Department of Computer Science and Engineering
CS6811 Final Year Project [2023]

A PRIVACY PRESERVING FEDERATED LEARNING PLATFORM SECURED WITH AUTHORITY CONSENSUS USING BLOCKCHAIN

Project Guide

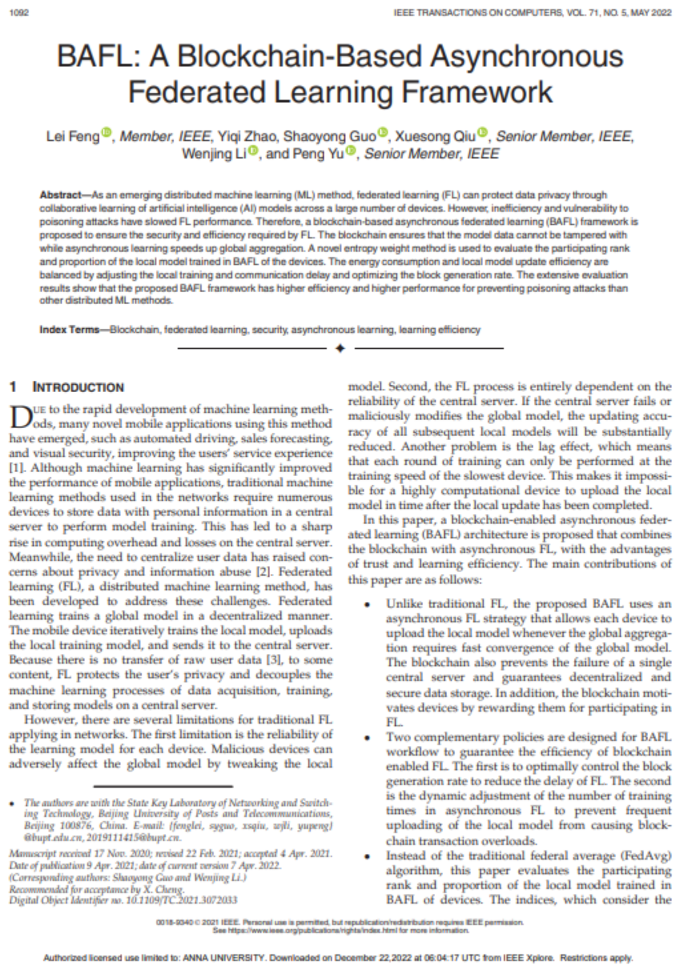
Dr. S. Sudha,
Professor, DCSE
College of Engineering,Guindy
Anna University

Team 17

2019103066 - Srivatsav R
2019103573 - Sachin Raghul T
2019103576 - Sanjeev KM

BASE PAPER

BAFL: A Blockchain-Based Asynchronous Federated Learning Framework



BAFL: A Blockchain-Based Asynchronous Federated Learning Framework

Lei Feng, *Member, IEEE*, Yiqi Zhao, Shaoyong Guo, Xuesong Qiu, *Senior Member, IEEE*, Wenjing Li, and Peng Yu, *Senior Member, IEEE*

Abstract—As an emerging distributed machine learning (ML) method, federated learning (FL) can protect data privacy through collaborative learning of artificial intelligence (AI) models across a large number of devices. However, inefficiency and vulnerability to poisoning attacks have slowed FL performance. Therefore, a blockchain-based asynchronous federated learning (BAFL) framework is proposed to ensure the security and efficiency required by FL. The blockchain ensures that the model data cannot be tampered with while asynchronous learning speeds up global aggregation. A novel entropy weight method is used to evaluate the participating rank and proportion of the local model trained in BAFL of the devices. The energy consumption and local model update efficiency are balanced by adjusting the local training and communication delay and optimizing the block generation rate. The extensive evaluation results show that the proposed BAFL framework has higher efficiency and higher performance for preventing poisoning attacks than other distributed ML methods.

Index Terms—Blockchain, federated learning, security, asynchronous learning, learning efficiency

1 Introduction

Due to the rapid development of machine learning methods, many novel mobile applications using this method have emerged, such as automated driving, sales forecasting, and visual security, improving the users' service experience [1]. Although machine learning has significantly improved the performance of mobile applications, traditional machine learning methods used in the networks require numerous devices to store data with personal information in a central server to perform model training. This has led to a sharp rise in computing overhead and losses on the central server. Meanwhile, the need to centralize user data has raised concerns about privacy and information abuse [2]. Federated learning (FL), a distributed machine learning method, has been developed to address these challenges. Federated learning trains a global model in a decentralized manner. The mobile device iteratively trains the local model, uploads the local training model, and sends it to the central server. Because there is no transfer of raw user data [3], to some content, FL protects the user's privacy and decouples the machine learning processes of data acquisition, training, and storing models on a central server.

However, there are several limitations for traditional FL applying in networks. The first limitation is the reliability of the learning model for each device. Malicious devices can adversely affect the global model by tweaking the local model. Second, the FL process is entirely dependent on the reliability of the central server. If the central server fails or maliciously modifies the global model, the updating accuracy of all subsequent local models will be substantially reduced. Another problem is the lag effect, which means that each round of training can only be performed at the training speed of the slowest device. This makes it impossible for a highly computational device to upload the local model in time after the local update has been completed.

In this paper, a blockchain-enabled asynchronous federated learning (BAFL) architecture is proposed that combines the blockchain with asynchronous FL, with the advantages of trust and learning efficiency. The main contributions of this paper are as follows:

- Unlike traditional FL, the proposed BAFL uses an asynchronous FL strategy that allows each device to upload the local model whenever the global aggregation requires fast convergence of the global model. The blockchain also prevents the failure of a single central server and guarantees decentralized and secure data storage. In addition, the blockchain motivates devices by rewarding them for participating in FL.
- Two complementary policies are designed for BAFL workflow to guarantee the efficiency of blockchain enabled FL. The first is to optimally control the block generation rate to reduce the delay of FL. The second is the dynamic adjustment of the number of training times in asynchronous FL to prevent frequent uploading of the local model from causing blockchain transaction overloads.
- Instead of the traditional federal average (FedAvg) algorithm, this paper evaluates the participating rank and proportion of the local model trained in BAFL of devices. The indices, which consider the

- *The authors are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China. E-mail: {fenglei, syguo, xsqiu, wjli, yupeng}@bupt.edu.cn, 2019111415@bupt.cn.*

*Manuscript received 17 Nov. 2020; revised 22 Feb. 2021; accepted 4 Apr. 2021. Date of publication 9 Apr. 2021; date of current version 7 Apr. 2022.
(Corresponding authors: Shaoyong Guo and Wenjing Li.)
Recommended for acceptance by X. Cheng.
Digital Object Identifier no. 10.1109/TC.2021.3072033*

INTRODUCTION

- In recent years, there has been an enormous corpus of research and development dedicated to and focused on accelerating the processes of machine learning in every conceivable shape or form.

- This provides greater flexibility and interoperability features than Monolith as they are easily scalable and robust to failures.
- Computer hardware was not adequate enough to handle already existing machine learning algorithms, yet it seems the rapidly growing demand for machine learning applications has put us in a situation that yet again undermines the capabilities of individual machines.
- The Solution for this problem is to multiple machines collaborate on training the model at hand.

INTRODUCTION

- The idea of federated learning solves another major problem: the scarcity of data at a single machine. This is solved through the collaboration of different nodes in the training, each using its local data then sharing its model updates, realizing a single model trained on the sum of their local data.

- In Federated Learning, they train it on their private data, then summarize and encrypt the model's new configuration. The model updates are sent back to the cloud, decrypted, averaged, and integrated into the centralized model. Iteration after iteration, the collaborative training continues until the model is fully trained.

- DFedL offers feasible solutions to common challenges in decentralized federated learning, namely, privacy, efficiency, and Byzantine fault-tolerance.

OVERALL OBJECTIVES

- To achieve federated learning over decentralized network which eliminates following issues:
  - Central node inferring important information
  - Single point of failure on the server node
- To prevent malicious clients, who tries to attack the global model with incorrect gradients (Byzantine Imposter)
- To solve the issue of expensive communication where each training node shares its update with all the other nodes using authority consensus

LITERATURE SURVEY

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
1	Federated Learning System without Model Sharing through Integration of Dimensional Reduced Data Representations (2020)	Explored an federated learning system that enables integration of dimensionality reduced representations of distributed data prior to a supervised learning task, thus avoiding model sharing among the parties.	The Results show that this approach can achieve similar accuracy as Federated Averaging and performs better than Federated Averaging in a small-user setting.	Dimensionlity reduction may reduce accuracy in some cases

LITERATURE SURVEY

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
2	Trustless Machine Learning Contracts; Evaluating and Exchanging Machine Learning Models on the Ethereum Blockchain (2018)	This paper discovered that it is possible to create contracts that offer a reward in exchange for a trained machine learning model for a particular data set. This would allow users to train machine learning models for a reward in a trustless manner. Contracts can be created easily by anyone with a dataset, even programmatically by software agents.	This will incentivize the creation of better machine learning models, and make AI more accessible to companies and software agents.	Without better design it leads to increase in gas cost

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
3	Comparative Study of Byzantine Fault Tolerant Consensus Algorithms on Permissioned Blockchains (2020)	This paper provides idea for the private blockchains to use permissioned blockchain consensus algorithms as the participants need the permission of the authority to be able to join the system and investigated permissioned and permissionless blockchain,	Trust models between the nodes, incentives, number of nodes & parties involved, and scalability regarding the number of transactions.	Each business application has different problems and priorities which is hard to generalise.
4	Federated Learning: Strategies for Improving Communication Efficiency (2017)	This paper proposed two ways to reduce the uplink communication costs in implementing federated learning: structured updates and sketched updates	Both convolutional and recurrent networks show that the proposed methods can reduce the communication cost by two orders of magnitude.	Reducing communication cost may lead to downside of security.

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
5	Consensus-Based Data-Privacy Preserving Data Aggregation (2019)	This paper addresses to solve the challenges in privacy preserving data aggregation by exploiting the distributed consensus technique. They first proposed a secure consensus-based DA algorithm that guarantees an accurate sum aggregation while preserving the privacy of sensitive data, then proved that the proposed algorithm converges accurately.	Extensive simulations have shown that the proposed algorithm has high accuracy and low complexity, and they are robust against network dynamics.	Intensive simulation may lead to low accuracy and high complexity
6	A PoW-less Bitcoin with Certified Byzantine Consensus (2022)	This paper explored one possible way to solve the problem where the Distributed Ledger Technologies, when managed by a few trusted checkers requires most but not all of the machinery available in public DLTs.	The combined protocol may operate as a modern,safe foundation for digital payment systems and Central Bank Digital Currencies.	PoA is more centralised compared to PoW

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
7	Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data (2020)	This paper investigated the effects of data distribution across collaborating institutions on model quality and learning patterns, indicating that increased access to data through data private multi-institutional collaborations can benefit model quality more than the errors introduced by the collaborative method.	Models having a catalytic impact towards precision/personalized medicine.	Errors may occur if done in small scale
8	Privacy-Preserving Resource Sharing Using Permissioned Blockchains, (2021)	Designed a decentralized resource sharing platform that uses a permissioned blockchain to allow users to share their digital items with their specified attributed-based access policies that are enforced through a set of smart contracts, and removes the need for a trusted intermediary.	This decentralized attribute-based access control system achieves the same functionality while preserving the privacy of user's access	It has limited availability, has access to the resource requires it's owner to be online.

SNO	TITLE	METHODOLOGY	ADVANTAGES	LIMITATIONS
9	A Hybrid Approach to Privacy-Preserving Federated Learning (2019)	This paper proposed a Hybrid Approach to Privacy-Preserving Federated Learning by Combining differential privacy with secure multiparty computation which enables to reduce the growth of noise injection as the number of parties increases without sacrificing privacy while maintaining a pre-defined rate of trust.	This system can be used to train a variety of machine learning models, which demonstrate that our approach out-performs state of the art solutions.	Hybrid approach leads to moderate privacy and security
10	Blockchain-enabled Federated Learning: A Survey (2023)	Introduced a Blockchain-enabled Federated Learning that has been and will keep generating widespread attention in this big data era. However, the existing paradigms become increasingly impractical and difficult to follow from both perspectives of introductory and in-depth exploration of state-of-the-art models.	Systematically introduced the evaluation matrix and analysis criteria	Lacking consensus algorithm as it may be prone to attacks

SUMMARY OF ISSUES

- Each business application has different problems and priorities which is hard to generalize.

- Reducing communication costs may lead to downside of security.

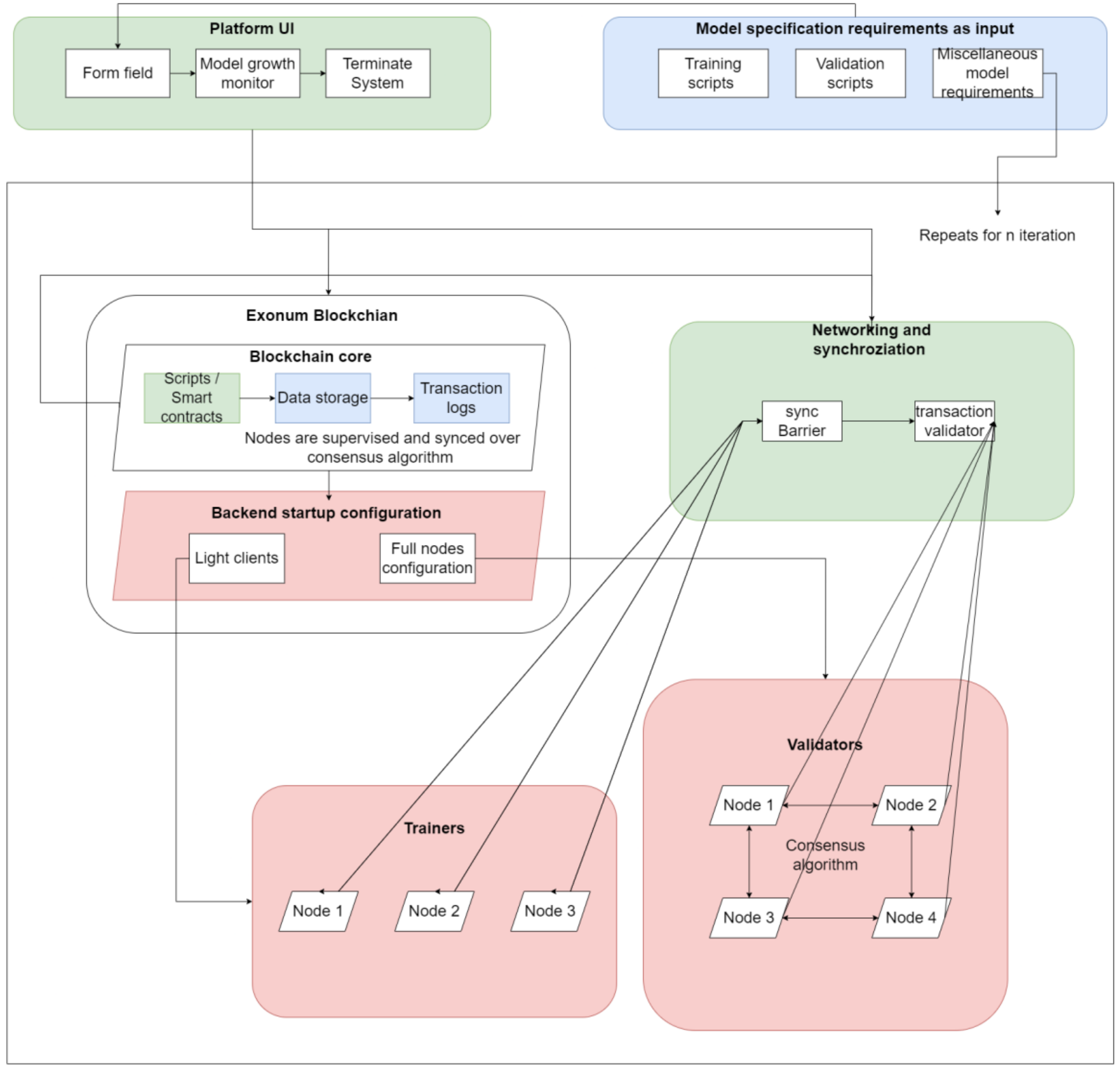
- Intensive simulation may lead to low accuracy and high complexity.

- Proof of Authority is more centralized when compared with Proof of Work.

PROPOSED SYSTEM

- We propose a framework Decentralized Federated Learning (DFedL) for building a generic decentralized federated learning system using the blockchain technology, accommodating any machine learning model that is compatible with gradient descent optimization.
- DFedL uses proof-of-concept used to spawn mini-systems – checkers and clients. The system design comprising of two decentralized actors: client and checker, alongside the methodology for ensuring reliable and efficient operation of the system.
- Hence, DFedL as an experimental sandbox to compare and contrast the effects of client-to-checker ratio, reward-penalty policy, and model synchronization schemes on the overall system performance, ultimately showing by example that a decentralized federated learning system is indeed a feasible alternative to more centralized architectures.

ARCHITECTURE DIAGRAM

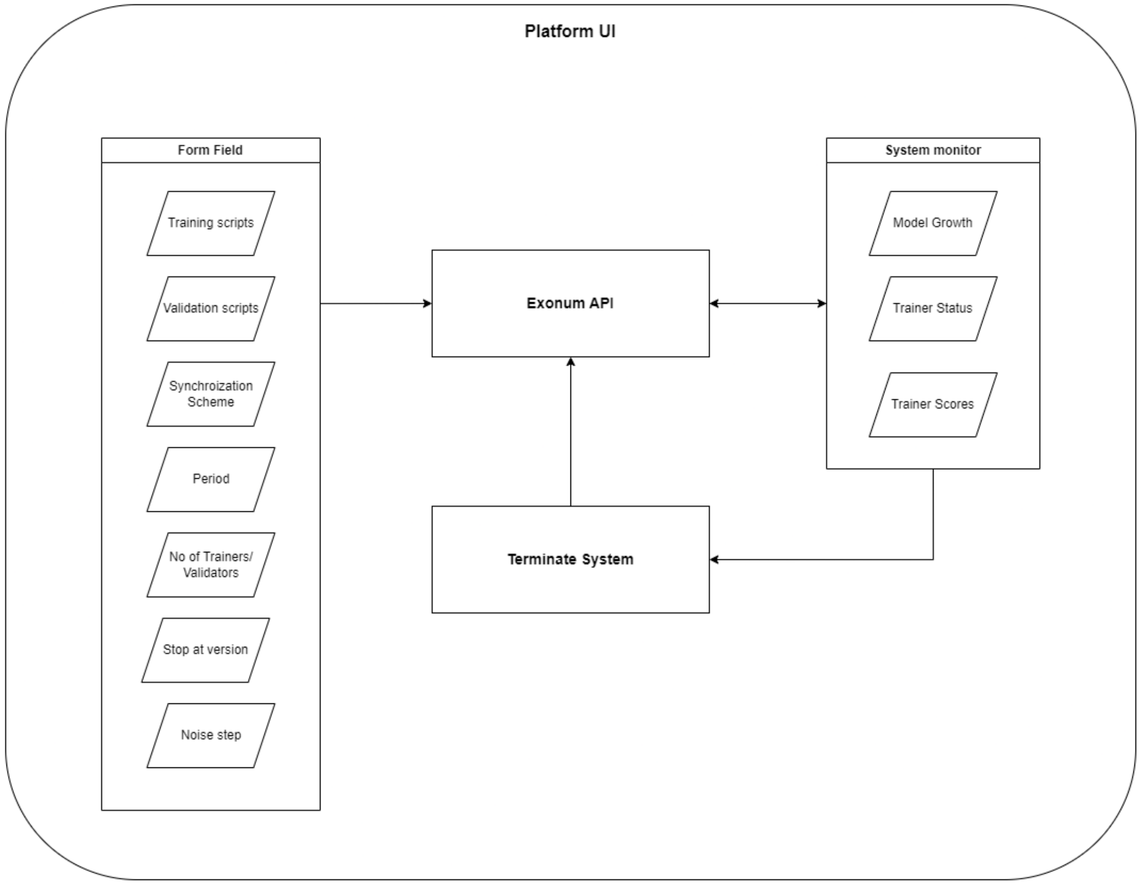


Platform UI
Form field
Model growth monitor
Terminate System
Model specification requirements as input
Training scripts
Validation scripts
Miscellaneous model requirements
Repeats for n iteration
Exonum Blockchian
Blockchain core
Scripts / Smart contracts
Data storage
Transaction logs
Nodes are supervised and synced over consensus algorithm
Backend startup configuration
Light clients
Full nodes configuration
Networking and synchroziation
sync Barrier
transaction validator
Validators
Node 1
Node 2
Consensus algorithm
Node 3
Node 4
Trainers
Node 1
Node 2
Node 3

LIST OF MODULES

1) Model Specification and platform UI
2) Blockchain core and startup configuration
3) Networking and synchronization
4) Model clients
5) Model checkers

Module 1 - Model Specification and platform UI



Module 1 - Model Specification and platform UI

INPUT: Training and validation Scripts and Number, Synchronization scheme, Noise step
OUTPUT: Forwards to startup configuration.

STEPS:

1)Develop application to get the scripts to distribute them across clients and checkers to train the local dataset upon it.

2) User can configure the system to run model

3) Configuration can be done on

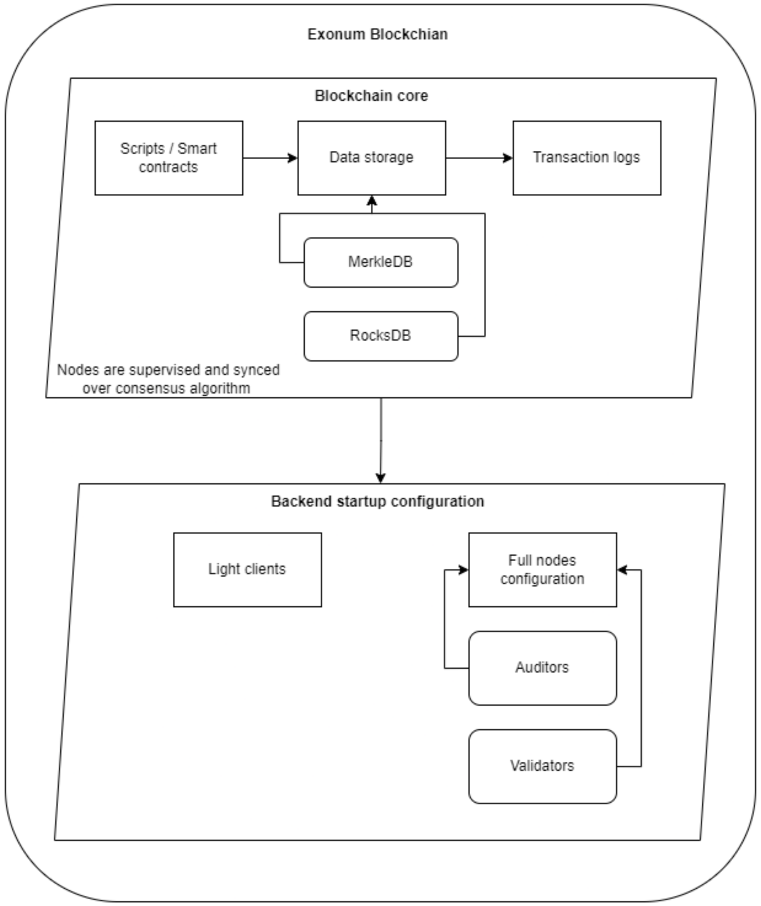
- fixing amount of clients and checkers,
- choice on Synchronization Scheme (BAP, BSP, SSP),
- Stop at version
- Noise step

3) To monitor the model growth system is equipped with client status and client scores

4) Creating the jobs and start training

5) Terminating thre process

Module 2 - Blockchain core and startup configuration



Module 2 - Blockchain core and startup configuration (1/2)

INPUT: Parameters for Environment Creation
OUTPUT: checkers and clients Environment

1) Data storage, which contains data structured into objects. Objects represent high-level wrappers over the key-value store.

2) Transaction log, i.e., the complete history of all transactions ever applied to the data storage.

3) Exonum gathers transactions into blocks; the whole block is approved atomically.

4) After a block is approved, every transaction in it is executed sequentially, with changes applied to the data storage.

5) Smart contracts define business logic of the blockchain, allow to retrieve data from the blockchain, and can be reused across different projects.

Module 2 - Blockchain core and startup configuration (2/2)

STEPS:

1) Full nodes replicate the entire contents of the blockchain and correspond to replicas in distributed databases. All the full nodes are authenticated with public-key cryptography.

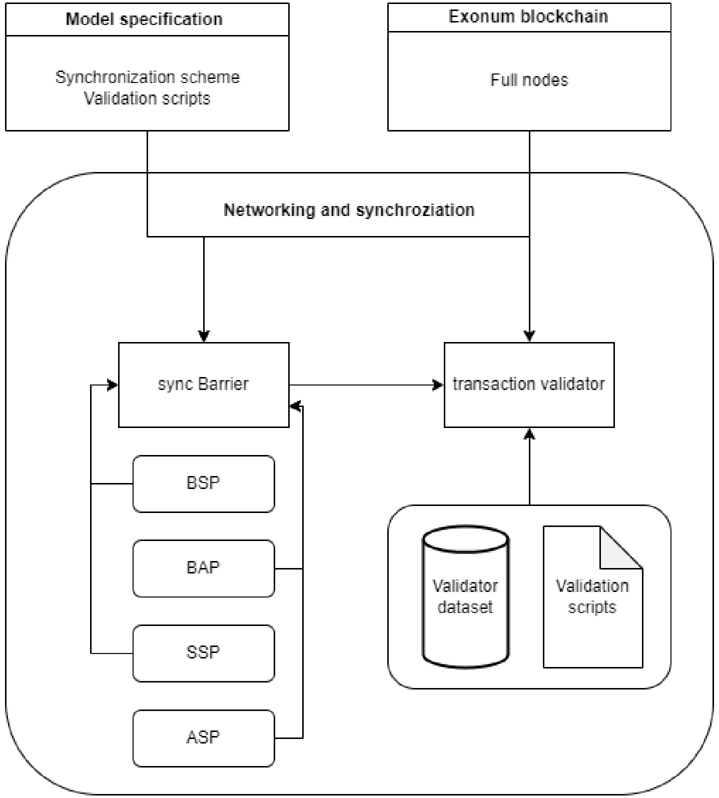
2) Full nodes are further subdivided into 2 categories Auditors and checkers

3) Auditors replicate the entire content of the blockchain. They can generate new transactions, but cannot choose which transactions should be committed

4) Only checkers can generate new blocks by using a Byzantine fault tolerant consensus algorithm. checkers receive transactions, verify them, and include into a new block.

5) Light clients represent clients in the client-server paradigm; they connect to full nodes to retrieve information from the blockchain they are interested in and to send transactions.

Module 3 - Networking and synchronization

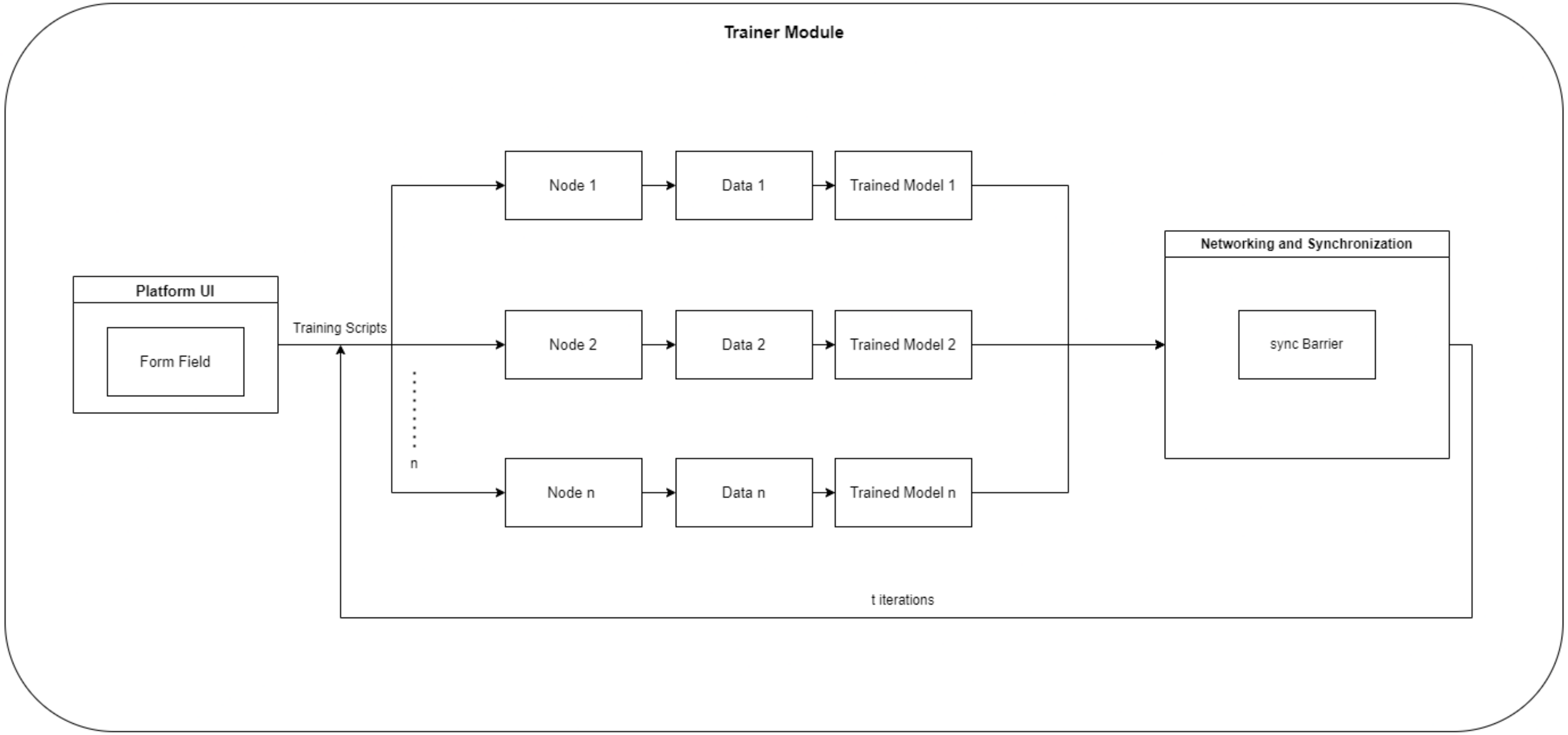


Module 3 - Elastic Scalability (2/2)

INPUT: Sync updated parameters from clients and validates with checkers
OUTPUT: Commits the transaction logs in exonum blockchain

1. There are many techniques to handle node update synchronization to prevent model degradation
2. In Bulk Synchronous Parallel (BSP) each training round is limited by a fixed period that ends with a strict deadline after which no further client updates are considered for that round.
3. In Stale Synchronous Parallel (SSP) SSP is modelled as BSP with the possibility of a deadline extension when proportion to the ratio of clients that have not yet finished training for that round
4. Barrierless Asynchronous Parallel (BAP) BSP as it almost totally eliminates the cost of synchronization by allowing wait less, asynchronous communication between nodes
5. Trust scores are adjusted based on validation results.
6. Upon receiving a client update, a checker adds its gradients to the latest model weights, and computes its validation score.

Module 4 - Model clients

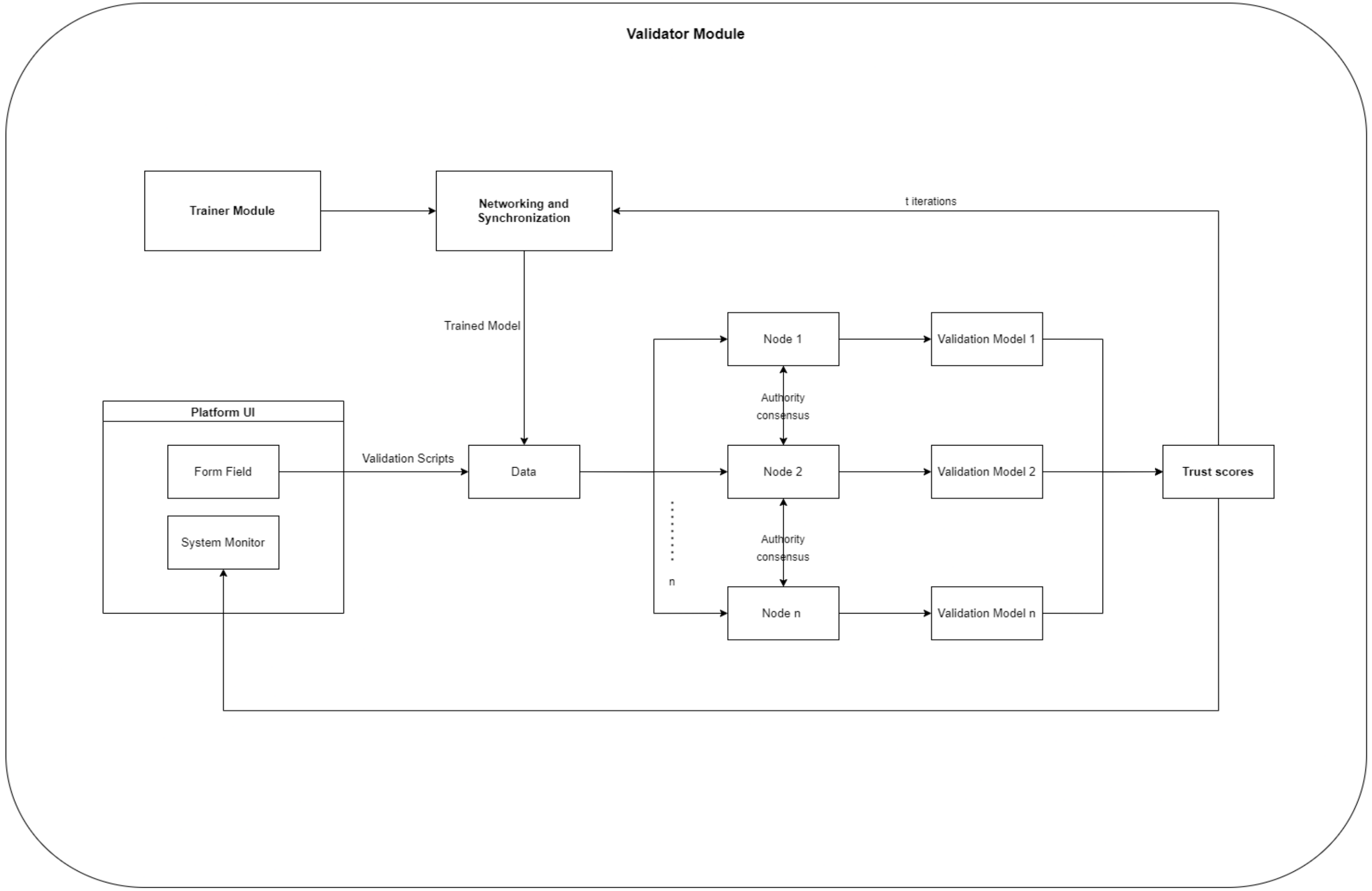


Module 4 - Model clients

INPUT: client scripts
OUTPUT: Updated model parameters

- Performs model training including all needed intermediate transformations such as model flattening and rebuilding at the client side.
- This layer sends and receives flattened models and flattened gradients built on top of exonum lightweight clients.
- To improve system auditability, Exonum includes a light client. Light client is a JavaScript library with a number of helper functions available for use by frontend developers.
- There are two typical use cases for the light client forming and sending transactions to the Exonum blockchain network
- Forming requests to full nodes of the network (normally, HTTP GET requests) and validation of the responses

Module 5 - Model checkers



Module 5 - Model checkers

INPUT: Updated model parameters and client scripts.
OUTPUT: Trust scores

- checkers are responsible for maintaining models on the underlying blockchain network through decentralized consensus based on voting.
- Ideally, a checker's vote, on whether some update to the model (made by a client node) is acceptable, should be decided by validation on data presumably unrevealed to the client.
- That, combined with the voting-based nature of consensus, stipulates the following requirements and assumptions in order for the system to operate soundly.
- One important property is that clients are not fully connected to the checkers while checkers are fully interconnected.
- yet a node can be a checker on a subnetwork while being a client for another. Further, each model subnetwork requires at least one checker to function.

IMPLEMENTATION DETAILS

Blockchain Setup (Exonum):

Exonum is written in Rust and depends on the following third-party system libraries:

- RocksDB (persistent storage)
- libsodium (cryptography engine)
- Protocol Buffers (mechanism for serializing structured data)

Prerequisites:

- git
- Node.js with npm
- Rust compiler
- Exonum launcher

Dependencies installation:

For distributives with deb-based package managers (such as Debian or Ubuntu), use

```
add-apt-repository ppa:exonum/rocksdb
apt-get update
apt-get install build-essential jq libsodium-dev
libsnappy-dev libssl-dev \
    librocksdb6.2 pkg-config clang-7 lldb-7 lld-7
```

For `protobuf` installation add the following dependencies:

```
add-apt-repository ppa:maarten-fonville/protobuf
apt install libprotobuf-dev protobuf-compiler
```

Package names and installation methods may differ in other Linux distributives; use package manager tools to locate and install dependencies.

Depending on the version of your distributive, libsodium, RocksDB and Protobuf may not be present in the default package lists. In this case you may need to install these packages from third-party PPAs, or build them from sources.

Adding Environment Variables

If your OS contains pre-compiled rocksdb or snappy libraries, you may setup ROCKSDB_LIB_DIR and/or SNAPPY_LIB_DIR environment variable to point to a directory with these libraries. This will significantly reduce compile time.

```
export ROCKSDB_LIB_DIR=/usr/lib
export SNAPPY_LIB_DIR=/usr/lib/x86_64-linux-gnu
```

Rust Toolchain

Exonum repositories use the stable Rust toolchain that can be installed by using the **rustup** program:

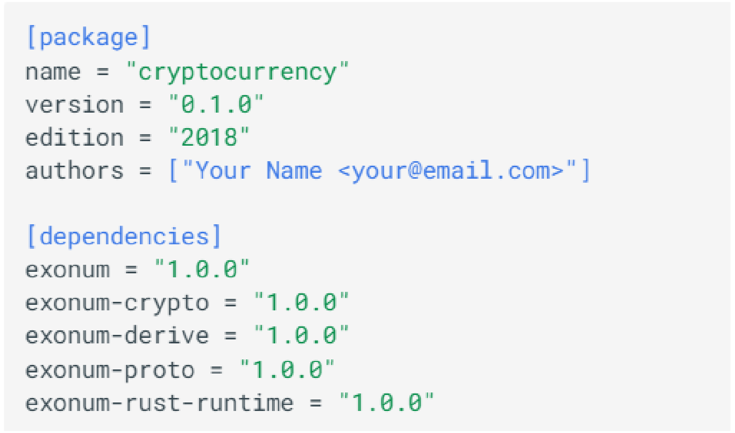
```
curl https://sh.rustup.rs -sSf | sh -s -- --default-
toolchain stable
```

Compiling Exonum

We can verify that you installed dependencies and the Rust toolchain correctly by cloning the `exonum` repository and running its built-in unit test suite:

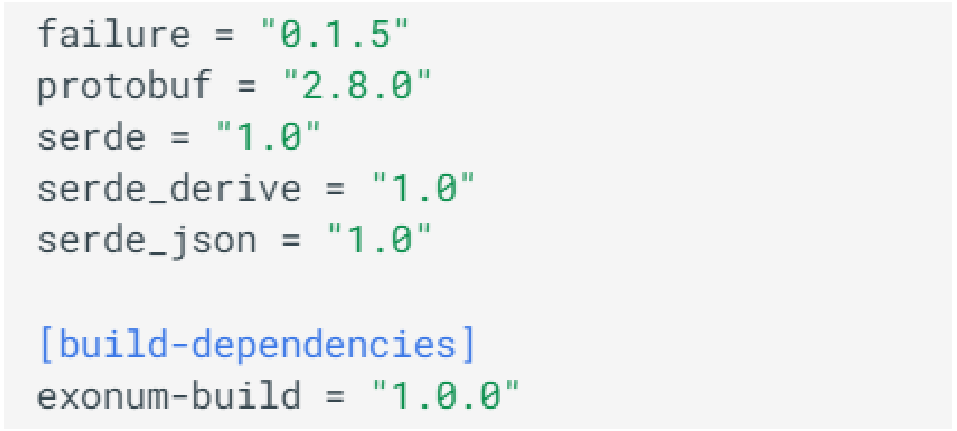
```
git clone https://github.com/exonum/exonum.git
cd exonum
cargo test -p exonum
```

Necessary dependencies(Cargo.toml):



```
[package]
name = "cryptocurrency"
version = "0.1.0"
edition = "2018"
authors = ["Your Name <your@email.com>"]

[dependencies]
exonum = "1.0.0"
exonum-crypto = "1.0.0"
exonum-derive = "1.0.0"
exonum-proto = "1.0.0"
exonum-rust-runtime = "1.0.0"
```

NECESSARY DEPENDENCIES



```
failure = "0.1.5"
protobuf = "2.8.0"
serde = "1.0"
serde_derive = "1.0"
serde_json = "1.0"

[build-dependencies]
exonum-build = "1.0.0"
```

NECESSARY DEPENDENCIES

Generate node configuration template:

```
mkdir example
exonum-<project-name> generate-template \
  example/common.toml \
  --checkers-count 4
```

Generate public and secret keys for each node:

```
exonum-<project-name> -advanced generate-config \
  example/common.toml example/1 \
  --peer-address 127.0.0.1:6331 -n
exonum-<project-name> generate-config \
  example/common.toml example/2 \
  --peer-address 127.0.0.1:6332 -n
exonum-<project-name> generate-config \
  example/common.toml example/3 \
  --peer-address 127.0.0.1:6333 -n
exonum-<project-name> generate-config \
  example/common.toml example/4 \
  --peer-address 127.0.0.1:6334 -n
```

Finalize configs:

```
exonum-<project-name> finalize \
  --public-api-address 0.0.0.0:8200 \
  --private-api-address 0.0.0.0:8091 \
  example/1/sec.toml example/1/node.toml \
  --public-configs example/{1,2,3,4}/pub.toml
```

Similar commands for other 3 nodes, with adjusted paths and socket addresses

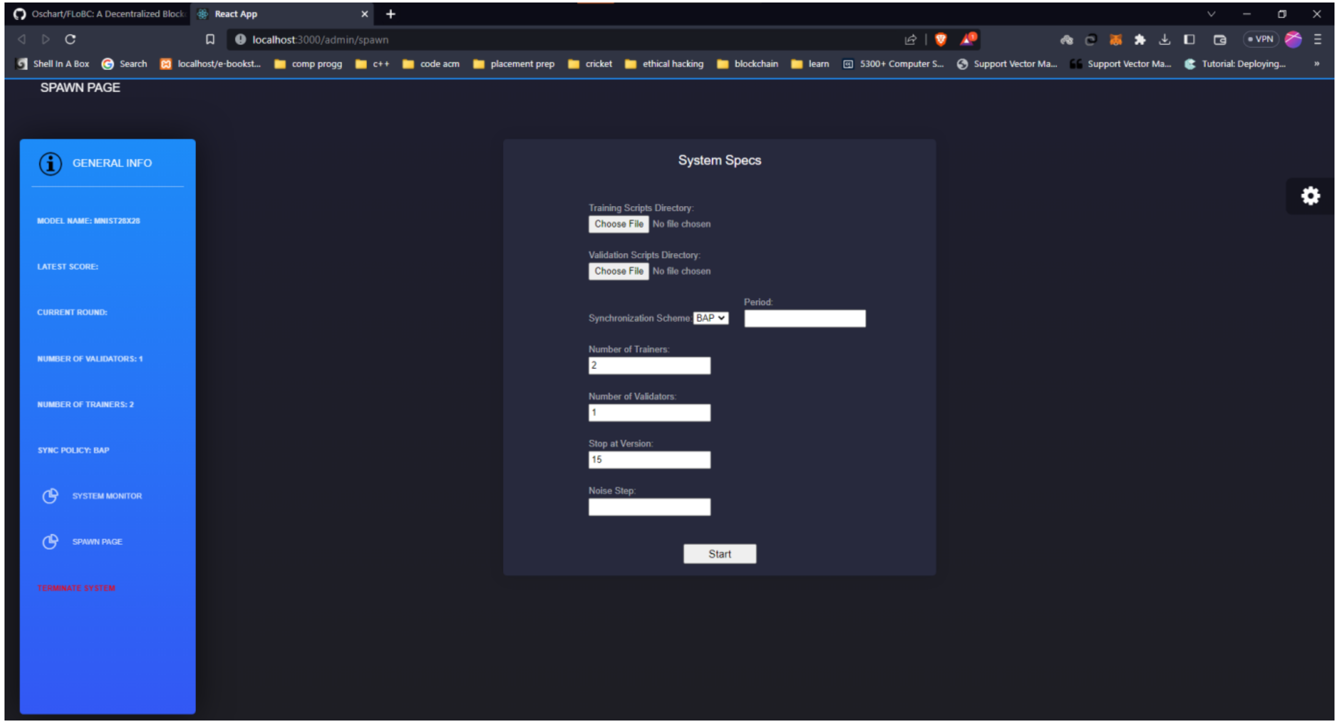
Run nodes:

```
exonum--<project-name> run \
  --node-config example/1/node.toml \
  --db-path example/1/db \
  --public-api-address 0.0.0.0:8200 \
  --master-key-pass pass
```

Similar commands for other 3 nodes, with adjusted paths and socket addresses

Filling Requirements in Spawn Page :

Frontend is connected to node server via REST api. All the requirements to spawn the system is passed to node server which will run script to initiate client and checker nodes



MODEL NAME: MNIST28X28

LATEST SCORE:

CURRENT ROUND:

NUMBER OF VALIDATORS: 1

NUMBER OF TRAINERS: 2

SYNC POLICY: BAP

Training Scripts Directory: No file chosen

Validation Scripts Directory: No file chosen

Synchronization Scheme: BAP

Period:

Number of Trainers: 2

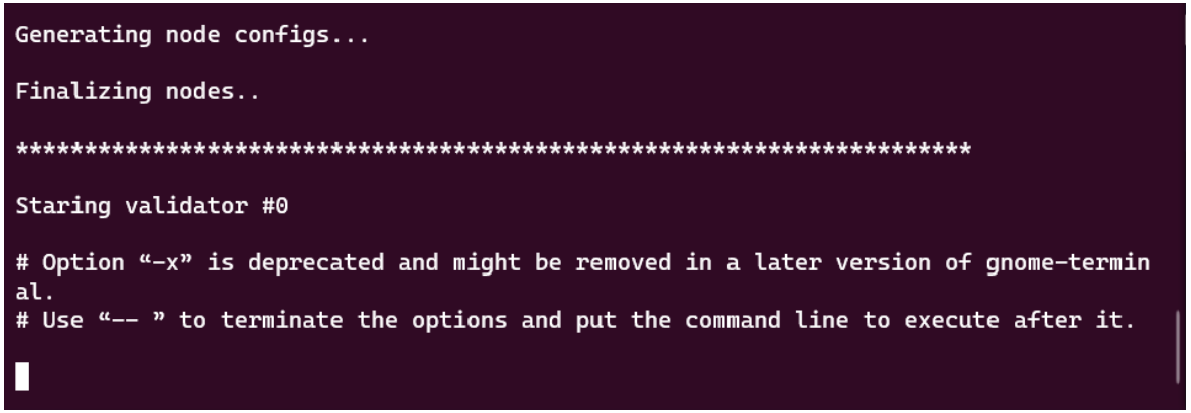
Number of Validators: 1

Stop at Version: 15

Noise Step:

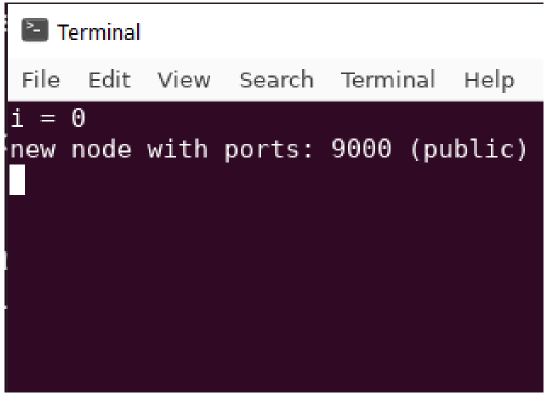
Starting checker Node :

Backend node spawn script starts the mentioned number of checkers and the checker nodes starts receiving for model updates by the clients



```
Generating node configs...

Finalizing nodes..

**********************************************************************

Staring validator #0

# Option “-x” is deprecated and might be removed in a later version of gnome-termin
al.
# Use “-- ” to terminate the options and put the command line to execute after it.
```

SPAWN SCRIPT BOOTING UP THE checkers



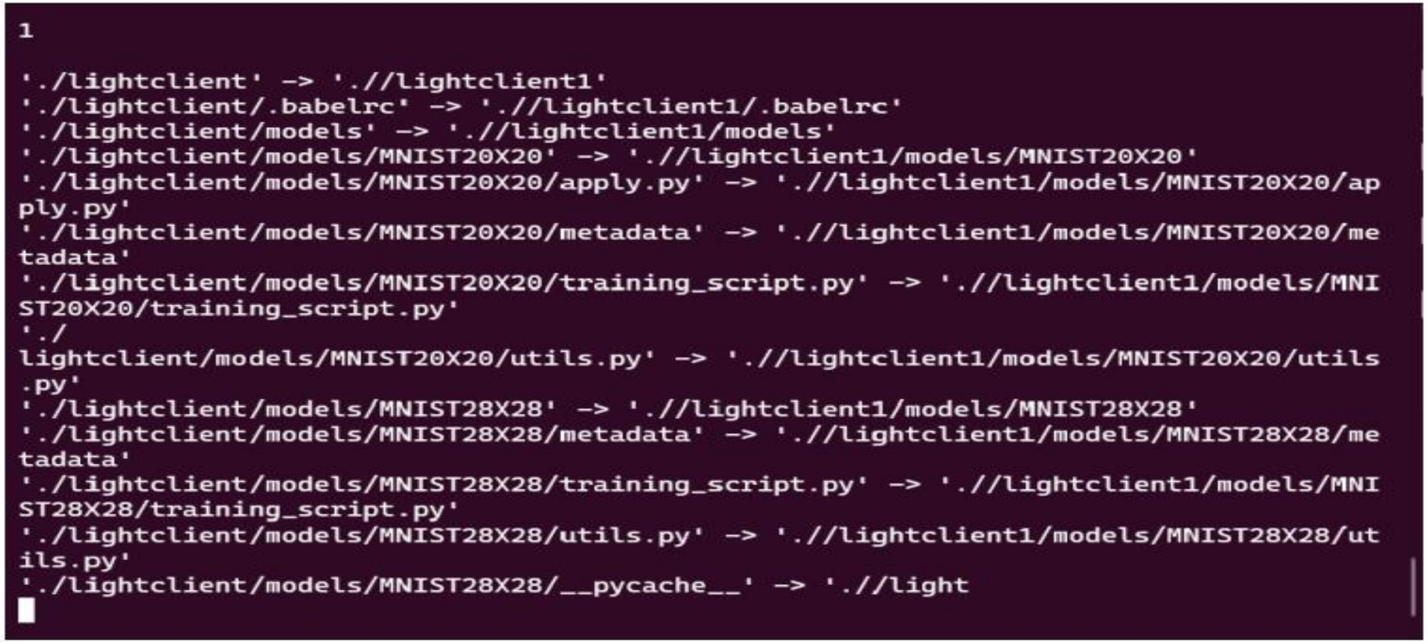
```
i = 0
new node with ports: 9000 (public)
```

checker WAITING FOR MODEL UPDATE BY clients

Starting checker Node :

Miners/clients – Light Clients (in Exonum)

Backend node spawn script starts the mentioned number of clients that trains the model with local data and sends it to the checker node

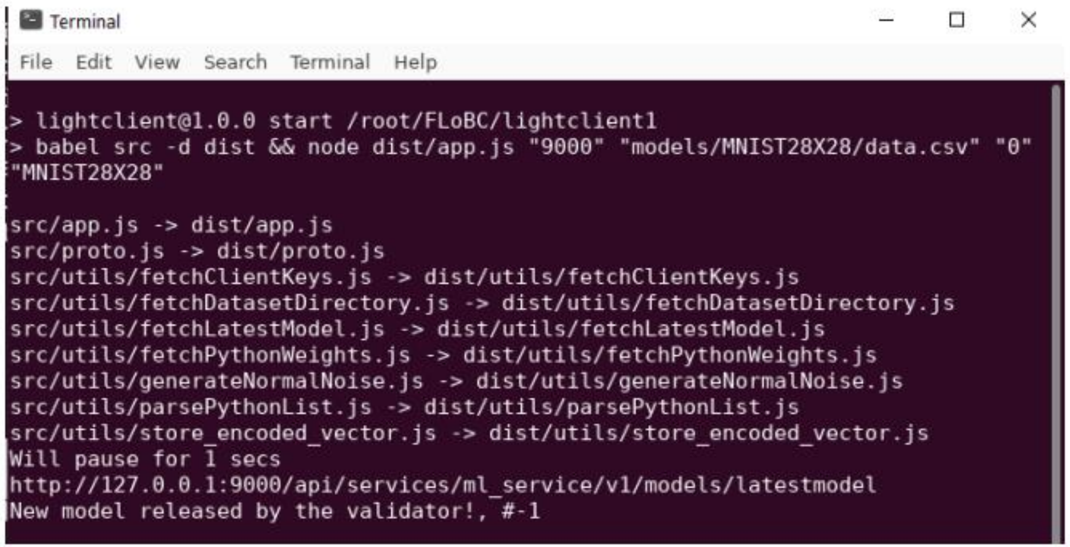


```
1

'./lightclient' -> './/lightclient1'
'./lightclient/.babelrc' -> './/lightclient1/.babelrc'
'./lightclient/models' -> './/lightclient1/models'
'./lightclient/models/MNIST20X20' -> './/lightclient1/models/MNIST20X20'
'./lightclient/models/MNIST20X20/apply.py' -> './/lightclient1/models/MNIST20X20/ap
ply.py'
'./lightclient/models/MNIST20X20/metadata' -> './/lightclient1/models/MNIST20X20/me
tadata'
'./lightclient/models/MNIST20X20/training_script.py' -> './/lightclient1/models/MNI
ST20X20/training_script.py'
'./
lightclient/models/MNIST20X20/utils.py' -> './/lightclient1/models/MNIST20X20/utils
.py'
'./lightclient/models/MNIST28X28' -> './/lightclient1/models/MNIST28X28'
'./lightclient/models/MNIST28X28/metadata' -> './/lightclient1/models/MNIST28X28/me
tadata'
'./lightclient/models/MNIST28X28/training_script.py' -> './/lightclient1/models/MNI
ST28X28/training_script.py'
'./lightclient/models/MNIST28X28/utils.py' -> './/lightclient1/models/MNIST28X28/ut
ils.py'
'./lightclient/models/MNIST28X28/__pycache__' -> './/light
```

SPAWN SCRIPT BOOTING UP THE LIGHT CLIENT (clients)

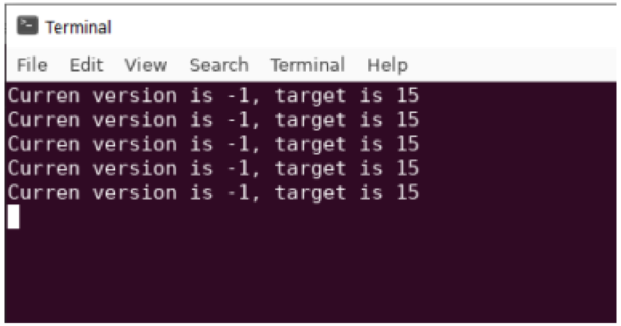
Starting checker Node :

Version manager is used to manage the current version of the released model with the target version



```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 1 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
```

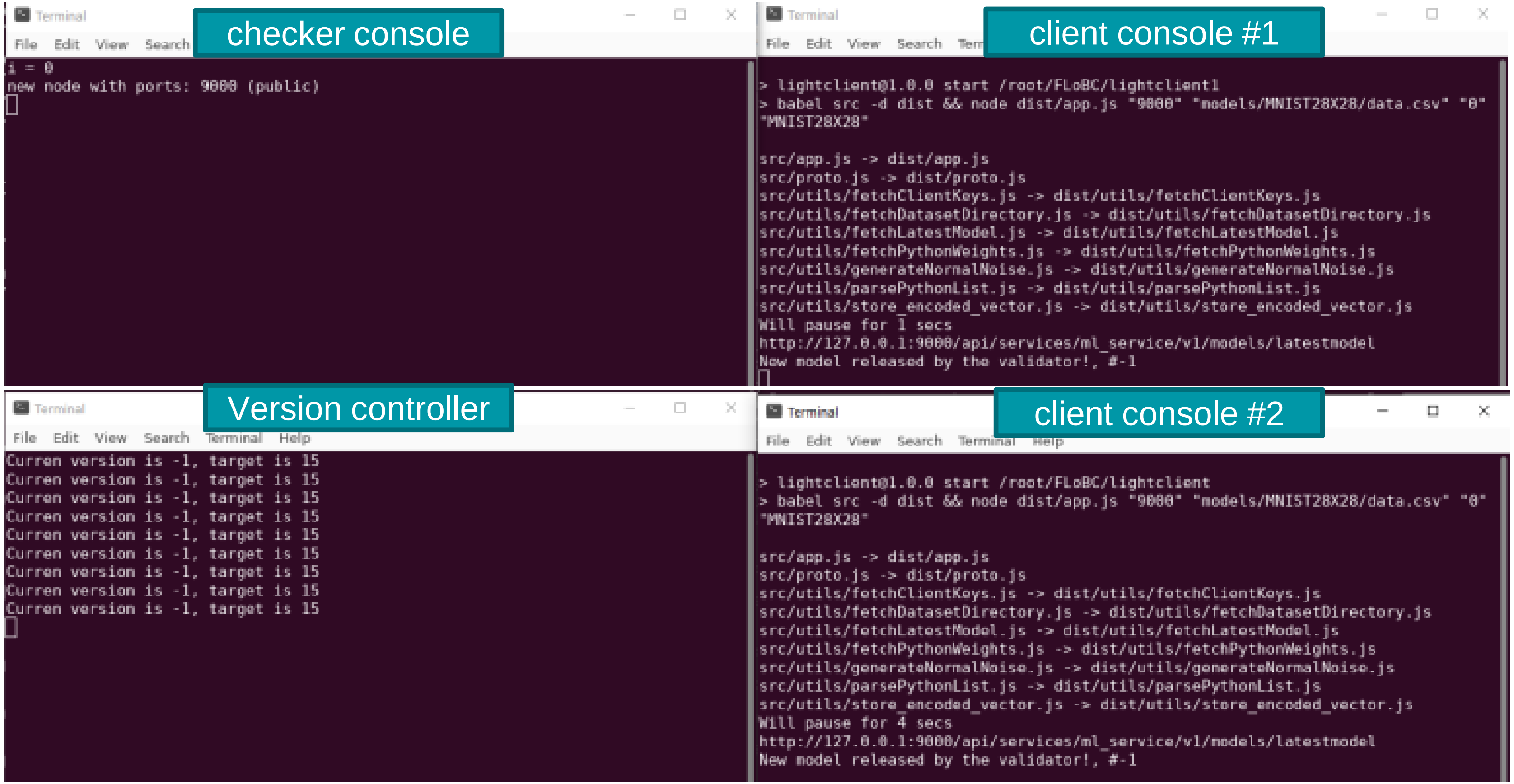
LIGHT CLIENT #1



```
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
```

VERSION MANAGER

Due to CPU restrictions, we have done the demo for 2 clients and 1 checker. Further this can be extended with the help of powerful CPUs



```
i = 0
new node with ports: 9000 (public)
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0" "MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 1 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
```

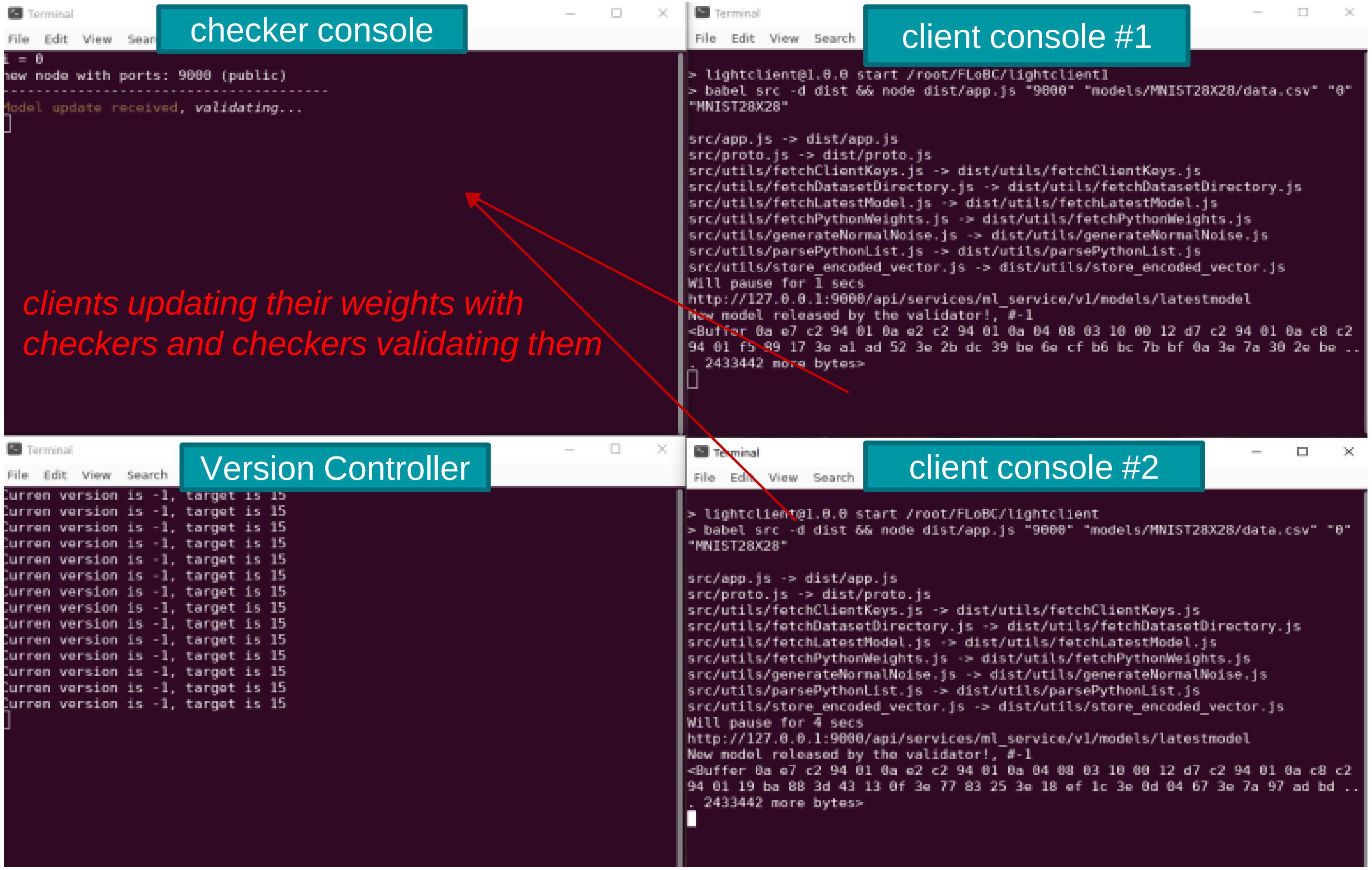
```
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0" "MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
```

CONSOLIDATED BACKEND ENVIRONMENT

Model Released by client :

client trains the model with local data and shares the model parameters with the checkers, asynchronously. clients are allowed to train for as long as a round may extend; that is, until a new model is released, clients can keep advancing local training, periodically sharing it with a checker. A new model is released when a minimum ratio of labor has been submitted, and that’s when clients should pull in the new model.



```
i = 0
new node with ports: 9000 (public)
----------------------------------------
Model update received, validating...
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 1 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 f5 89 17 3e a1 ad 52 3e 2b dc 39 be 6e cf b6 bc 7b bf 0a 3e 7a 30 2e be ..
. 2433442 more bytes>
```

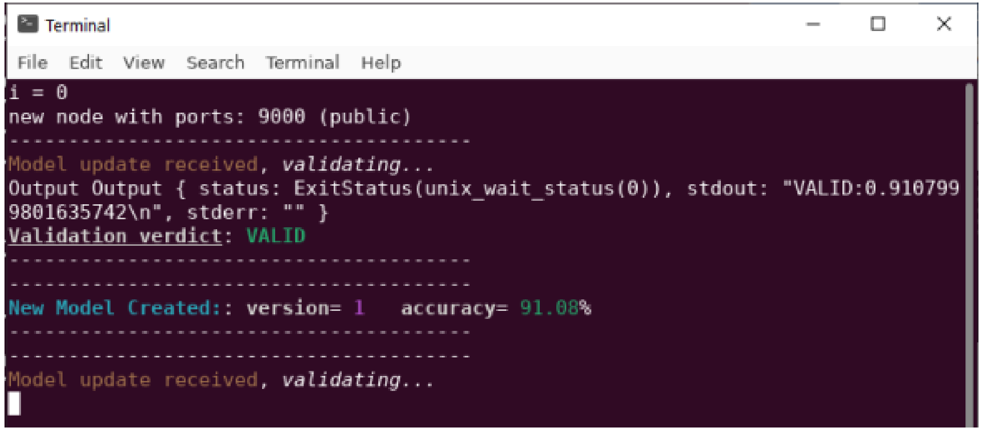
```
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 19 ba 88 3d 43 13 0f 3e 77 83 25 3e 18 ef 1c 3e 0d 04 67 3e 7a 97 ad bd ..
. 2433442 more bytes>
```

TRAINER RELEASING THE MODEL

Validating Model :

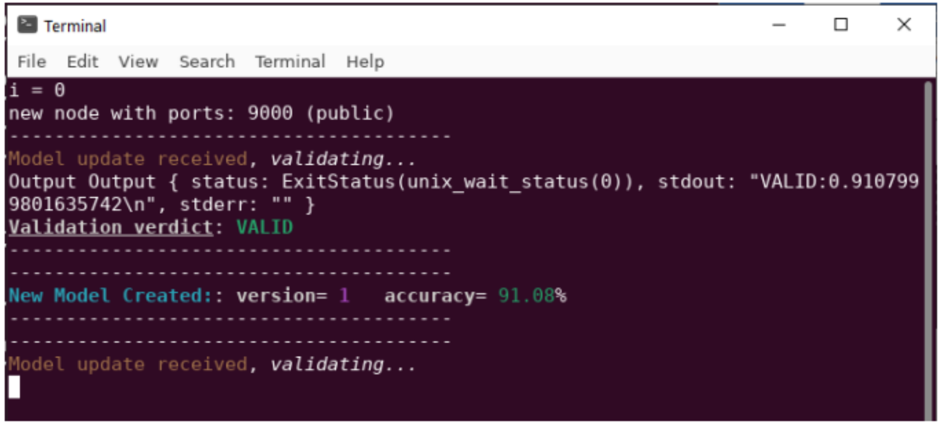
The model released by the client is validated by the checker using validating sample to check for any Byzantine attack. If the model accuracy is greater than the threshold accuracy then new model version is released by aggregating the previous version with the current version using aggregator function. Then the new aggregated model is synced across other checkers and clients. If the current version reaches the target version then the model is available for download.



```
i = 0
new node with ports: 9000 (public)
----------------------------------------
Model update received, validating...
Output Output { status: ExitStatus(unix_wait_status(0)), stdout: "VALID:0.910799
9801635742\n", stderr: "" }
Validation verdict: VALID
----------------------------------------
----------------------------------------
New Model Created:: version= 1   accuracy= 91.08%
----------------------------------------
----------------------------------------
Model update received, validating...
```

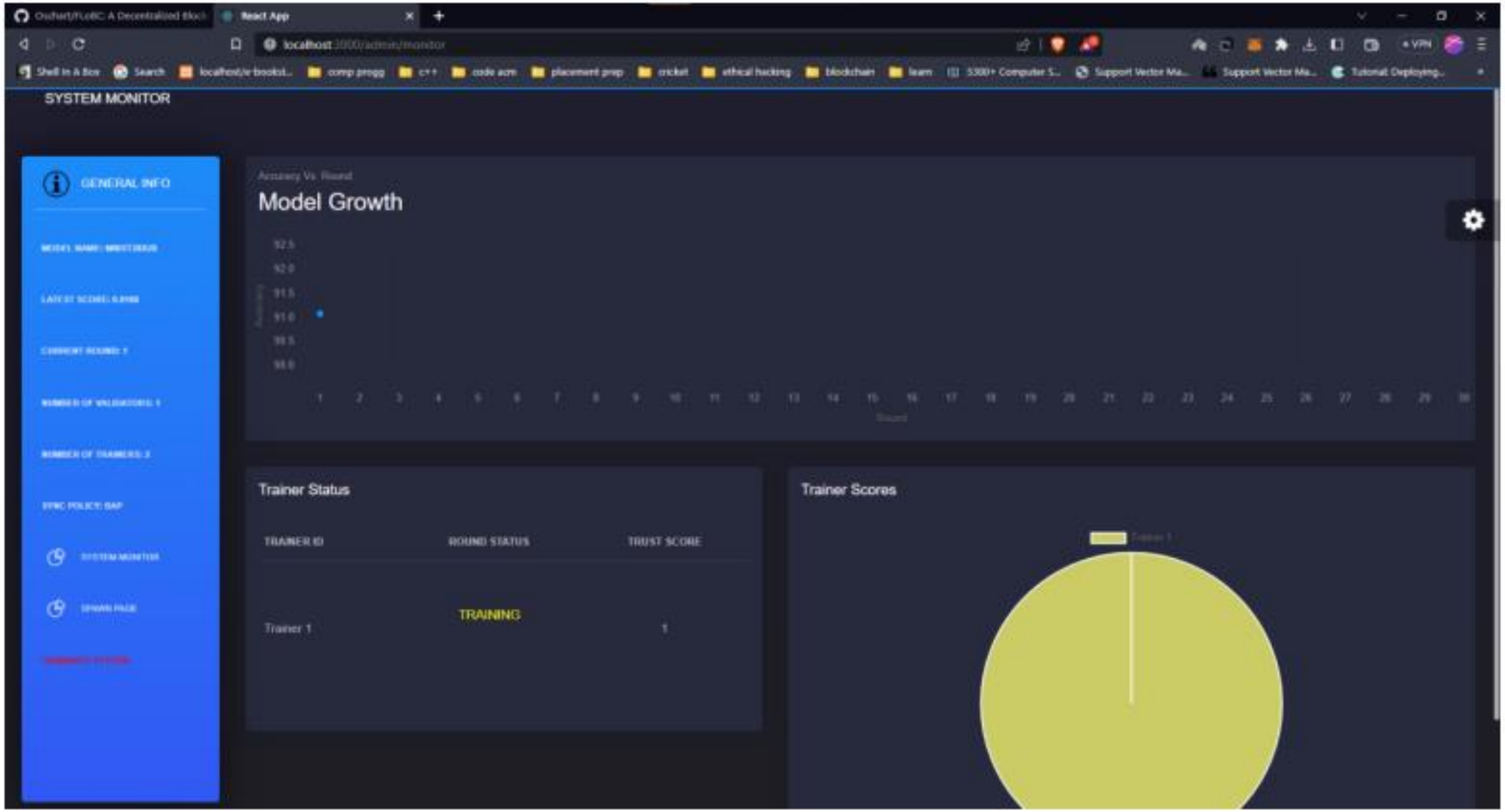
VALIDATOR VALIDATING THE MODEL

Releasing Model :



```
i = 0
new node with ports: 9000 (public)
----------------------------------------
Model update received, validating...
Output Output { status: ExitStatus(unix_wait_status(0)), stdout: "VALID:0.910799
9801635742\n", stderr: "" }
Validation verdict: VALID
----------------------------------------
----------------------------------------
New Model Created:: version= 1   accuracy= 91.08%
----------------------------------------
----------------------------------------
Model update received, validating...
```

Figure 23.1. VALIDATOR CREATING NEW MODEL



GENERAL INFO

MODEL NAME: [illegible]

LATEST SCORE: [illegible]

CURRENT ROUND: 1

NUMBER OF VALIDATORS: 1

NUMBER OF TRAINERS: 2

SYNC POLICY: BAP

SYSTEM MONITOR

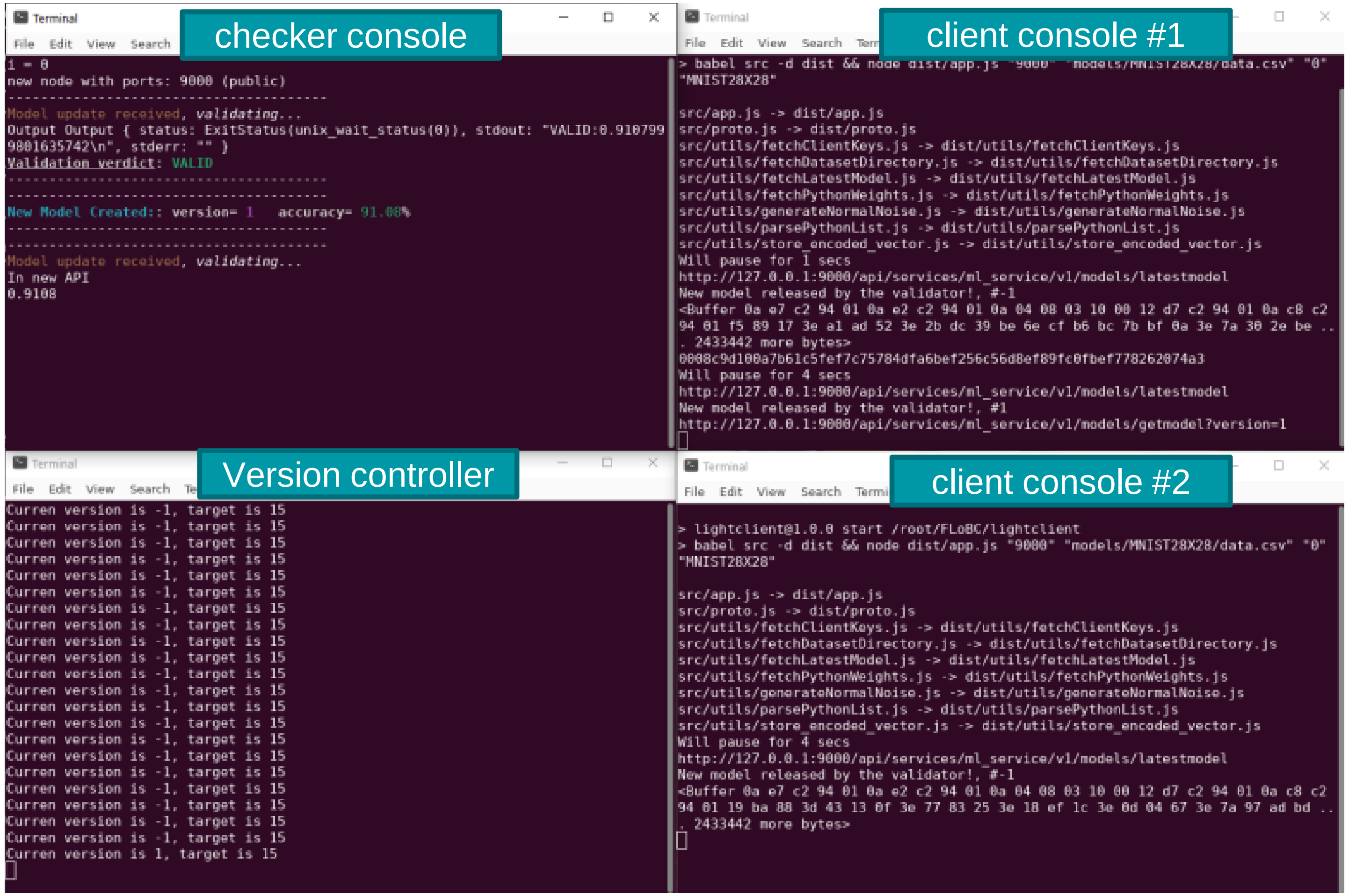
SPAWN PAGE

[illegible]

Trainer Status

TRAINER ID	ROUND STATUS	TRUST SCORE
Trainer 1	TRAINING	1

NEW MODEL REFLECTED IN DASHBOARD



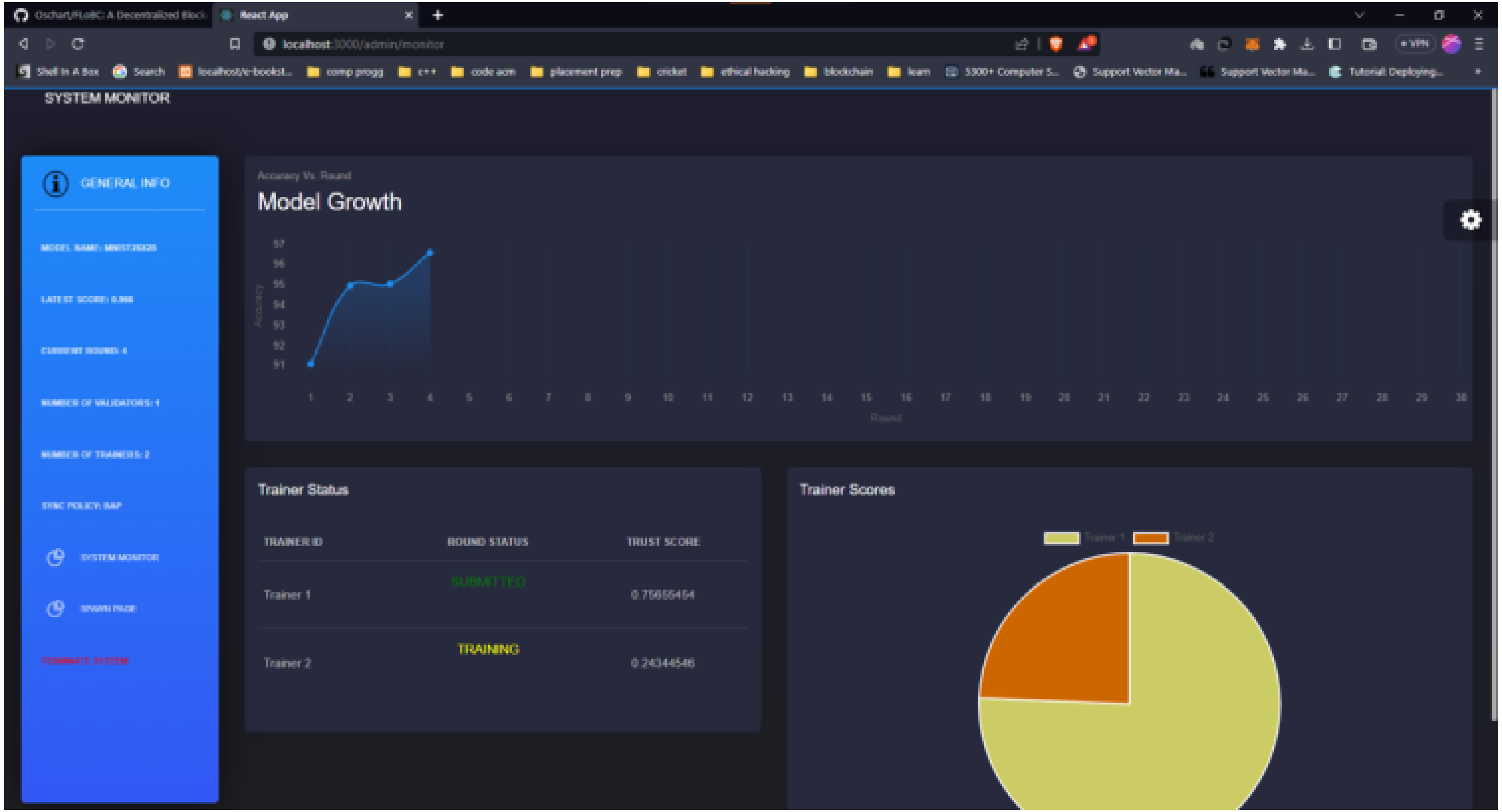
```
i = 0
new node with ports: 9000 (public)
--------------------------------------
Model update received, validating...
Output Output { status: ExitStatus(unix_wait_status(0)), stdout: "VALID:0.910799
9801635742\n", stderr: "" }
Validation verdict: VALID
--------------------------------------
--------------------------------------
New Model Created:: version= 1   accuracy= 91.08%
--------------------------------------
--------------------------------------
Model update received, validating...
In new API
0.9108
```

```
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 1 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 f5 89 17 3e a1 ad 52 3e 2b dc 39 be 6e cf b6 bc 7b bf 0a 3e 7a 30 2e be ..
. 2433442 more bytes>
0008c9d100a7b61c5fef7c75784dfa6bef256c56d8ef89fc0fbef778262074a3
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #1
http://127.0.0.1:9000/api/services/ml_service/v1/models/getmodel?version=1
```

```
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is -1, target is 15
Curren version is 1, target is 15
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 19 ba 88 3d 43 13 0f 3e 77 83 25 3e 18 ef 1c 3e 0d 04 67 3e 7a 97 ad bd ..
. 2433442 more bytes>
```

NEW MODEL SYNCHRONISED ACROSS TRAINERS



MODEL NAME: MNIST28X28

LATEST SCORE: 0.966

CURRENT ROUND: 4

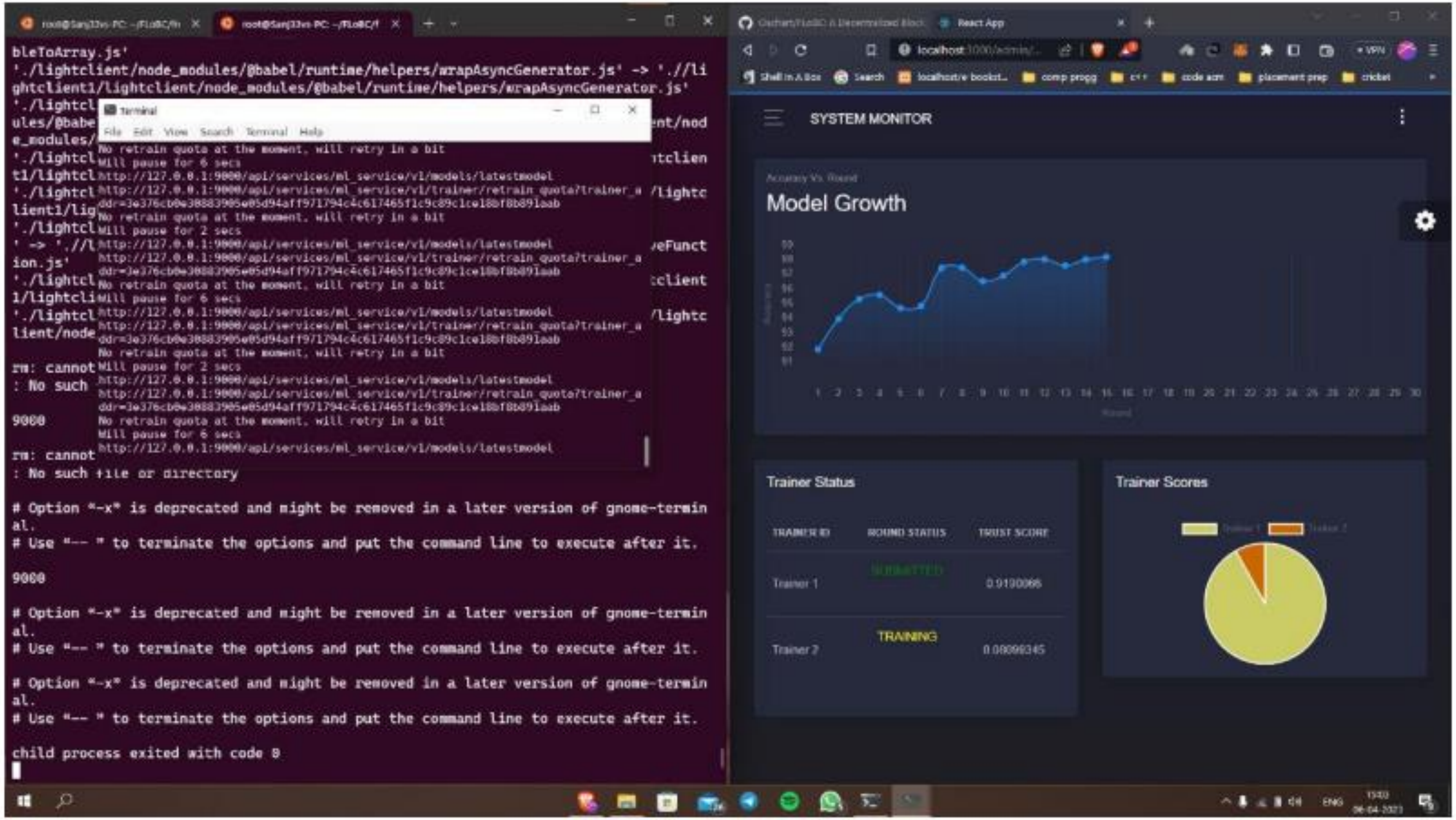
NUMBER OF VALIDATORS: 1

NUMBER OF TRAINERS: 2

SYNC POLICY: BAP

TRAINER ID	ROUND STATUS	TRUST SCORE
Trainer 1	SUBMITTED	0.75655454
Trainer 2	TRAINING	0.24344546

MODEL VERSION #4

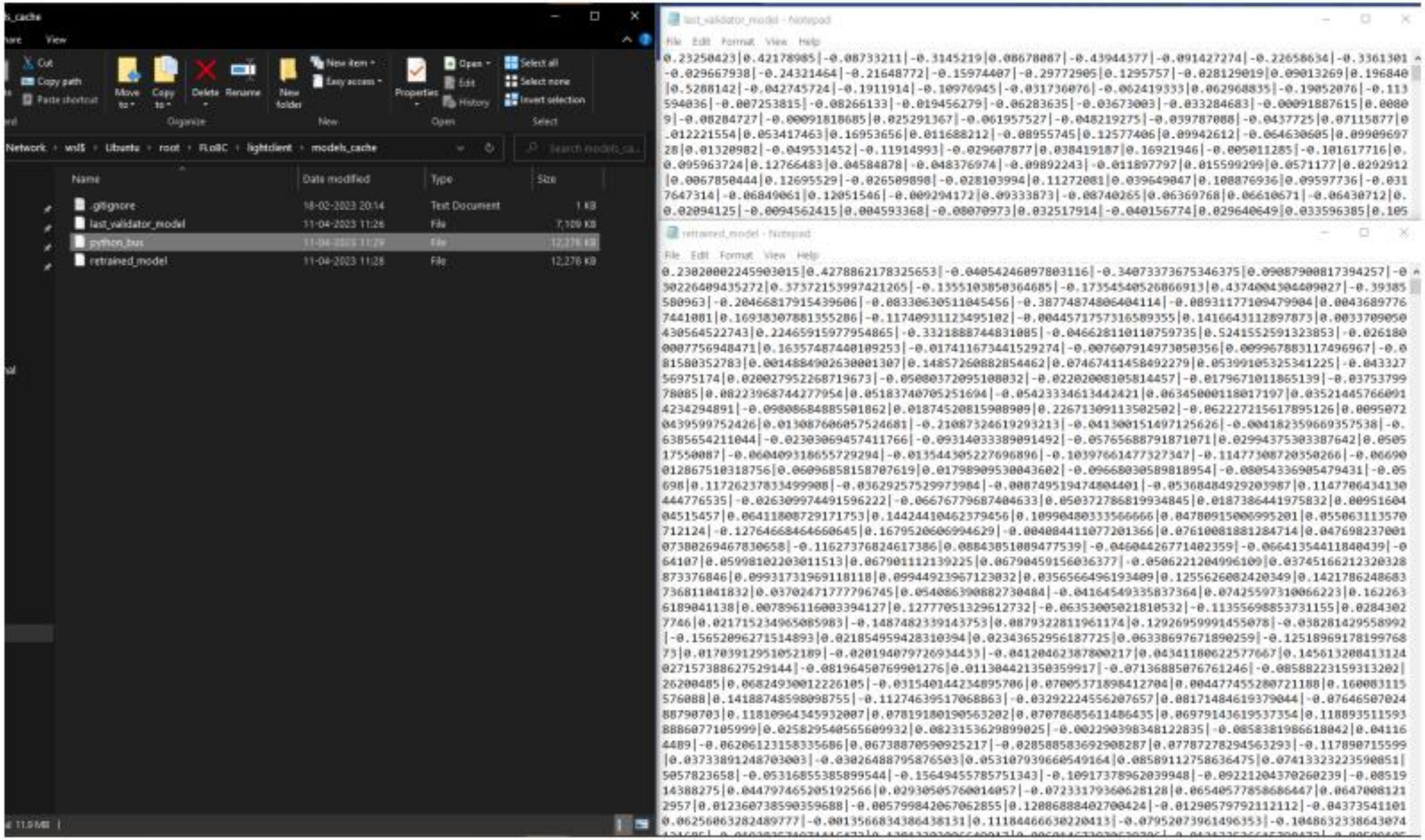


```
bleToArray.js'
'./lightclient/node_modules/@babel/runtime/helpers/wrapAsyncGenerator.js' -> './/li
ghtclient1/lightclient/node_modules/@babel/runtime/helpers/wrapAsyncGenerator.js'
'./lightcl
ules/@babe                                                                  ent/nod
e_modules/
'./lightcl                                                                  tclien
t1/lightcl
'./lightcl                                                                  /lightc
lient1/lig
'./lightcl
' -> './/l                                                                  veFunct
ion.js'
'./lightcl                                                                  tclient
1/lightcli
'./lightcl                                                                  /lightc
lient/node

rm: cannot
: No such

9000

rm: cannot
: No such file or directory

# Option "-x" is deprecated and might be removed in a later version of gnome-termin
al.
# Use "-- " to terminate the options and put the command line to execute after it.

9000

# Option "-x" is deprecated and might be removed in a later version of gnome-termin
al.
# Use "-- " to terminate the options and put the command line to execute after it.

# Option "-x" is deprecated and might be removed in a later version of gnome-termin
al.
# Use "-- " to terminate the options and put the command line to execute after it.

child process exited with code 0
```

```
No retrain quota at the moment, will retry in a bit
Will pause for 6 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=3e376cb0e30883905e05d94aff971794c4c617465f1c9c89c1ce18bf8b891aab
No retrain quota at the moment, will retry in a bit
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=3e376cb0e30883905e05d94aff971794c4c617465f1c9c89c1ce18bf8b891aab
No retrain quota at the moment, will retry in a bit
Will pause for 6 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=3e376cb0e30883905e05d94aff971794c4c617465f1c9c89c1ce18bf8b891aab
No retrain quota at the moment, will retry in a bit
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=3e376cb0e30883905e05d94aff971794c4c617465f1c9c89c1ce18bf8b891aab
No retrain quota at the moment, will retry in a bit
Will pause for 6 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
```

TRAINER ID	ROUND STATUS	TRUST SCORE
Trainer 1	SUBMITTED	0.9130066
Trainer 2	TRAINING	0.08099345

FINAL MODEL GROWTH GRAPH VERSION #15



```
Network › wsl$ › Ubuntu › root › FLoBC › lightclient › models_cache

Name                   Date modified       Type            Size
.gitignore             18-02-2023 20:14    Text Document   1 KB
last_validator_model   11-04-2023 11:26    File            7,109 KB
python_bus             11-04-2023 11:29    File            12,276 KB
retrained_model        11-04-2023 11:28    File            12,276 KB
```

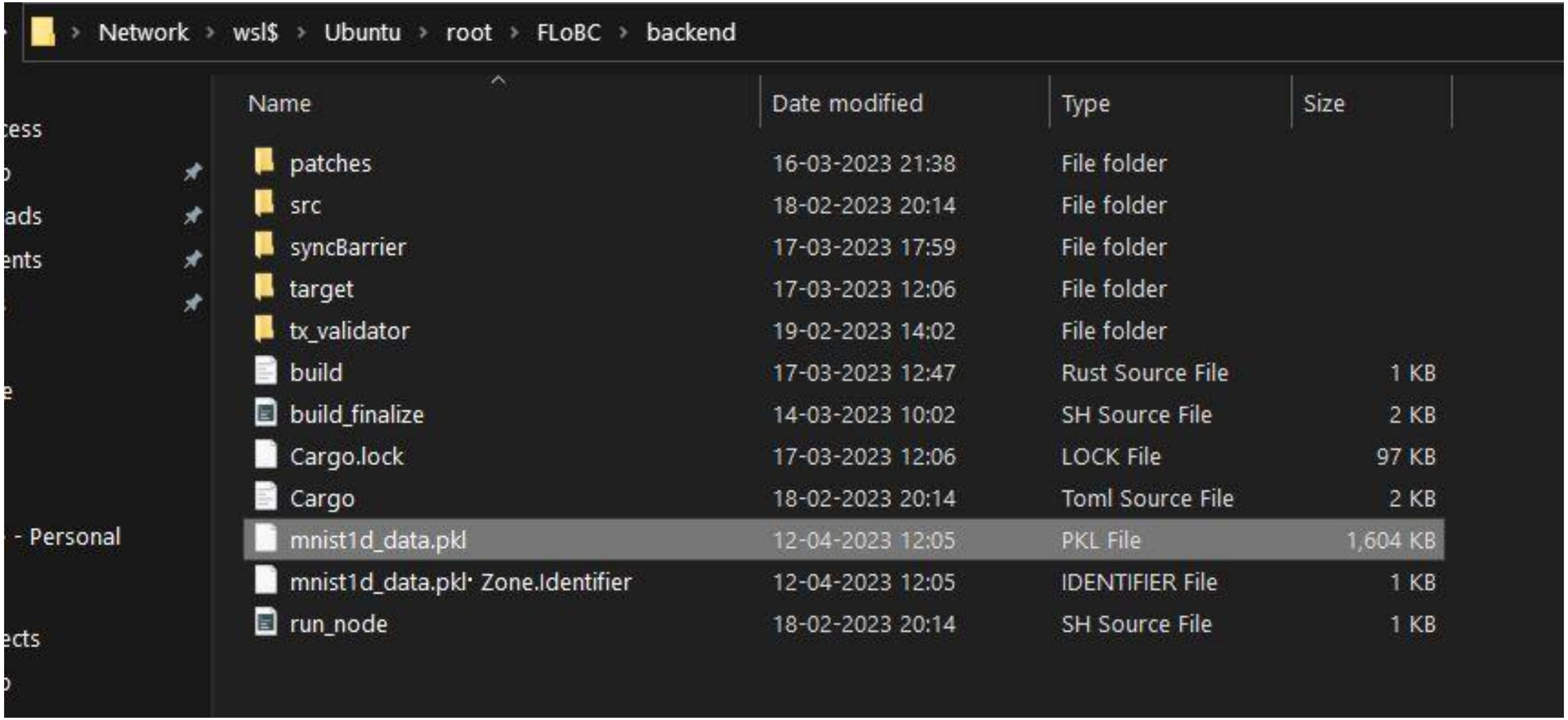
last_validator_model - Notepad

```
0.23250423|0.42178985|-0.08733211|-0.3145219|0.08678087|-0.43944377|-0.091427274|-0.22658634|-0.3361301
-0.029667938|-0.24321464|-0.21648772|-0.15974407|-0.29772905|0.1295757|-0.028129019|0.09013269|0.196840
|0.5288142|-0.042745724|-0.1911914|-0.10976945|-0.031736076|-0.062419333|0.062968835|-0.19052076|-0.113
594036|-0.007253815|-0.08266133|-0.019456279|-0.06283635|-0.03673003|-0.033284683|-0.00091887615|0.0080
9|-0.08284727|-0.00091818685|0.025291367|-0.061957527|-0.048219275|-0.039787088|-0.0437725|0.07115877|0
.012221554|0.053417463|0.16953656|0.011688212|-0.08955745|0.12577406|0.09942612|-0.064630605|0.09909697
28|0.01320982|-0.049531452|-0.11914993|-0.029607877|0.038419187|0.16921946|-0.005011285|-0.101617716|0.
0.095963724|0.12766483|0.04584878|-0.048376974|-0.09892243|-0.011897797|0.015599299|0.0571177|0.0292912
|0.0067850444|0.12695529|-0.026509898|-0.028103994|0.11272081|0.039649047|0.108876936|0.09597736|-0.031
7647314|-0.06849061|0.12051546|-0.009294172|0.09333873|-0.08740265|0.06369768|0.06610671|-0.06430712|0.
0.02094125|-0.0094562415|0.004593368|-0.08070973|0.032517914|-0.040156774|0.029640649|0.033596385|0.105
```

retrained_model - Notepad

```
0.23020002245903015|0.4278862178325653|-0.040542460978031116|-0.3407337367534637 5|0.09087900817394257|-0
30226409435272|0.37372153997421265|-0.1355103850364685|-0.17354540526866913|0.4374004304409027|-0.39385
580963|-0.20466817915439606|-0.08330630511045456|-0.38774874806404114|-0.08931177109479904|0.0043689776
7441081|0.16938307881355286|-0.11740931123495102|-0.004457175731658 9355|0.14166431128 97873|0.0033709050
430564522743|0.22465915977954865|-0.33218887448310 85|-0.04662811011075 9735|0.5241552591323853|-0.026180
0007756948471|0.16357487440109253|-0.017411673441529274|-0.007607914973050356|0.009967883117496967|-0.0
81580352783|0.0014884902630001307|0.14857260882854462|0.07467411458492279|0.05399105325341225|-0.043327
56975174|0.020027952268719673|-0.05080372095108032|-0.022020081058 14457|-0.017967101186 5139|-0.03753799
78085|0.08223968744277954|0.05183740705251694|-0.05423334613442421|0.06345000118017197|0.0352144576609 1
4234294891|-0.09808684885501862|0.018745208159 08909|0.22671309113502502|-0.06222721561789 5126|0.0095072
043959975 2426|0.013087606057524681|-0.21087324619293213|-0.041300151497125626|-0.004182359669357538|-0.
6385654211044|-0.02303069457411766|-0.09314033389091492|-0.05765688791871071|0.029943753033876 42|0.0505
17550087|-0.060409318655729294|-0.013544305227696896|-0.10397661477327347|-0.11477308720350266|-0.06690
012867510318756|0.06096858158707619|0.01798909530043602|-0.09668030589818954|-0.008054336905479431|-0.05
698|0.11726237833499908|-0.03629257529973984|-0.008749519474804401|-0.053684849292039 87|0.1147706434130
444776535|-0.026309974491596222|-0.06676779687404633|0.050372786819934845|0.0187386441975832|0.00951604
04515457|0.06411808729171753|0.14424410462379456|0.10990480333566666|0.047809150069 95201|0.05506311357 0
712124|-0.12764668464660645|0.16795206069946 29|-0.004084411077201366|0.07610081881284714|0.047698237001
073802694678306 58|-0.11627376824617386|0.08843851089477539|-0.046044267714023 59|-0.06641354411840439|-0
64107|0.05998102203011513|0.067901112139225|0.06790459156036377|-0.050622120499610 9|0.03745166212320328
873376846|0.09931731969118118|0.09944923967123032|0.035656649619340 9|0.1255626082420349|0.142178624868 3
736811041832|0.03702471777796745|0.05408639088273048 4|-0.04164549335837364|0.07425597310066223|0.162263
6189041138|0.007896116003394127|0.12777051329612732|-0.06353005021810532|-0.11355698853731155|0.0284302
7746|0.021715234965085983|-0.14874823391437 53|0.08793228119611 74|0.12926959991455078|-0.03828142955899 2
|-0.15652096271514893|0.021854959428310394|0.02343652956187725|0.06338697671890259|-0.12518969178199768
73|0.01703912951052189|-0.020194079726934433|-0.04120462387800217|0.04341180622577667|0.145613208413124
027157388627529144|-0.08196450769901276|0.011304421350359917|-0.07136885076761246|-0.08588223159313202|
26200485|0.06824930012226105|-0.03154014423489 5706|0.07005371898412704|0.004477455280721188|0.160083115
576088|0.14188748598098755|-0.11274639517068863|-0.03292224556207657|0.08171484619379044|-0.07646507024
88790703|0.11810964345932007|0.07819180190563202|0.07078685611486435|0.06979143619537354|0.11889351159 3
8886077105999|0.025829540565609932|0.08231536298990 25|-0.002290398348122835|-0.08583819866180 42|0.04116
4489|-0.06206123158335686|0.06738870590925217|-0.028588583692908287|0.07787278294563293|-0.117890715599
|0.03733891248703003|-0.030264887958765 03|0.053107939660549164|0.08589112758636475|0.07413323223590851|
5057823658|-0.05316855385899544|-0.15649455785751343|-0.10917378962039948|-0.09221204370260239|-0.08519
14388275|0.04479746520519 2566|0.029305057600140 57|-0.07233179360628128|0.06540577858686447|0.0647008121
2957|0.012360738590359688|-0.005799842067062855|0.12086888402700424|-0.012905797921121 12|-0.04373541101
0.06256063282489777|-0.0013566834386438131|0.11184466630220413|-0.07952073961496353|-0.104863233864307 4
```

FINAL MODEL WEIGHTS VERSION #15



Network › wsl$ › Ubuntu › root › FLoBC › backend

Name	Date modified	Type	Size
patches	16-03-2023 21:38	File folder	
src	18-02-2023 20:14	File folder	
syncBarrier	17-03-2023 17:59	File folder	
target	17-03-2023 12:06	File folder	
tx_validator	19-02-2023 14:02	File folder	
build	17-03-2023 12:47	Rust Source File	1 KB
build_finalize	14-03-2023 10:02	SH Source File	2 KB
Cargo.lock	17-03-2023 12:06	LOCK File	97 KB
Cargo	18-02-2023 20:14	Toml Source File	2 KB
mnist1d_data.pkl	12-04-2023 12:05	PKL File	1,604 KB
mnist1d_data.pkl·Zone.Identifier	12-04-2023 12:05	IDENTIFIER File	1 KB
run_node	18-02-2023 20:14	SH Source File	1 KB

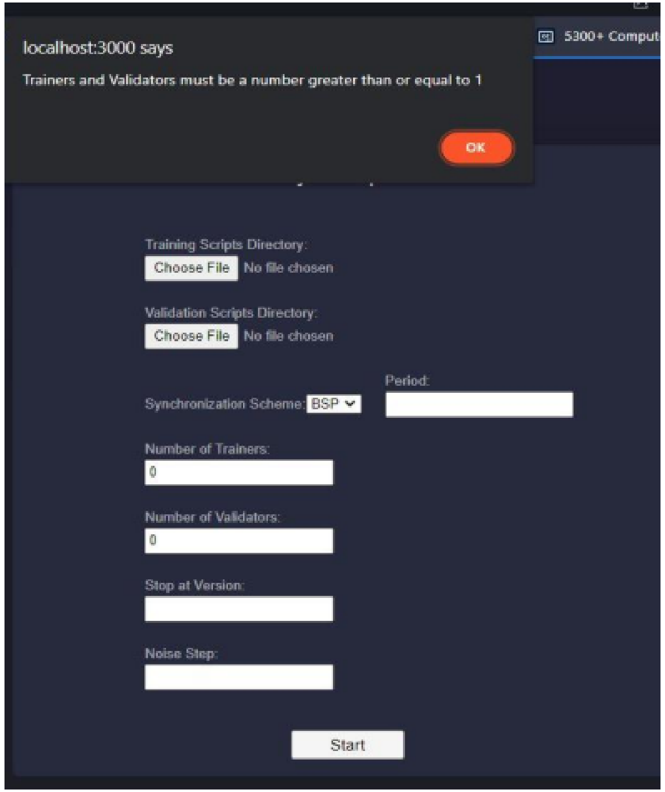
Finally, the trained model has been downloaded

TEST CASES

1)	**When clients and checkers are less than or equal to 0**	Link
2)	**When period is 0 and the scheme is synchronous**	Link
3)	**When spot at version is left null or is filled with float value**	Link
4)	**Force Terminating System**	Link
5)	**Synchronization policy**	Link
6)	**Sync Signal Propagation**	Link
7)	**Sending Model Parameters**	Link
8)	**Validating Model Parameters**	Link
9)	**Releasing New Model**	Link
10)	**Reflecting model update in frontend**	Link

TEST CASES:

Case 1 (When Trainers and Validators are less than or equal to 0)


localhost:3000 says

Trainers and Validators must be a number greater than or equal to 1

Training Scripts Directory: No file chosen

Validation Scripts Directory: No file chosen

Synchronization Scheme: BSP

Period:

Number of Trainers: 0

Number of Validators: 0

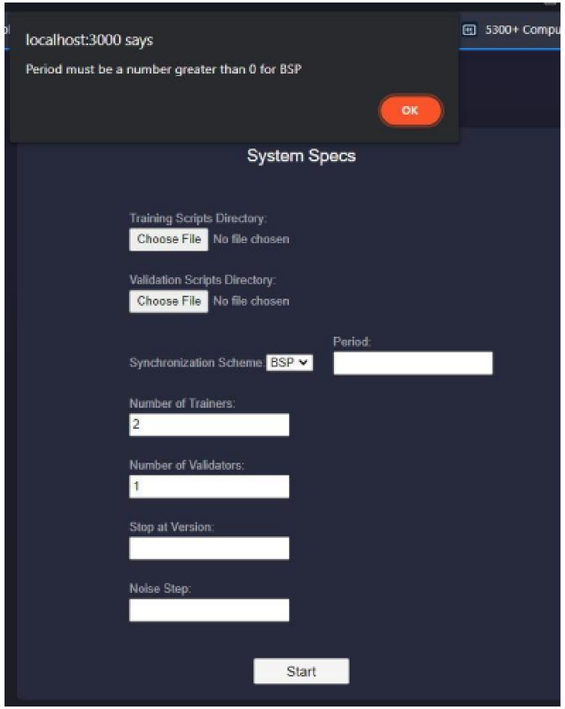
Stop at Version:

Noise Step:

PROMPT ARAISES IF CONSTRAINTS ARE VIOLATED [TRAINERS AND VALIDATORS ARE NULL]

Input : Clients and checkers are <= 0
Output : Alert prompt appears if clients and checkers are less than or equal to 0

Case 2 (When period is 0 and the scheme is synchronous)


localhost:3000 says

Period must be a number greater than 0 for BSP

Synchronization Scheme: BSP

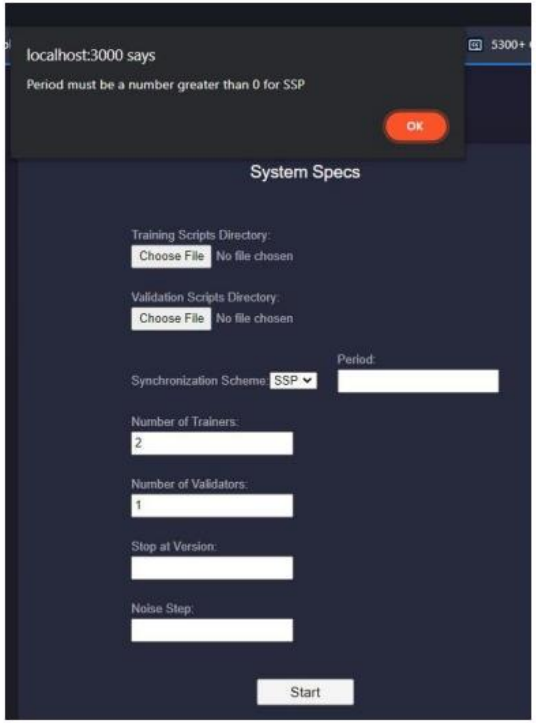
Number of Trainers: 2

Number of Validators: 1

Stop at Version:

Noise Step:

Figure 25.1. PROMPT ARAISES IF CONSTRAINTS ARE VIOLATED [BSP]


localhost:3000 says

Period must be a number greater than 0 for SSP

Synchronization Scheme: SSP

Number of Trainers: 2

Number of Validators: 1

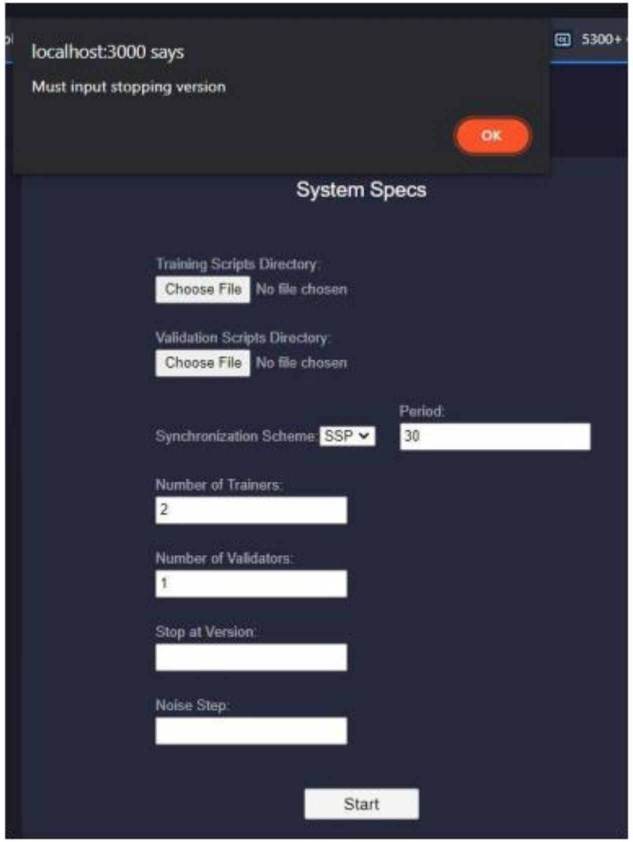
Stop at Version:

Noise Step:

Figure 25.2. PROMPT ARAISES IF CONSTRAINTS ARE VIOLATED [SSP]

Input : Synchronous scheme and period is 0
Output : Alert prompt appears if period is 0 and scheme is synchronous

Case 3 (When spot at version is left null or is filled with float value)

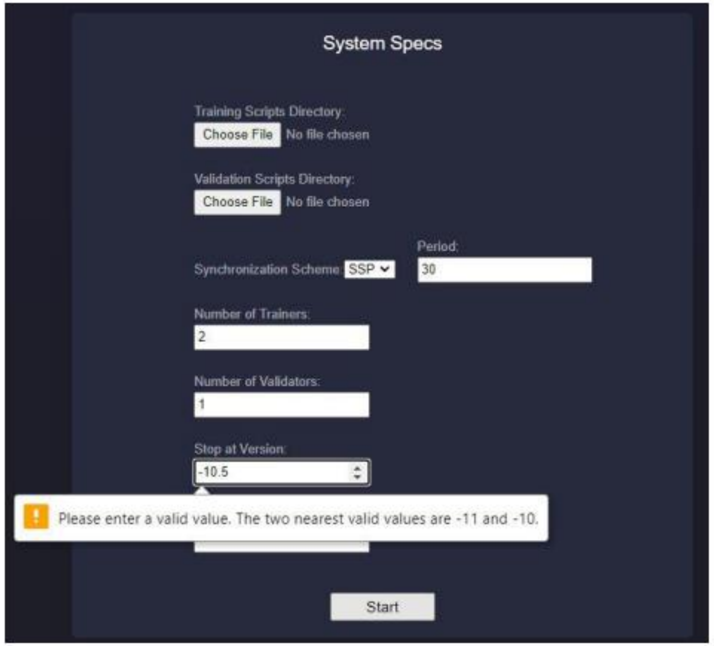


localhost:3000 says

Must input stopping version

OK
System Specs
Training Scripts Directory:
Choose File No file chosen
Validation Scripts Directory:
Choose File No file chosen
Synchronization Scheme: SSP
Period:
30
Number of Trainers:
2
Number of Validators:
1
Stop at Version:
Noise Step:
Start

Figure 26.1. PROMPT ARAISES IF CONSTRAINTS ARE VIOLATED [VERSION CONTROL]



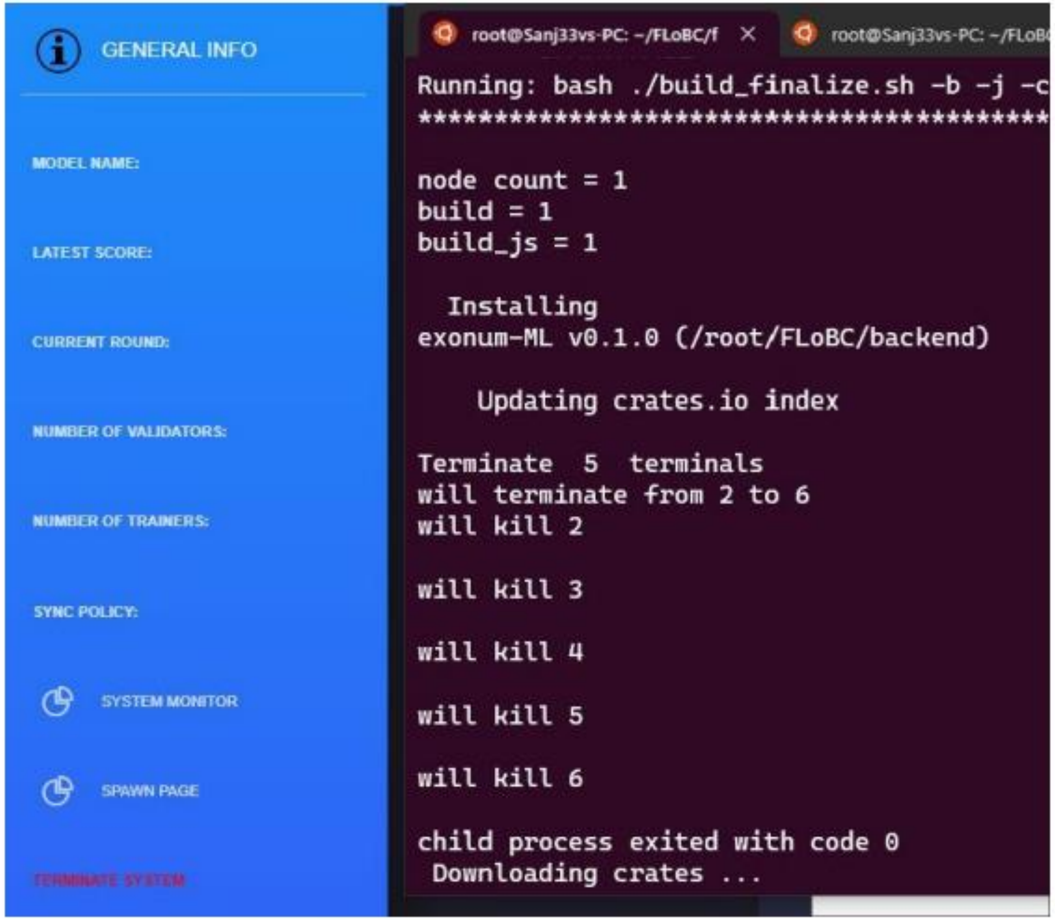
Please enter a valid value. The two nearest valid values are -11 and -10.

System Specs
Training Scripts Directory:
Choose File No file chosen
Validation Scripts Directory:
Choose File No file chosen
Synchronization Scheme: SSP
Period:
30
Number of Trainers:
2
Number of Validators:
1
Stop at Version:
-10.5
Start

Figure 26.2. PROMPT ARAISES IF CONSTRAINTS ARE VIOLATED [VERSION

Input : Stop at version field is left null or filled with float value
Output : Alert prompt appears to rectify the stop at version field

Case 4 (Force Terminating System)



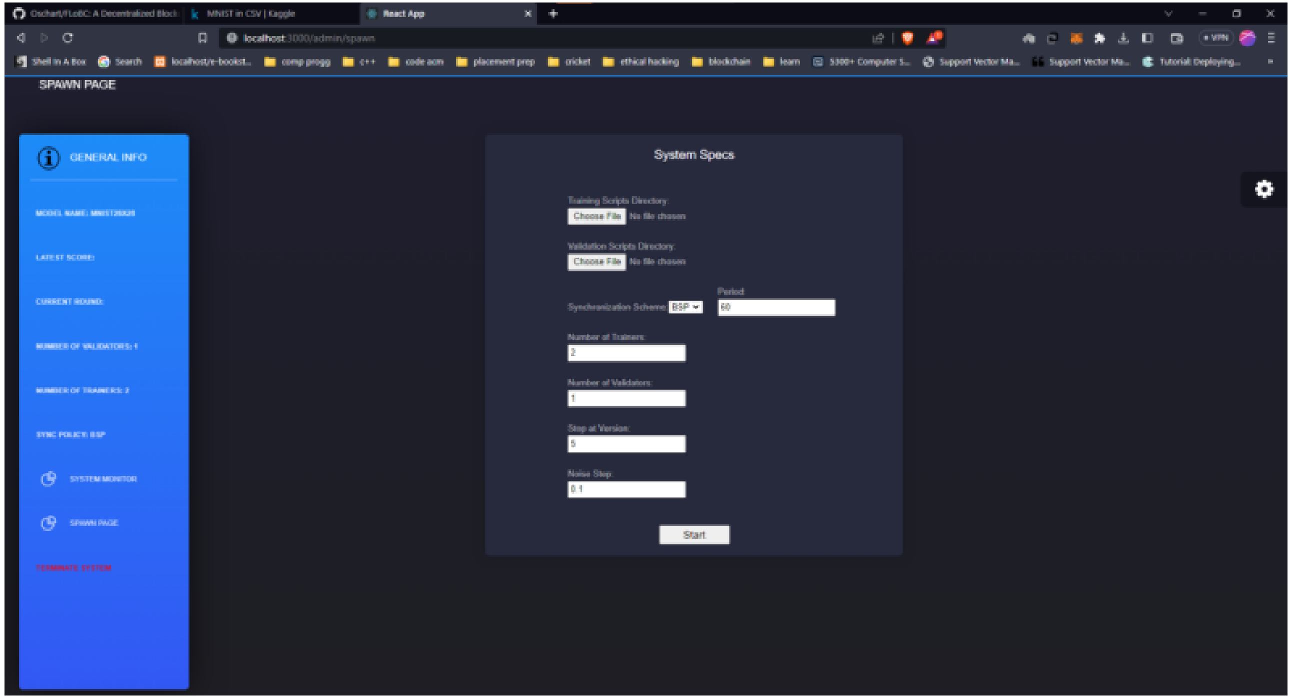
```
Running: bash ./build_finalize.sh -b -j -c
*******************************************

node count = 1
build = 1
build_js = 1

  Installing
exonum-ML v0.1.0 (/root/FLoBC/backend)

    Updating crates.io index

Terminate  5  terminals
will terminate from 2 to 6
will kill 2

will kill 3

will kill 4

will kill 5

will kill 6

child process exited with code 0
 Downloading crates ...
```

Figure 27. TERMINATING

Input : Fore kill framework instance signal sent to REST API

Output : The framework instance should kill and clear all the residues from blockchain

Case 5 (Synchronization policy)



MODEL NAME: MNIST2020

LATEST SCORE:

CURRENT ROUND:

NUMBER OF VALIDATORS: 1

NUMBER OF TRAINERS: 2

SYNC POLICY: BSP

System Specs

Training Scripts Directory: No file chosen

Validation Scripts Directory: No file chosen

Synchronization Scheme: BSP

Period: 60

Number of Trainers: 2

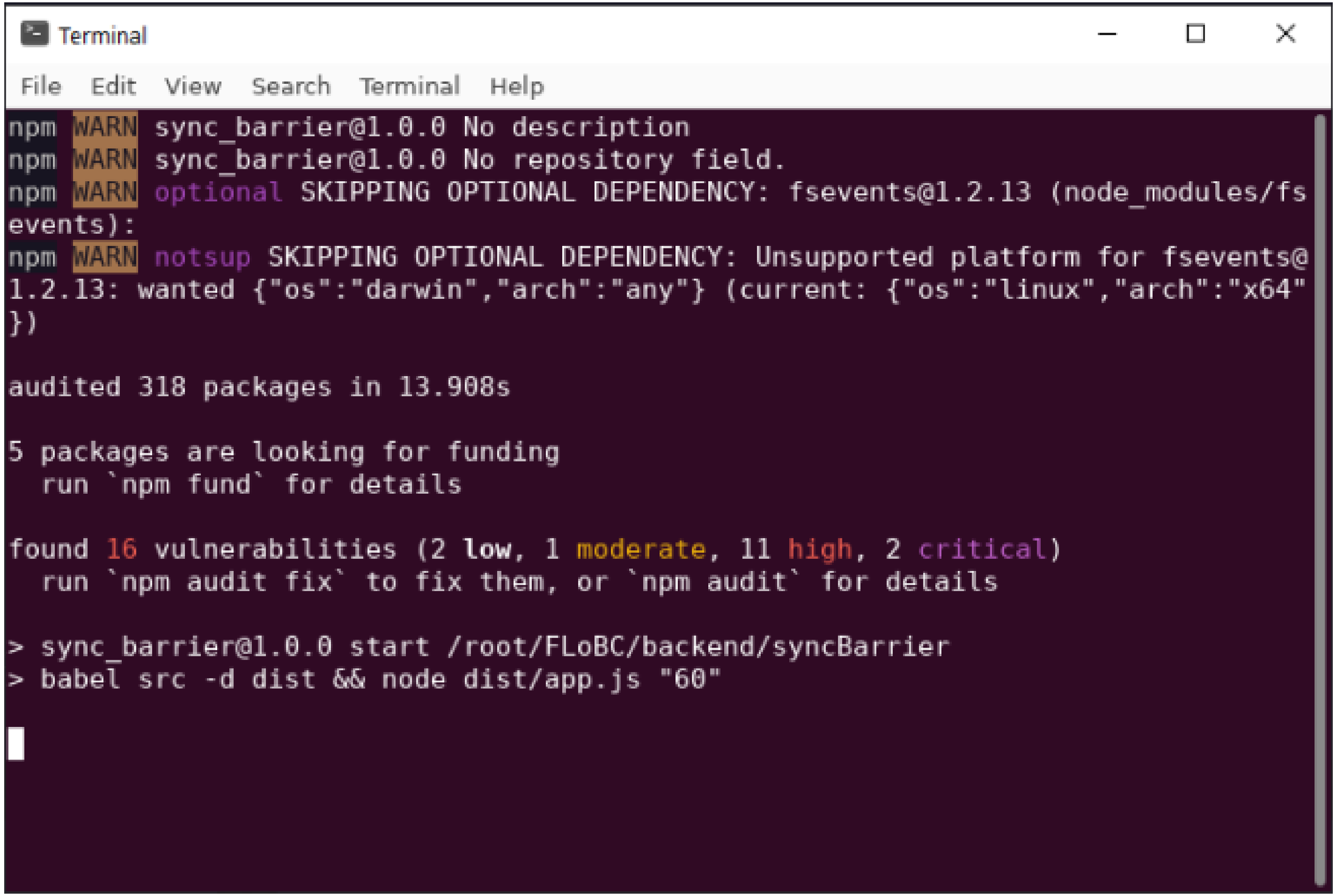
Number of Validators: 1

Stop at Version: 5

Noise Step: 0.1

Figure 28.1. SPAWN PAGE

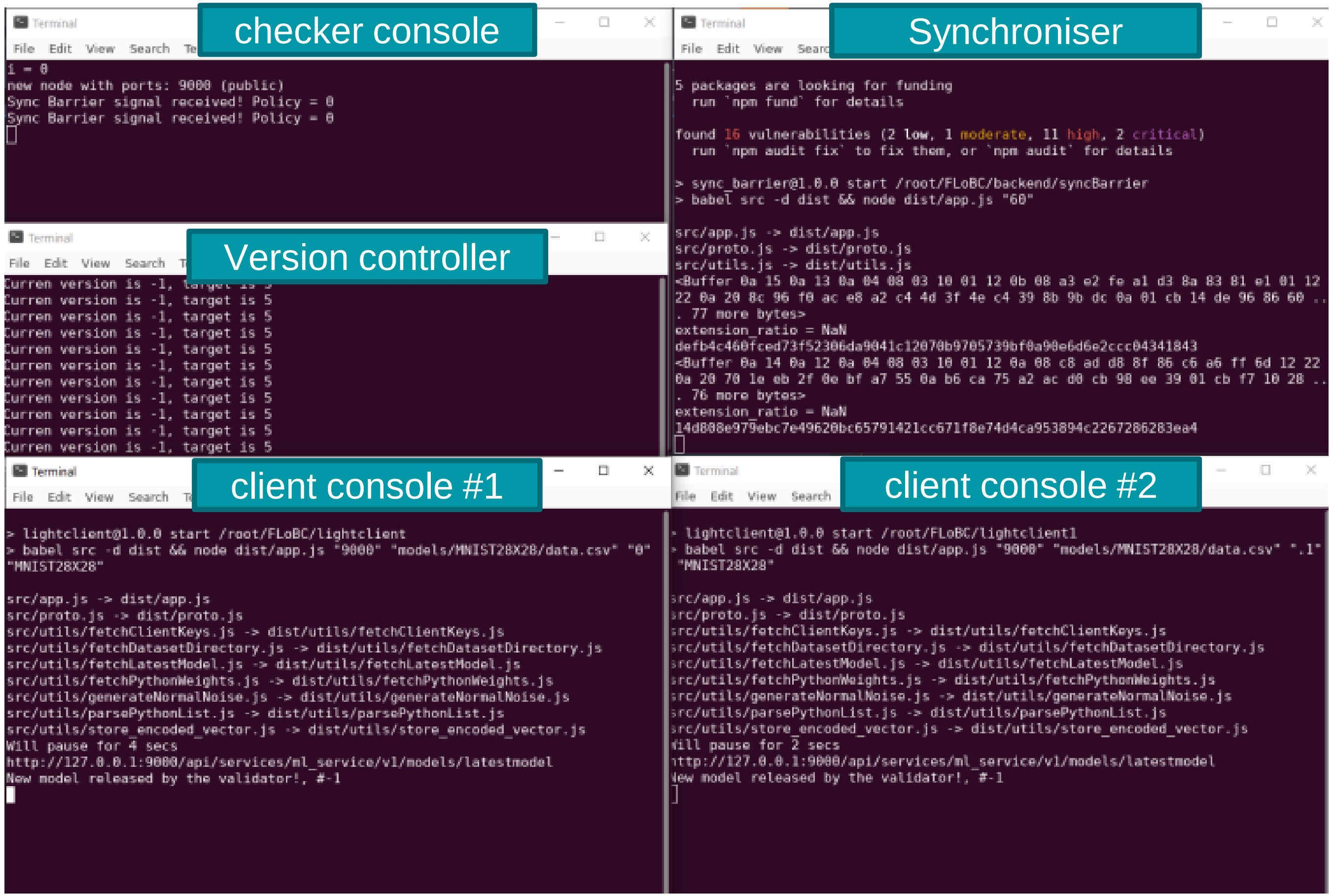
Input : Synchronization scheme chosen from the frontend dropdown
Output : API correctly interprets the policy and spawns clients and checkers corresponding to the policy



```
npm WARN sync_barrier@1.0.0 No description
npm WARN sync_barrier@1.0.0 No repository field.
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.13 (node_modules/fs
events):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@
1.2.13: wanted {"os":"darwin","arch":"any"} (current: {"os":"linux","arch":"x64"
})

audited 318 packages in 13.908s

5 packages are looking for funding
  run `npm fund` for details

found 16 vulnerabilities (2 low, 1 moderate, 11 high, 2 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

> sync_barrier@1.0.0 start /root/FLoBC/backend/syncBarrier
> babel src -d dist && node dist/app.js "60"

```

SYNCHRONIZER

Case 6 (Sync Signal Propagation)



```
1 = 0
new node with ports: 9000 (public)
Sync Barrier signal received! Policy = 0
Sync Barrier signal received! Policy = 0
```

```

5 packages are looking for funding
  run `npm fund` for details

found 16 vulnerabilities (2 low, 1 moderate, 11 high, 2 critical)
  run `npm audit fix` to fix them, or `npm audit` for details

> sync_barrier@1.0.0 start /root/FLoBC/backend/syncBarrier
> babel src -d dist && node dist/app.js "60"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils.js -> dist/utils.js
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 a3 e2 fe a1 d3 8a 83 81 e1 01 12
22 0a 20 8c 96 f0 ac e8 a2 c4 4d 3f 4e c4 39 8b 9b dc 0a 01 cb 14 de 96 86 60 ..
. 77 more bytes>
extension_ratio = NaN
defb4c460fced73f52306da9041c12070b9705739bf0a90e6d6e2ccc04341843
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 c8 ad d8 8f 86 c6 a6 ff 6d 12 22
0a 20 70 1e eb 2f 0e bf a7 55 0a b6 ca 75 a2 ac d0 cb 98 ee 39 01 cb f7 10 28 ..
. 76 more bytes>
extension_ratio = NaN
14d808e979ebc7e49620bc65791421cc671f8e74d4ca953894c2267286283ea4
```

```
Curren version is -1,
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
```

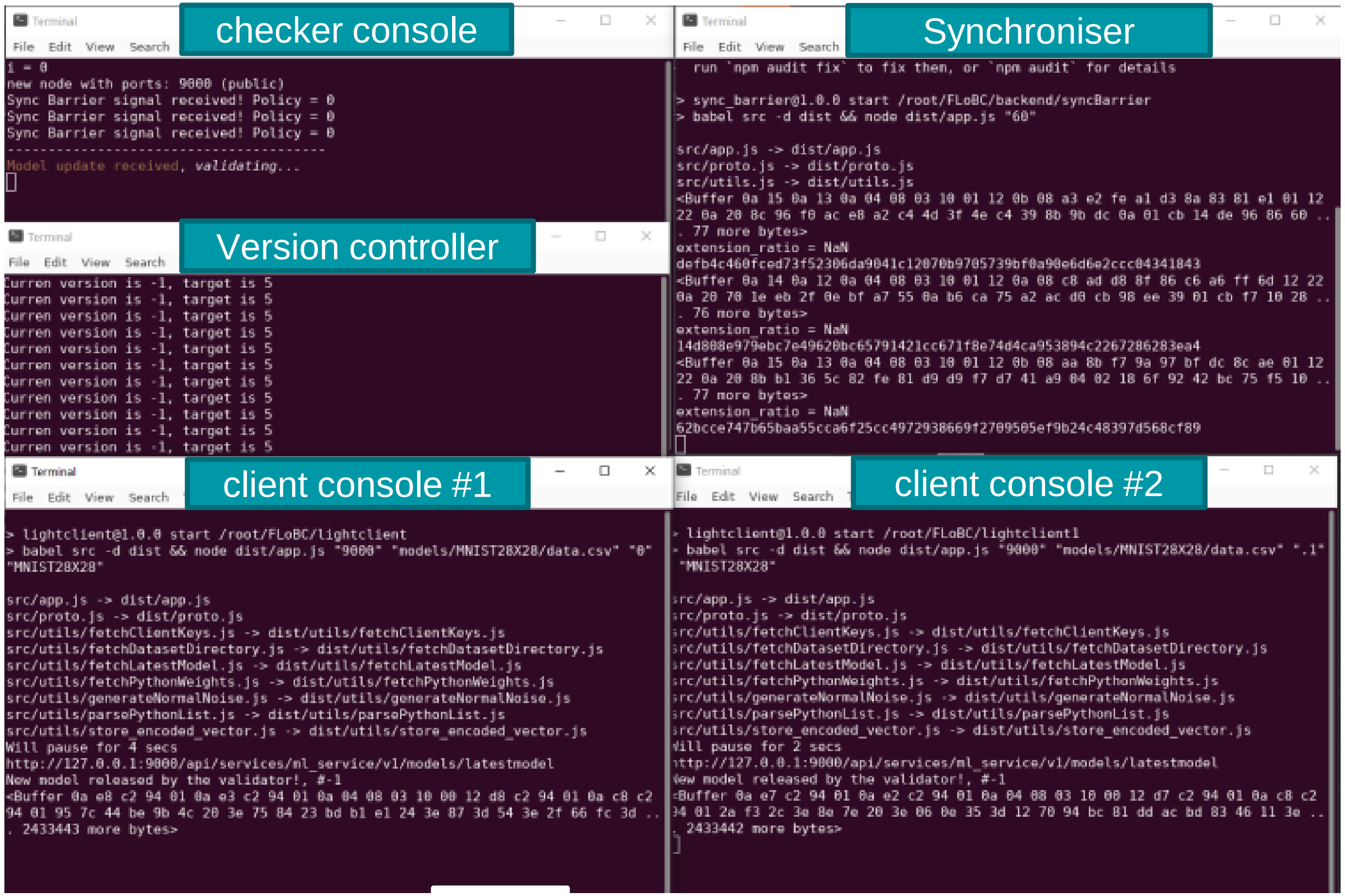
```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" ".1"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
```

Figure 29. TRAINER ALS RECEIVED BY VALIDATOR

Input : Clients and checkers are created

Output : Clients and checkers communicate with each other and shares model updates

Case 7 (Sending Model Parameters)



```
i = 0
new node with ports: 9000 (public)
Sync Barrier signal received! Policy = 0
Sync Barrier signal received! Policy = 0
Sync Barrier signal received! Policy = 0
----------------------------------------
Model update received, validating...
```

```
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
```

```
  run `npm audit fix` to fix them, or `npm audit` for details

> sync_barrier@1.0.0 start /root/FLoBC/backend/syncBarrier
> babel src -d dist && node dist/app.js "60"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils.js -> dist/utils.js
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 a3 e2 fe a1 d3 8a 83 81 e1 01 12
22 0a 20 8c 96 f0 ac e8 a2 c4 4d 3f 4e c4 39 8b 9b dc 0a 01 cb 14 de 96 86 60 ..
. 77 more bytes>
extension_ratio = NaN
defb4c460fced73f52306da9041c12070b9705739bf0a90e6d6e2ccc04341843
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 c8 ad d8 8f 86 c6 a6 ff 6d 12 22
0a 20 70 1e eb 2f 0e bf a7 55 0a b6 ca 75 a2 ac d0 cb 98 ee 39 01 cb f7 10 28 ..
. 76 more bytes>
extension_ratio = NaN
14d808e979ebc7e49620bc65791421cc671f8e74d4ca953894c2267286283ea4
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 aa 8b f7 9a 97 bf dc 8c ae 01 12
22 0a 20 8b b1 36 5c 82 fe 01 d9 d9 f7 d7 41 a9 04 02 18 6f 92 42 bc 75 f5 10 ..
. 77 more bytes>
extension_ratio = NaN
62bcce747b65baa55cca6f25cc4972938669f2709505ef9b24c48397d568cf89
```

```
> lightclient@1.0.0 start /root/FLoBC/lightclient
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" "0"
"MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e8 c2 94 01 0a e3 c2 94 01 0a 04 08 03 10 00 12 d8 c2 94 01 0a c8 c2
94 01 95 7c 44 be 9b 4c 20 3e 75 84 23 bd b1 e1 24 3e 87 3d 54 3e 2f 66 fc 3d ..
. 2433443 more bytes>
```

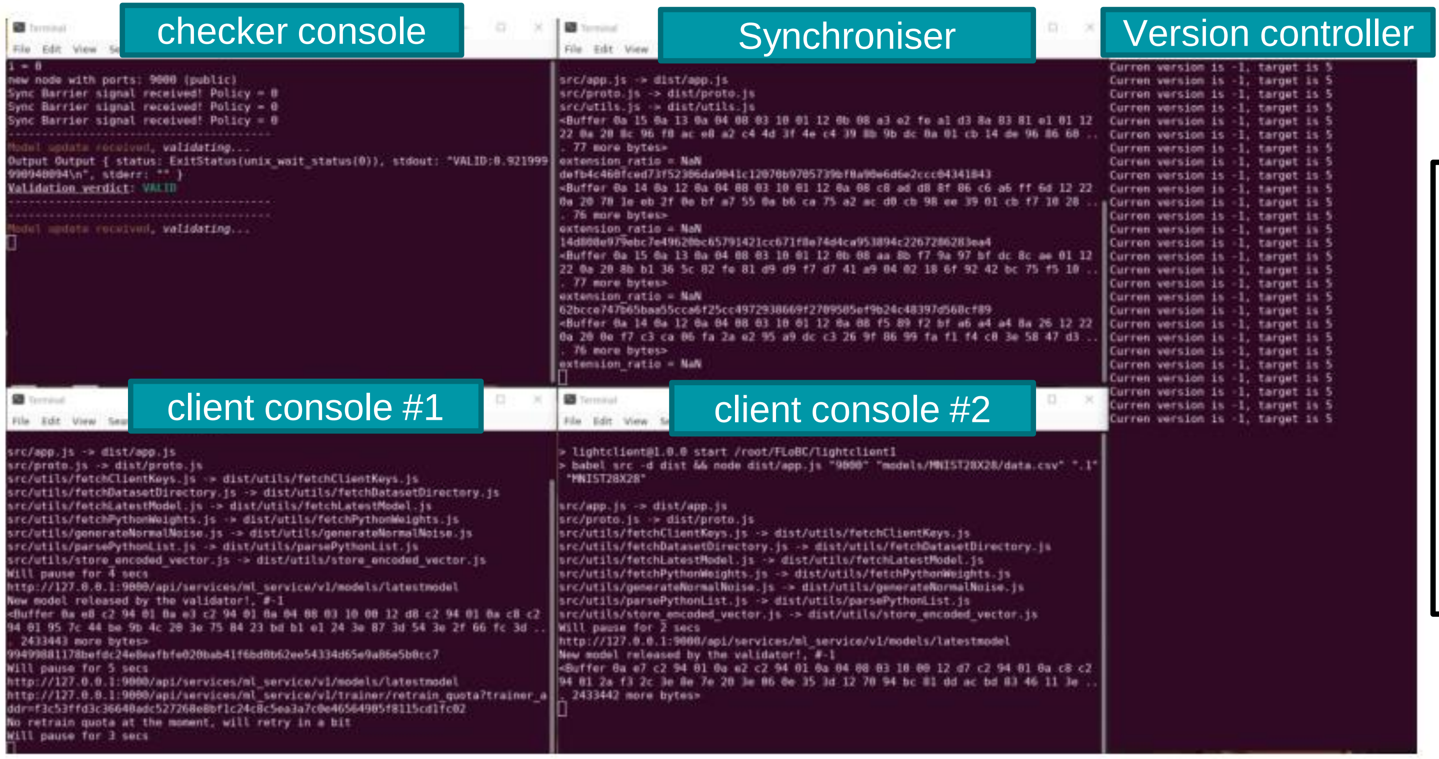
```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" ".1"
 "MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 2a f3 2c 3e 0e 7e 20 3e 06 0e 35 3d 12 70 94 bc 81 dd ac bd 83 46 11 3e ..
. 2433442 more bytes>
```

Input : Clients train the model with local dataset

Output : Trained model updates are shared to checkers

Figure 30. TRAINE[illegible]G MODEL PARAMETERS WITH VALIDATORS

Case 8 (Validating Model Parameters)



```
i = 0
new node with ports: 9000 (public)
Sync Barrier signal received! Policy = 0
Sync Barrier signal received! Policy = 0
Sync Barrier signal received! Policy = 0
----------------------------------------
Model update received, validating...
Output Output { status: ExitStatus(unix_wait_status(0)), stdout: "VALID:0.921999
990940094\n", stderr: "" }
Validation verdict: VALID
----------------------------------------
----------------------------------------
Model update received, validating...
```

```
src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils.js -> dist/utils.js
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 a3 e2 fe a1 d3 8a 83 81 e1 01 12
22 0a 20 8c 96 f0 ac e8 a2 c4 4d 3f 4e c4 39 8b 9b dc 0a 01 cb 14 de 96 86 60 ..
. 77 more bytes>
extension_ratio = NaN
defb4c460fced73f52306da9041c12070b9705739bf0a90e6d6e2ccc04341043
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 c8 ad d8 8f 86 c6 a6 ff 6d 12 22
0a 20 70 1e eb 2f 0e bf a7 55 0a b6 ca 75 a2 ac d0 cb 98 ee 39 01 cb f7 10 20 ..
. 76 more bytes>
extension_ratio = NaN
14d000e979ebc7e49620bc65791421cc671f0e74d4ca953894c2267286283ea4
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 aa 8b f7 9a 97 bf dc 8c ae 01 12
22 0a 20 8b b1 36 5c 82 fe 81 d9 d9 f7 d7 41 a9 04 02 18 6f 92 42 bc 75 f5 10 ..
. 77 more bytes>
extension_ratio = NaN
62bcce747b65baa55cca6f25cc4972938669f2709505ef9b24c48397d568cf89
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 f5 89 f2 bf a6 a4 a4 8a 26 12 22
0a 20 0e f7 c3 ca 06 fa 2a e2 95 a9 dc c3 26 9f 86 99 fa f1 f4 c0 3e 58 47 d3 ..
. 76 more bytes>
extension_ratio = NaN
```

```
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
```

```
src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e8 c2 94 01 0a e3 c2 94 01 0a 04 08 03 10 00 12 d8 c2 94 01 0a c8 c2
94 01 95 7c 44 be 9b 4c 20 3e 75 84 23 bd b1 e1 24 3e 87 3d 54 3e 2f 66 fc 3d ..
. 2433443 more bytes>
99499881178befdc24e0eafbfe020bab41f6bd0b62ee54334d65e9a86e5b0cc7
Will pause for 5 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=f3c53ffd3c36640adc527268e0bf1c24c8c5ea3a7c0e46564905f8115cd1fc02
No retrain quota at the moment, will retry in a bit
Will pause for 3 secs
```

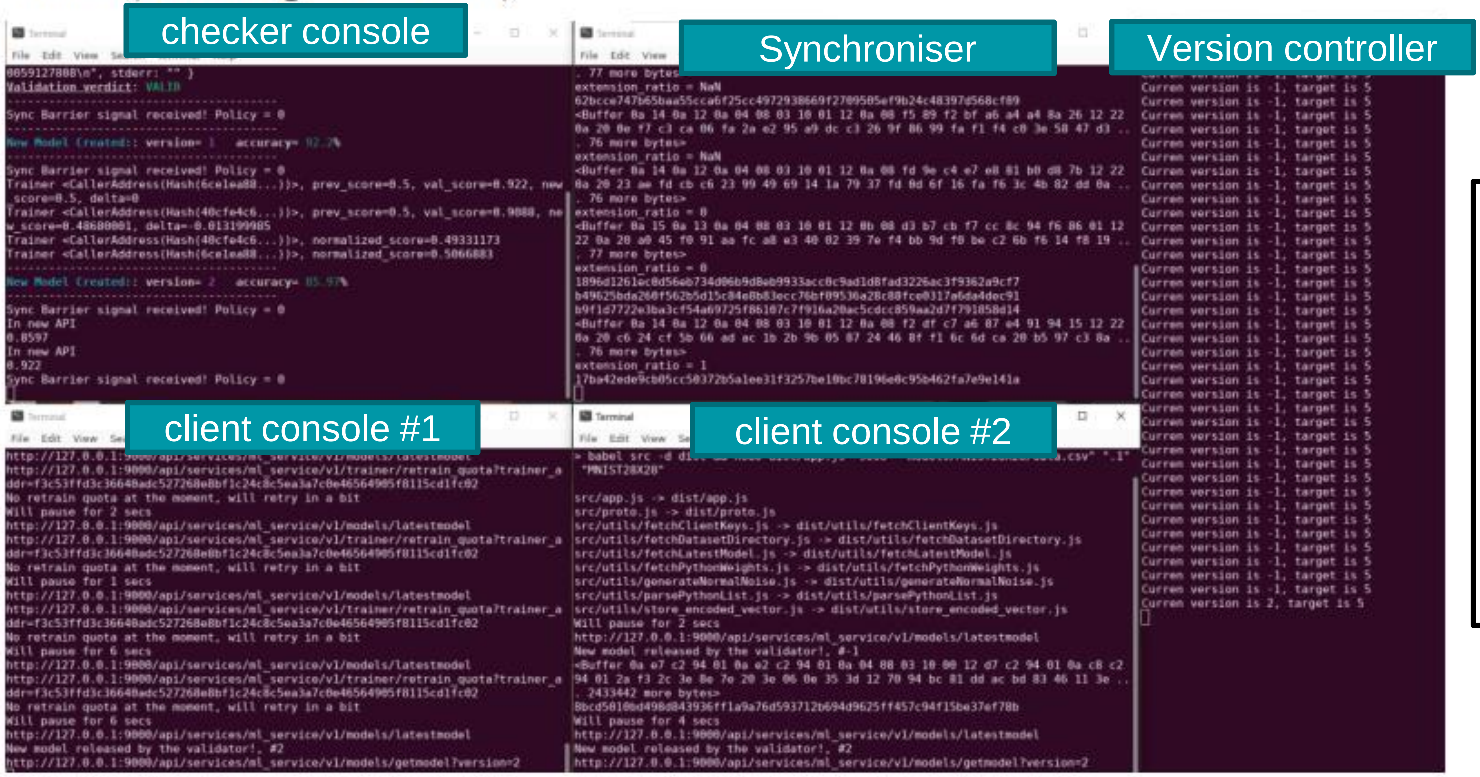
```
> lightclient@1.0.0 start /root/FLoBC/lightclient1
> babel src -d dist && node dist/app.js "9000" "models/MNIST28X28/data.csv" ".1"
 "MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 2a f3 2c 3e 0e 7e 20 3e 06 0e 35 3d 12 70 94 bc 81 dd ac bd 83 46 11 3e ..
. 2433442 more bytes>
```

Figure 31. VALIDATORS VALIDATING TRAINER MODELS

Input   : All trainers shared their model updates

Output : A new model is released which aggregates the updates from all clients

Case 9 (Releasing New Model)



```
0059127808\n", stderr: "" }
Validation verdict: VALID
----------------------------------------
Sync Barrier signal received! Policy = 0
----------------------------------------
New Model Created:: version= 1   accuracy= 92.2%
----------------------------------------
Sync Barrier signal received! Policy = 0
Trainer <CallerAddress(Hash(6ce1ea88...))>, prev_score=0.5, val_score=0.922, new
_score=0.5, delta=0
Trainer <CallerAddress(Hash(40cfe4c6...))>, prev_score=0.5, val_score=0.9088, ne
w_score=0.48680001, delta=-0.013199985
Trainer <CallerAddress(Hash(40cfe4c6...))>, normalized_score=0.49331173
Trainer <CallerAddress(Hash(6ce1ea88...))>, normalized_score=0.5066883
----------------------------------------
New Model Created:: version= 2   accuracy= 85.97%
----------------------------------------
Sync Barrier signal received! Policy = 0
In new API
0.8597
In new API
0.922
Sync Barrier signal received! Policy = 0
```

```
, 77 more bytes
extension_ratio = NaN
62bcce747b65baa55cca6f25cc4972938669f2709505ef9b24c48397d568cf09
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 f5 89 f2 bf a6 a4 a4 8a 26 12 22
0a 20 0e f7 c3 ca 06 fa 2a e2 95 a9 dc c3 26 9f 86 99 fa f1 f4 c0 3e 58 47 d3 ..
. 76 more bytes>
extension_ratio = NaN
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 fd 9e c4 e7 e8 81 b0 d8 7b 12 22
0a 20 23 ae fd cb c6 23 99 49 69 14 1a 79 37 fd 0d 6f 16 fa f6 3c 4b 82 dd 0a ..
. 76 more bytes>
extension_ratio = 0
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 d3 b7 cb f7 cc 8c 94 f6 86 01 12
22 0a 20 a0 45 f0 91 aa fc a8 e3 40 02 39 7e f4 bb 9d f0 be c2 6b f6 14 f8 19 ..
. 77 more bytes>
extension_ratio = 0
1896d1261ec0d56eb734d06b9d8eb9933acc0c9ad1d8fad3226ac3f9362a9cf7
b49625bda260f562b5d15c84e8b83ecc76bf09530a28c80fce0317a6da4dec91
b9f1d7722e3ba3cf54a69725f86107c7f916a20ac5cdcc859aa2d7f791858d14
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 f2 df c7 a6 87 e4 91 94 15 12 22
0a 20 c6 24 cf 5b 66 ad ac 1b 2b 9b 05 07 24 46 8f f1 6c 6d ca 20 b5 97 c3 8a ..
. 76 more bytes>
extension_ratio = 1
17ba42ede9cb05cc50372b5a1ee31f3257be10bc78196e0c95b462fa7e9e141a
```

```
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is -1, target is 5
Curren version is 2, target is 5
```

```
http://127.0.0.1:
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=f3c53ffd3c36640adc527268e8bf1c24c8c5ea3a7c0e46564905f8115cd1fc02
No retrain quota at the moment, will retry in a bit
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=f3c53ffd3c36640adc527268e8bf1c24c8c5ea3a7c0e46564905f8115cd1fc02
No retrain quota at the moment, will retry in a bit
Will pause for 1 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=f3c53ffd3c36640adc527268e8bf1c24c8c5ea3a7c0e46564905f8115cd1fc02
No retrain quota at the moment, will retry in a bit
Will pause for 6 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
http://127.0.0.1:9000/api/services/ml_service/v1/trainer/retrain_quota?trainer_a
ddr=f3c53ffd3c36640adc527268e8bf1c24c8c5ea3a7c0e46564905f8115cd1fc02
No retrain quota at the moment, will retry in a bit
Will pause for 6 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #2
http://127.0.0.1:9000/api/services/ml_service/v1/models/getmodel?version=2
```

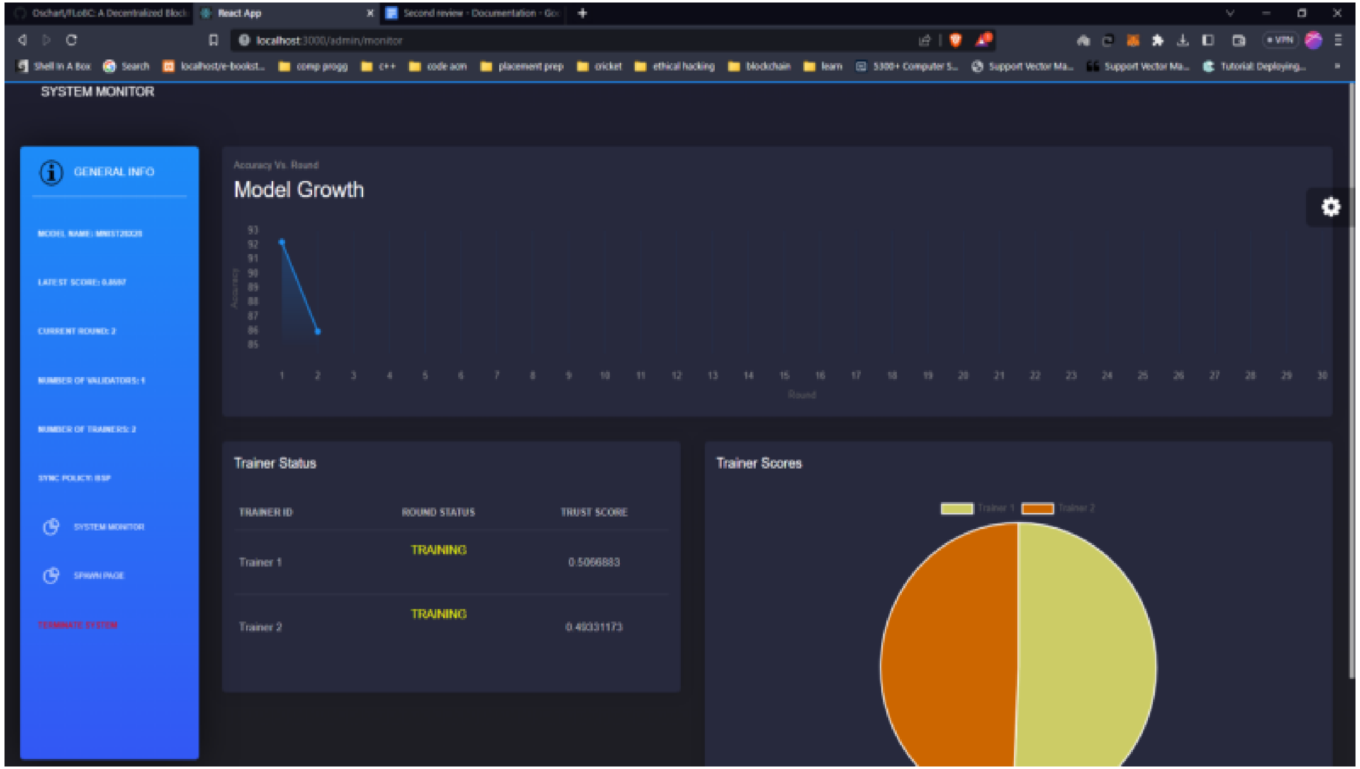
```
> babel src -d di                                            .csv" ".1"
 "MNIST28X28"

src/app.js -> dist/app.js
src/proto.js -> dist/proto.js
src/utils/fetchClientKeys.js -> dist/utils/fetchClientKeys.js
src/utils/fetchDatasetDirectory.js -> dist/utils/fetchDatasetDirectory.js
src/utils/fetchLatestModel.js -> dist/utils/fetchLatestModel.js
src/utils/fetchPythonWeights.js -> dist/utils/fetchPythonWeights.js
src/utils/generateNormalNoise.js -> dist/utils/generateNormalNoise.js
src/utils/parsePythonList.js -> dist/utils/parsePythonList.js
src/utils/store_encoded_vector.js -> dist/utils/store_encoded_vector.js
Will pause for 2 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #-1
<Buffer 0a e7 c2 94 01 0a e2 c2 94 01 0a 04 08 03 10 00 12 d7 c2 94 01 0a c8 c2
94 01 2a f3 2c 3e 8e 7e 20 3e 06 0e 35 3d 12 70 94 bc 81 dd ac bd 83 46 11 3e ..
. 2433442 more bytes>
8bcd5010bd498d843936ff1a9a76d593712b694d9625ff457c94f15be37ef78b
Will pause for 4 secs
http://127.0.0.1:9000/api/services/ml_service/v1/models/latestmodel
New model released by the validator!, #2
http://127.0.0.1:9000/api/services/ml_service/v1/models/getmodel?version=2
```

Figure 32. VALIDATORS RELESING THE NEW MODEL

Input : New model is released by the checker

Output : All clients will receive the new model update from the checker

Case 10 (Reflecting model update in frontend)


SYSTEM MONITOR

GENERAL INFO

MODEL NAME: MNIST2020

LATEST SCORE: 0.8597

CURRENT ROUND: 2

NUMBER OF VALIDATORS: 1

NUMBER OF TRAINERS: 2

SYNC POLICY: BSP

SYSTEM MONITOR

SPAWN PAGE

TERMINATE SYSTEM

Accuracy Vs. Round

Model Growth

Trainer Status

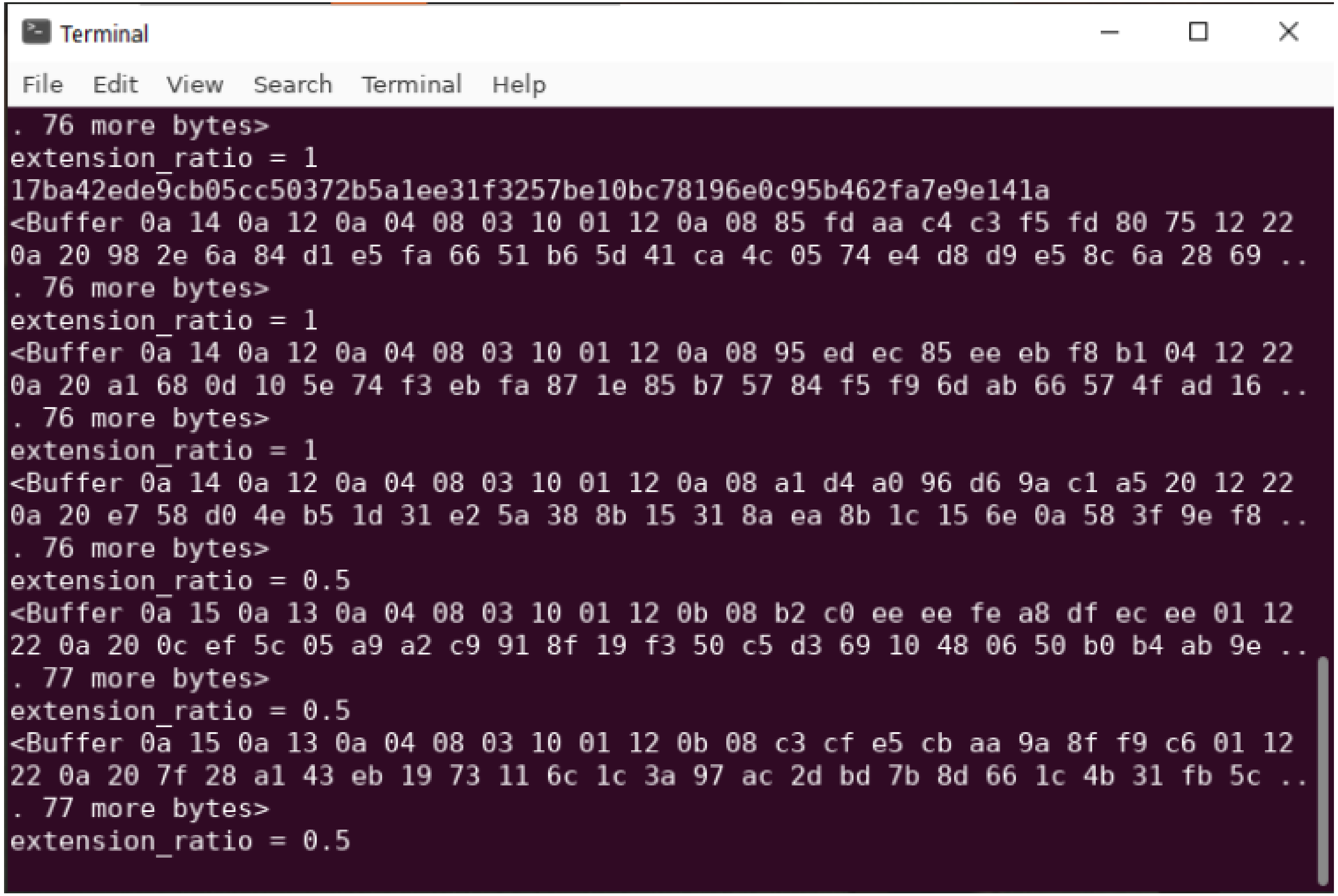
TRAINER ID	ROUND STATUS	TRUST SCORE
Trainer 1	TRAINING	0.5066883
Trainer 2	TRAINING	0.49331173

Trainer Scores

Figure 33. MODEL UPDATED IN FRONTEND

Input : New model is released

Output : New model metadata is updated in the frontend and in its buffer



```
. 76 more bytes>
extension_ratio = 1
17ba42ede9cb05cc50372b5a1ee31f3257be10bc78196e0c95b462fa7e9e141a
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 85 fd aa c4 c3 f5 fd 80 75 12 22
0a 20 98 2e 6a 84 d1 e5 fa 66 51 b6 5d 41 ca 4c 05 74 e4 d8 d9 e5 8c 6a 28 69 ..
. 76 more bytes>
extension_ratio = 1
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 95 ed ec 85 ee eb f8 b1 04 12 22
0a 20 a1 68 0d 10 5e 74 f3 eb fa 87 1e 85 b7 57 84 f5 f9 6d ab 66 57 4f ad 16 ..
. 76 more bytes>
extension_ratio = 1
<Buffer 0a 14 0a 12 0a 04 08 03 10 01 12 0a 08 a1 d4 a0 96 d6 9a c1 a5 20 12 22
0a 20 e7 58 d0 4e b5 1d 31 e2 5a 38 8b 15 31 8a ea 8b 1c 15 6e 0a 58 3f 9e f8 ..
. 76 more bytes>
extension_ratio = 0.5
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 b2 c0 ee ee fe a8 df ec ee 01 12
22 0a 20 0c ef 5c 05 a9 a2 c9 91 8f 19 f3 50 c5 d3 69 10 48 06 50 b0 b4 ab 9e ..
. 77 more bytes>
extension_ratio = 0.5
<Buffer 0a 15 0a 13 0a 04 08 03 10 01 12 0b 08 c3 cf e5 cb aa 9a 8f f9 c6 01 12
22 0a 20 7f 28 a1 43 eb 19 73 11 6c 1c 3a 97 ac 2d bd 7b 8d 66 1c 4b 31 fb 5c ..
. 77 more bytes>
extension_ratio = 0.5
```

CORRESPONDING TESTCASE BUFFER

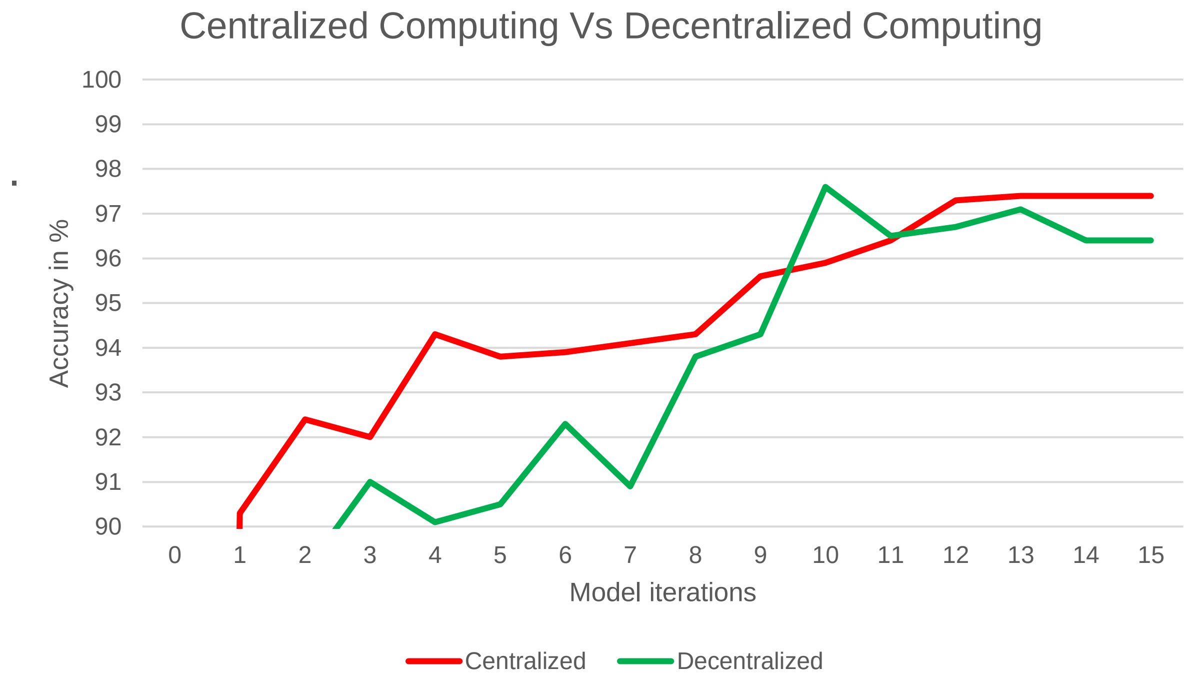
PERFORMANCE MEASURES

Benchmark: Decentralized vs. Centralized Performance

The purpose of this comparison is to check whether the security and privacy achieved from decentralized computing comes with a trade-off from loss of model quality and accuracy or whether model accuracy is unaffected by the advantages imposed by the decentralized model.

PERFORMANCE MEASURES

Benchmark: Decentralized vs. Centralized Performance



PERFORMANCE MEASURES

Benchmark: Decentralized vs. Centralized Performance

RESULT:

From the graph we can infer that with decentralized approach the model accuracy stays low as compared to centralized model for initial few numbers of iterations (6 – 7), it takes time to build up on the model accuracy as many decentralized actors would be contributing to build a model in each iteration with different datasets for them to train on. After initial few iterations we can infer that accuracy of centralized and decentralized model are in comparable with each other, and ends up with decentralized model having just 0.8% accuracy less than the centralized model. As for the benefits of security and privacy the decentralized model provides a very small percentage loss in accuracy is acceptable, also the trend can be subjected to change on different dataset or with more iterations where accuracy of both models would be similar.

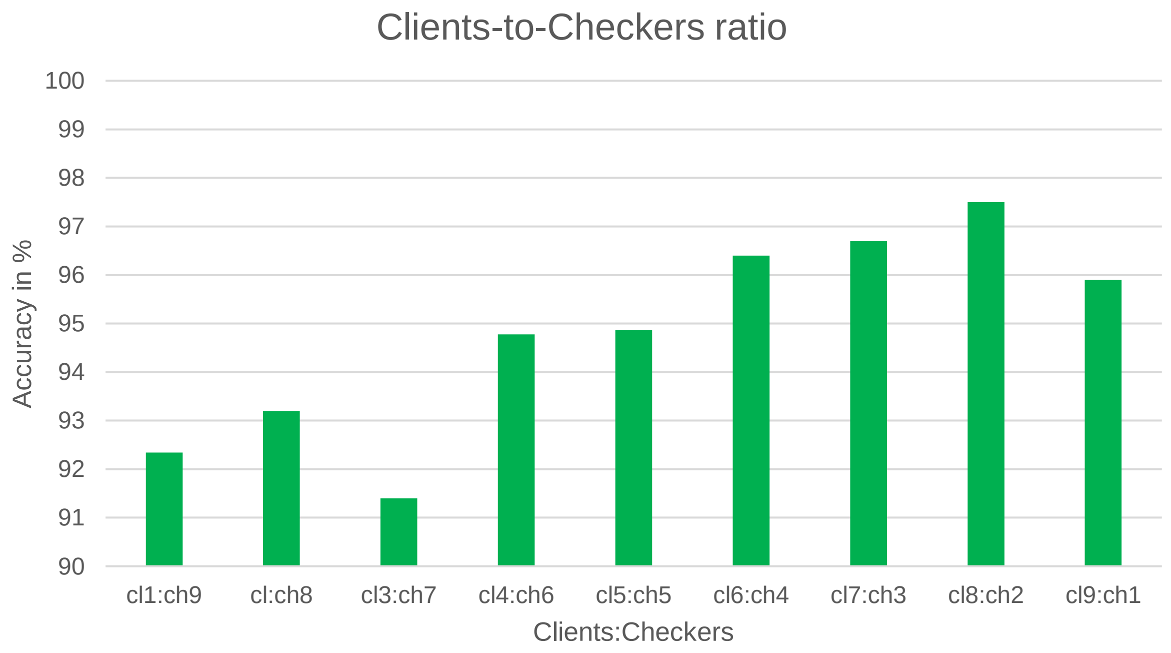
PERFORMANCE MEASURES

clients-to-checkers Ratio

The purpose of this comparison is to check whether the difference in number of clients and checkers affects model accuracy, if so then we have to estimate the ratio of clients and checkers which would lead us to higher accuracy model as compared to others.

PERFORMANCE MEASURES

clients-to-checkers Ratio



PERFORMANCE MEASURES

clients-to-checkers Ratio

RESULT:

As we can infer from the graph lower number of clients has very ow accuracy, because they are trained on very low number of dataset rows, thus leading to lowest accuracy. And as the clients increases the model is trained on diverse number of dataset instances thus accuracy increases. And during 9 clients and 1 checker there is a bottleneck in checkers side as it cannot validate all 9 model instances which decreases the accuracy eventually. Thus Clients: Checkers ratio 8:2 should be maintained in order to achieve higher accuracy model with maximum utilization of available resources.

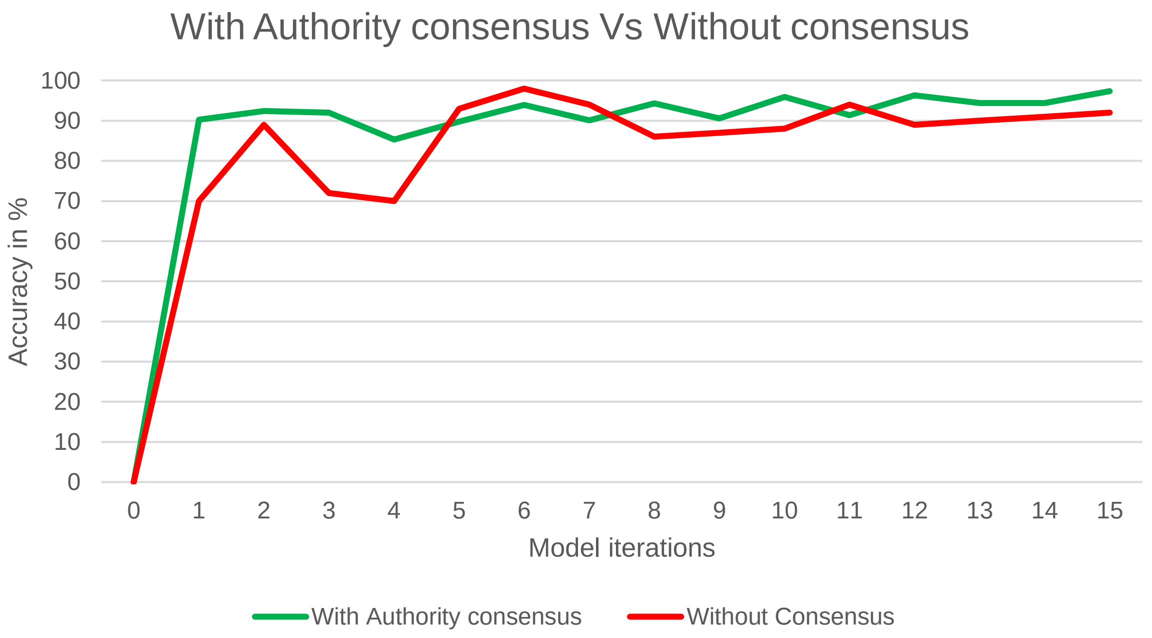
PERFORMANCE MEASURES

Authority consensus Policy

Another main component of D-FedL framework is Authority consensus which is responsible to make the overall system stable by avoiding model updates from the clients which constantly provides low quality model by penalising them and rewarding the clients with good quality model. The goal of the experiment is to compare the model growth with consensus and without it, and to infer whether usage of authority consensus increases model accuracy and stability.

PERFORMANCE MEASURES

Authority consensus Policy



PERFORMANCE MEASURES

Authority consensus Policy

RESULT:

As we could infer from the graph, we could see large deviation in accuracy of the model trained without consensus, because nodes that provide low quality model would eventually bring down the accuracy of the model thus more fluctuation in model accuracy. Whereas on other hand model trained with authority consensus had very low deviation in the accuracy in successive rounds because it will stop receiving model updates from low performing nodes which increases model stability and model accuracy. After 15 rounds accuracy of the model trained with consensus came out to be 97.4% and the model without consensus was 91.9%, as we could notice the huge 5.5% difference in model accuracy between models trained with and without consensus. Thus using authority consensus seems to have improved model accuracy and stability.

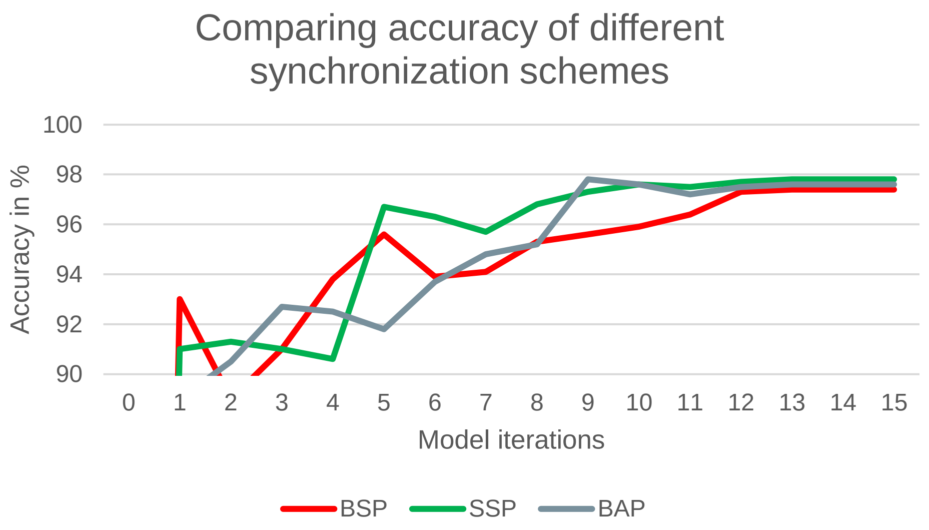
COMPARATIVE MEASURES

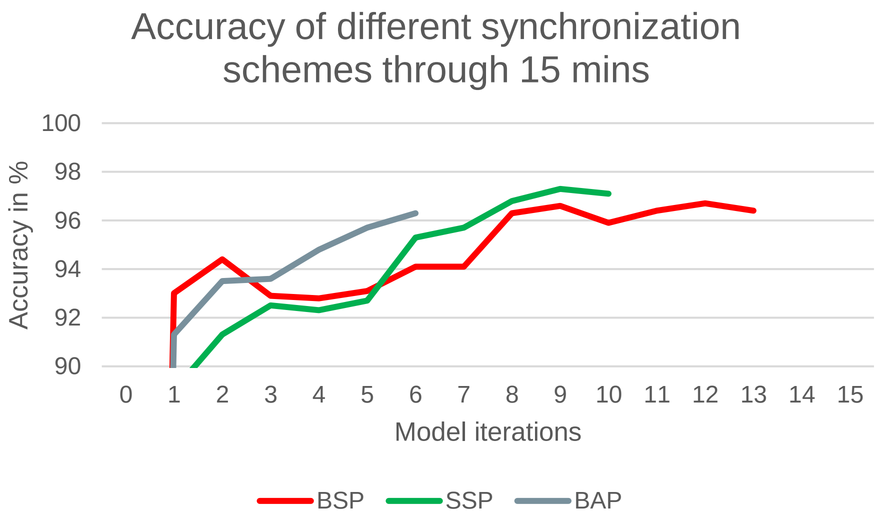
Synchronization Schemes

In this experiment, we compare some of the common synchronization schemes used in parallel computing after adapting our own variations of them. The schemes implemented in DFedL and those tested in the experiment are Bulk Synchronous Parallel (BSP), Stale Synchronous Parallel (SSP), and Barrierless Asynchronous Parallel (BAP), as previously described in the methodology.

COMPARATIVE MEASURES

Synchronization Schemes





COMPARATIVE MEASURES

Synchronization Schemes

RESULT:

- As we could infer from the graphs, at the end of 15 iterations the model accuracy of both synchronous and asynchronous schemes is almost same, but when we ran the system through particular time frame i.e) 15 minutes we could see BSP has higher number of iterations in 15 minutes but it has lower accuracy, followed by SSP which had completed 10 iterations in 15 minutes with higher accuracy and asynchronous scheme which has completed just 6 iterations.
- Though model accuracy is same at similar iteration, when it comes into the efficiency of using their resources synchronization schemes outperform asynchronous scheme by completing more iterations in fixed time frame.

.

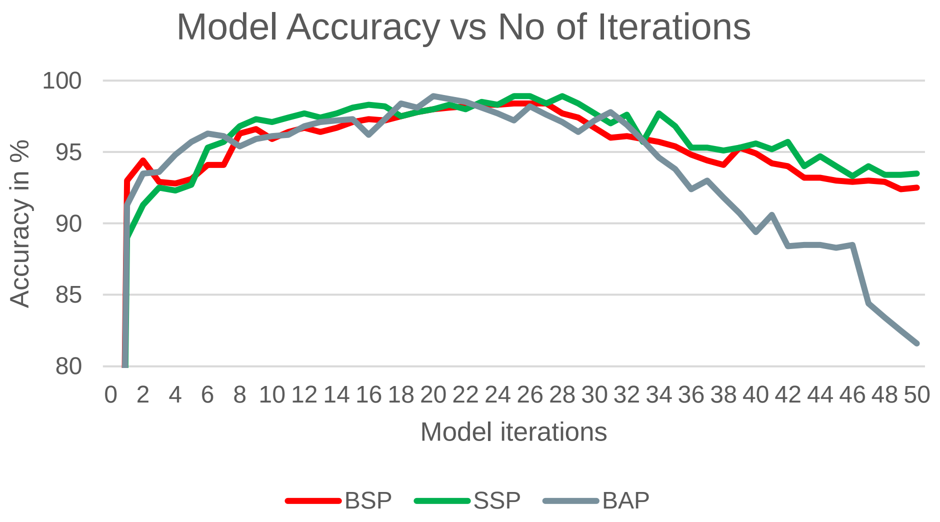
COMPARATIVE MEASURES

Model Accuracy vs No of Iterations

- From this experiment, the goal is to find whether the highest number of iterations till which the model is not affected by overfitting which decreases the model accuracy and find which synchronization scheme is more susceptible to overfitting and to find the one which is least affected by it.

COMPARATIVE MEASURES

Model Accuracy vs No of Iterations



Synchronization scheme	Iteration number
BSP	25
SSP	26
BAP	22

COMPARATIVE MEASURES

Model Accuracy vs No of Iterations

RESULT:

- From the table and graph, we can infer that the maximum iterations we can train without model overfitting is 30, after which the model accuracy gradually decreases. Also, from the graph we could find that BAP model accuracy is drastically affected by overfitting, while BSP and SSP can withstand the model overfitting to a certain limit. The conclusion from the experiment is in order to avoid overfitting the threshold no of iterations is 30 and synchronous scheme seems to withstand overfitting to a certain margin..

CONCLUSION

- D-FedL is a proof-of-concept that demonstrates the ability of its constituent parts to work together to create a coherent system with the requisite levels of generality, decentralization, efficiency, privacy, and Byzantine fault-tolerance.
- The decentralized model has just 0.8% less accuracy than the centralized model.
- There is a quasi-optimal balance to be struck on client-to-checker partitioning, empirically establishing that an 8:2 client-to-checker ratio works well. Second, even a straightforward authority consensus approach may significantly improve the caliber of models that are created.
- Finally, three main synchronization systems (BSP, SSP, and BAP). SSP has been shown to be more economical than the other two approaches. Also, the maximum no of iterations that can run without overfitting is 30 iterations (threshold number), and BSP, SSP (synchronous schemes) can withstand overfitting to certain extent, but BAP (asynchronous scheme) found very vulnerable to overfitting.

FUTURE WORK

D-FedL may easily be modified to support additional computational models even though the SGD-compatibility of a model is an assumption built into the system. By giving the sync periods greater flexibility based on how long it is anticipated that most clients will take to submit their updates in a single round, synchronisation schemes can be further improved. Additionally, a layer of differential privacy might be added to transaction communication to restrict how far gradients can be inverted in order to extract client data, which would provide a better level of data privacy.

REFERENCES

[1] Anna Bogdanova, Akie Nakai, Yukihiko Okada, Akira Imakura, Tetsuya Sakurai, “Federated Learning System without Model Sharing through Integration of Dimensional Reduced Data Representations”, November 2020

[2] Besir Kurtulmus, Daniel Kenny, “Trustless Machine Learning Contracts; Evaluating and Exchanging Machine Learning Models on the Ethereum Blockchain”, February 2018

[3] Isitan Görkey, Chakir El Moussaoui, Vincent Wijdeveld, Erik Sennema, “Comparative Study of Byzantine Fault Tolerant Consensus Algorithms on Permissioned Blockchains”, April 2020

[4] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, Dave Bacon, “Federated Learning: Strategies for Improving Communication Efficiency”, October 2017

[5] Jianping He, Lin Cai, Peng Cheng, Jianping Pan, Ling Shi, “Consensus-Based Data-Privacy Preserving Data Aggregation”, April 2019

REFERENCES

[6] Marco Benedetti, Francesco De Sclavis, Marco Favorito, Giuseppe Galano, Sara Giammusso, Antonio Muci, Matteo Nardelli, “A PoW-less Bitcoin with Certified Byzantine Consensus”, July 2022

[7] Micah J. Sheller, Brandon Edwards, G. Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R. Colen, Spyridon Bakas, “Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data”, July 2020

[8] Sepideh Avizheh, Mahmudun Nabi, Saoreen Rahman, Setareh Sharifian, Reihaneh Safavi-Naini, “Privacy-Preserving Resource Sharing Using Permissioned Blockchains”, September 2021

[9] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, Yi Zhou, “A Hybrid Approach to Privacy-Preserving Federated Learning”, November 2019

[10] Youyang Qu, Md Palash Uddin, Chenquan Gan, Yong Xiang, Longxiang Gao, John Yearwood, “Blockchain- enabled Federated Learning: A Survey”, April 2023

THANKYOU