (M_C), and error = E;
  end
  if *There are new blocks with flag=TRANSFER from C to S* **then**
    Update M_C using the data in S to generate the new model M_S and new error E_S;
    Set M = M_S;
    Create a transaction from S to S itself with flag = UPDATE, model = M_S, hash = HASH (M_S), and error = E_S;
    Wait for n-1 blocks with flag=EVALAUTE and collect all errors in the blockchain;
    if E_S *is not larger than all errors* **then**
      Identify the site L with the largest error E_L;
      Create a transaction from S to L with flag = TRANSFER, model = NULL, hash = HASH (M_S), and error = E_L;
    end
  end
end

Algorithm 3: Proof-Of-Information-Exit. This is the main algorithm for sites to exit.

Input : this site S
Output: The latest online machine learning model M

Create a transaction from S to S itself with flag = EXIT, model = NULL, hash = NULL, and error = NULL;
exit;

Discussion

The model of proof-of-information algorithm has to choose one site to update the model, so the sites in the private blockchain network has to synchronize. By adding entering and exiting information, other sites can automatically compute the number of sites in the blockchain network. We also compare the model chain with the bitcoin, the main difference is each node in the bitcoin works parallelly while each site in the modelchain each site works in a sequence order. By adding the entering and exiting information, we transform the work flow into a pseudo-parallel pattern which the model can continuously update regardless the entering and exiting of sites.

References

[1] https://bitcoin.org/bitcoin.pdf

[2] Tsung-Ting Kuo, PhD and Lucila Ohno-Machado, MD,PhD: ModelChain: Decentralized Privacy-Preserving Healthcare Predictive Modeling Framework on Private Blockchain Networks

[3] https://cointelegraph.com/news/blockchain-based-artificial-neural-networks-to-save-thousands-of-lives-from-medical-errors