User Guide for MobiCharged

Team Super Charged (No.33)
Nashit Mohammad - mohamn31
Eric Nguyen - nguyee13
Samuel De Haan - dehaas1
Eamon Earl - earle2
Mustafa Choueib - choueibm

April, 2023

Contents

List of Figures

List of Tables

1 Revision History

Table 1: **Revision History**

Author	Date	Version	Description
Mustafa Choueib	April 4th, 2023	Rev 0	Created first draft of document

2 Legal and Copyright Information

This version of the MobiCharged software (”Software Product”), any documentation, or other forms of intellectual property are reserved by the Computing and Software Department of McMaster University, MobiCharged development team, and Mobilite Power Inc. The MobiCharged Software is protected by copyright laws and treaties, and other laws governing different forms of intellectual property. These are the Terms and Conditions governing the use of this Service and the agreement that operates between You and all Affiliates of this Software Product.

2.1 Acceptance

By installing, using, or accessing the Service You agree to be bound by these Terms and Conditions. If You fail to agree with ALL of the Terms and Conditions presented in this document, You may not use this Service.

2.2 Prohibitions

You are prohibited from:

- Violating or attempting to violate the security of this Software Product.
- Violate any law, rule, or regulation while using this Software Product.
- Altering or attempting to alter, decompile, derive, or reverse engineer the Source code of this Software Product.

- Obtain or attempting to obtain unauthorized access to this Software Product.
- Using or attempting to use any device, software, or routine to interfere or attempt to interfere with the proper working of this Software Product.
- Using or attempting to use any device, software, or routine to interfere, access, alter, or corrupt or attempt to interfere, access, alter, or corrupt any data in the database portion of this Software Product.

2.3 Breach of Contract

If You violate ANY of these rules, MobiCharged Affiliates may take action against You. This includes possible legal persecution, revoked access to this Software Product or any other Software Products provided by the MobiCharged development team.

2.4 Governing Law

This Agreement is governed by the laws of the Province of Ontario.

2.5 Exclusion of Liability

TO THE FULLEST EXTENT PERMITTED BY LAW, ALL AFFILIATES OF THE MOBICHARGED SOFTWARE PRODUCT WILL NOT BE LIABLE IN CONNECTION WITH THIS CONTRACT FOR LOST PROFITS, BUSINESS OPPORTUNITIES, REPUTATION, DATA, OR ANY INDIRECT, INCIDENTAL, CONSEQUENTIAL, SPECIAL, OR PUNITIVE DAMAGES. YOU AGREE THAT YOU ARE MAKING USE OF THIS SOFTWARE PRODUCT AT YOUR OWN RISK AND THAT IT IS PROVIDED ON AN ”AS IS” BASIS. TO THE EXTENT PERMITTED BY LAW, WE EXCLUDE ALL EXPRESS OR IMPLIED WARRANTIES, TERMS AND CONDITIONS INCLUDING, BUT NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS, AND NON-INFRINGEMENT.

3 Introduction

The MobiCharged Product consists of a Server-Client portion that autonomously runs simulations and stores data in a database, a machine learning black board that uses the data to be trained and provide optimal configurations, and a hardware system that aids in optimizing the algorithms in a real world environment. This User's guide contains an overview on the proper installation and use of the MobiCharged software and hardware product.

Note: The Server-Client portion is run as a background process in order to generate data with specific configurations (determined by the user) that will be used to train a specific model of the machine learner.

4 Installation

4.1 Software Installation

4.1.1 Installing the Server

As the Server is run on a remote host, the dependencies required will already be installed on that system. Only the Tkinter external package is required for the server configuration initializer, which can be installed by running ”pip install tk” or ”python.exe -m pip install tk” in the command line. Steps to connect to the remote host and launch the MobiCharged server are available in the Usage section of this document.

1. Install any version of PuTTY (third party software to connect to remote hosts)

2. Under ”Host Name (or IP address)” enter ”172.105.14.186” and keep the port as 22

3. Provide a name and save the connection (to avoid entering the address each time)

4. Enter the credentials provided by the MobiCharged team and launch the ”launch_server.bat” by using the ”./launch_server.bat” command in the console

5. Ensure you are prompted with ”MobiCharged server is now running....”

4.1.2 Installing the Client

The client uses many different files and external dependencies in order to run correctly. Thus, on your machine you must ensure the following items are installed:

1. MATLAB (versions other than R2023a may cause issues)
2. MATLAB Engine (refer to: MATLAB Engine Instllation Guide)
3. Download the MobiCharged client files

4.1.3 Installing the Machine Learner

The Machine Learner uses the packages required to install the client and some additional external packages. The packages that must be installed on your machine are:

1. Tkinter (can be installed by running ”pip install tk” or ”python.exe -m pip install tk” in the command line)
2. Tenserflow (Refer to: Tenserflow installation with pip)
3. Firebase (Can be installed using ”pip install python-firebase” in the command line. Refer to: Firebase for Python for more information)
4. MATLAB (versions other than R2023a may cause issues)
5. MATLAB Engine (refer to: MATLAB Engine Instllation Guide)
6. The MobiCharged Machine Learner Black board package

4.2 Hardware Installation

The hardware system requires a small amount of setup/installation before it is ready to be used. To install the hardware system, you must do the following:

1. Download the phasedarray.ino file from the MobiCharged Repository
2. Download and Install the Arduino IDE (Refer to: Arduino IDE)
3. Have an Arduino system connected to your machine, and upload the phasedarray.ino file to the Arduino through the IDE

5 Usage

5.1 Using the Server-Client Application

Note: The Server-Client Application does not need to be running to run the Machine Learner Black Board or use the hardware system.

5.1.1 Configuring the MobiCharged Server

To configure the server prior to launch, you should locate your MobiCharged server files and run the mobicharged_initializer.exe to start the initializer application. If you have trouble launching the initializer with the .exe file, it can also be run through the command line by navigating to the path of the mobicharged_initializer.py file and using the command ”python mobicharged_initializer.py”.

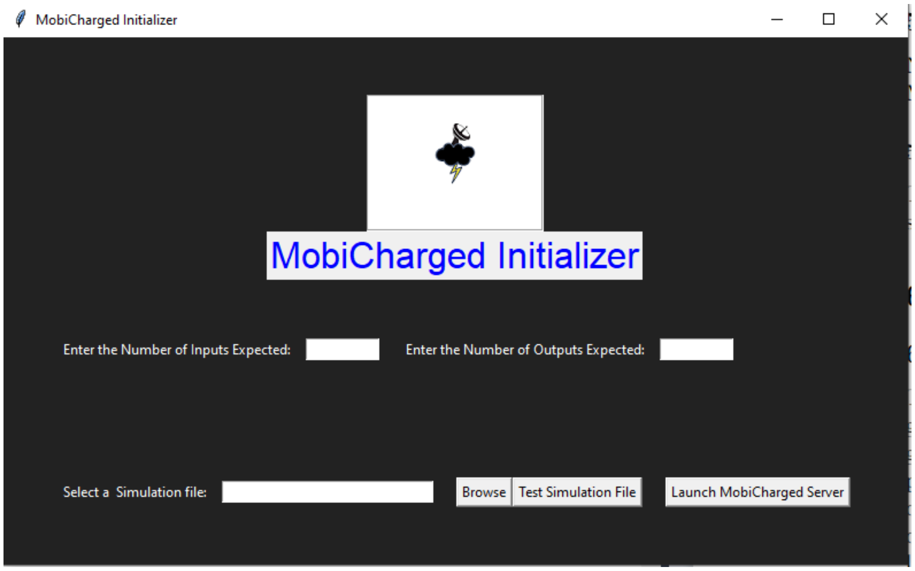
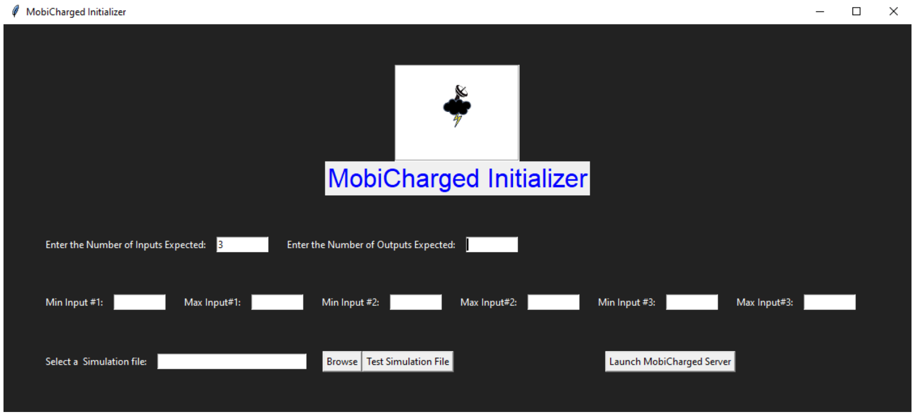

MobiCharged Initializer
MobiCharged Initializer
Enter the Number of Inputs Expected:
Enter the Number of Outputs Expected:
Select a Simulation file:
Browse
Test Simulation File
Launch MobiCharged Server

Figure 1: MobiCharged Initializer

Once the initializer is open, begin filling out the input boxes with your desired specifications. Once you enter the number of inputs expected (first entry), more fields will be generated asking for specific ranges to be used in generating random input (for autonomous data generation).



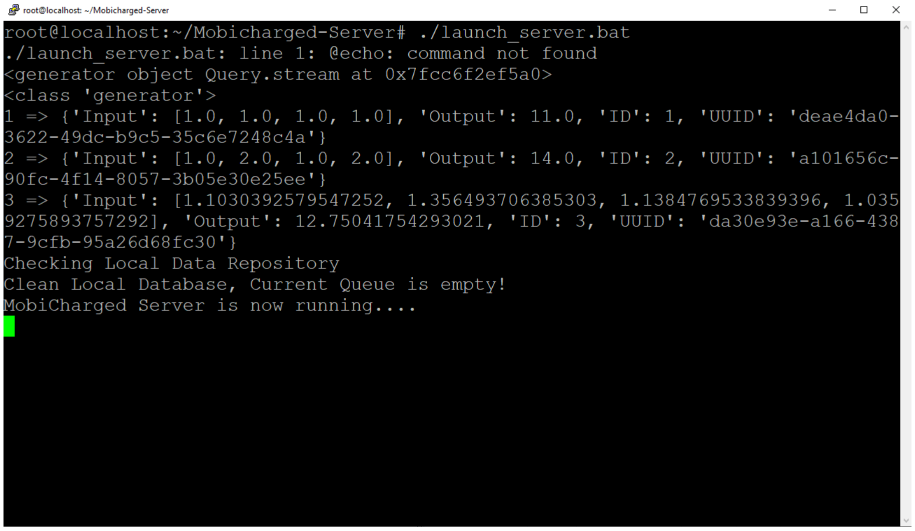
Enter the Number of Inputs Expected: 3

Figure 2: MobiCharged Initializer

You will also be asked to select a specific MATLAB simulation file and run a quick test using the inputs provided to ensure that the desired configuration is compatible with the desired simulation file. Once all the fields have been filled and a MATLAB simulation file has been selected and tested (test must be successful, otherwise you will not be able to launch the server), press the "Launch MobiCharged Server" button to create a server with the given specifications.

5.1.2 Starting the Configured Server

To start the server that was just configured, open the third-party application PuTTY and connect to the remote host specified in the Installation section of this document. You will be prompted to enter a username and password, which authorized users should have received. Upon providing the correct credentials, a file named "launch_server.bat" will exist providing all the specifications from the initializer. Use the command './launch_server.bat" to start the server.

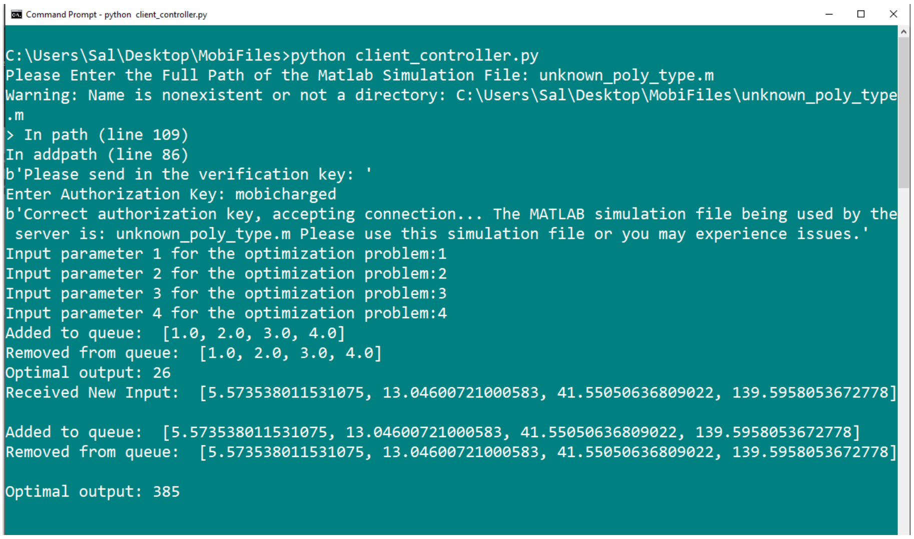


```
root@localhost:~/Mobicharged-Server# ./launch_server.bat
./launch_server.bat: line 1: @echo: command not found
<generator object Query.stream at 0x7fcc6f2ef5a0>
<class 'generator'>
1 => {'Input': [1.0, 1.0, 1.0, 1.0], 'Output': 11.0, 'ID': 1, 'UUID': 'deae4da0-
3622-49dc-b9c5-35c6e7248c4a'}
2 => {'Input': [1.0, 2.0, 1.0, 2.0], 'Output': 14.0, 'ID': 2, 'UUID': 'a101656c-
90fc-4f14-8057-3b05e30e25ee'}
3 => {'Input': [1.1030392579547252, 1.356493706385303, 1.1384769533839396, 1.035
9275893757292], 'Output': 12.75041754293021, 'ID': 3, 'UUID': 'da30e93e-a166-438
7-9cfb-95a26d68fc30'}
Checking Local Data Repository
Clean Local Database, Current Queue is empty!
MobiCharged Server is now running....
```

Figure 3: Remote Host

5.1.3 Starting the Clients

The clients can be run from any machine, given it was installed correctly. Thus, download the client files from the MobiCharged repository and run the ”client_controller.py” file using the command ”python client_controller.py”. The simulation file the server is expecting will be displayed to the client, and the client will enter the path to a simulation file (same simulation file should be used to avoid conflict). From there, the client will request more input then it will begin the autonomous cycle with the server.

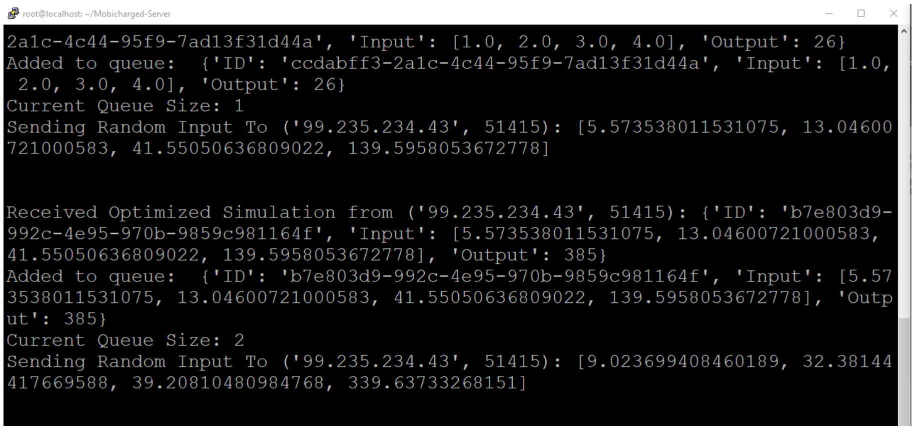


```
C:\Users\Sal\Desktop\MobiFiles>python client_controller.py
Please Enter the Full Path of the Matlab Simulation File: unknown_poly_type.m
Warning: Name is nonexistent or not a directory: C:\Users\Sal\Desktop\MobiFiles\unknown_poly_type
.m
> In path (line 109)
In addpath (line 86)
b'Please send in the verification key: '
Enter Authorization Key: mobicharged
b'Correct authorization key, accepting connection... The MATLAB simulation file being used by the
 server is: unknown_poly_type.m Please use this simulation file or you may experience issues.'
Input parameter 1 for the optimization problem:1
Input parameter 2 for the optimization problem:2
Input parameter 3 for the optimization problem:3
Input parameter 4 for the optimization problem:4
Added to queue:  [1.0, 2.0, 3.0, 4.0]
Removed from queue:  [1.0, 2.0, 3.0, 4.0]
Optimal output: 26
Received New Input:  [5.573538011531075, 13.04600721000583, 41.55050636809022, 139.5958053672778]

Added to queue:  [5.573538011531075, 13.04600721000583, 41.55050636809022, 139.5958053672778]
Removed from queue:  [5.573538011531075, 13.04600721000583, 41.55050636809022, 139.5958053672778]

Optimal output: 385
```

Figure 4: Client Side Application



```
2a1c-4c44-95f9-7ad13f31d44a', 'Input': [1.0, 2.0, 3.0, 4.0], 'Output': 26}
Added to queue:  {'ID': 'ccdabff3-2a1c-4c44-95f9-7ad13f31d44a', 'Input': [1.0,
 2.0, 3.0, 4.0], 'Output': 26}
Current Queue Size: 1
Sending Random Input To ('99.235.234.43', 51415): [5.573538011531075, 13.04600
721000583, 41.55050636809022, 139.5958053672778]


Received Optimized Simulation from ('99.235.234.43', 51415): {'ID': 'b7e803d9-
992c-4e95-970b-9859c981164f', 'Input': [5.573538011531075, 13.04600721000583,
41.55050636809022, 139.5958053672778], 'Output': 385}
Added to queue:  {'ID': 'b7e803d9-992c-4e95-970b-9859c981164f', 'Input': [5.57
3538011531075, 13.04600721000583, 41.55050636809022, 139.5958053672778], 'Outp
ut': 385}
Current Queue Size: 2
Sending Random Input To ('99.235.234.43', 51415): [9.023699408460189, 32.38144
417669588, 39.20810480984768, 339.63733268151]
```

Figure 5: Server Side Application

The first input parameters are just initial random parameters, after this, the server and client should interact with each other and generate data without any user involvement. As you can see from figure 5, the Server will send back ”Random Input” and the client will calculate optimal output for that input. This cycle will continue until stopped by the user.

5.2 Using the Machine Learner Black Board

To initialize the Machine Learning Blackboard system, we first want to run the src/reset.py script. This will ensure that the system state is correct and ready to be started anew. This script will ask you to confirm in the terminal with a ”Y” that you wish to reset the system state. This should be done every time you want to rerun the Blackboard with a clean slate (without any current best model permeated and without any under-performing models being labeled invalid).

From there, we would ask that you run the MLB_GUI.py script. This will initialize the front end of the system. From there, you can click on the ”ML Blackboard start” button. This will begin the process of pitting the models against each other. Note that the GUI will be mostly unresponsive until convergence. Upon convergence on a single best model, the GUI will present a graph representing its error progression throughout the run that yielded the model our ”locally optimal” weights. We should also expect the first table to be populated with the iteration number, the winning model, the average absolute error, and so on. From there, you can click on the button that says ”unknown_poly_type”, and then on the subsequent button of the same name (this will have more options if we have multiple simulation files available to be tested against). From here, you will see a table render below the error graph, where you can input your data points you wish to predict on. For the time being, we are also computing the real outputs of these inputs and calculating the actual error to compare it to the expected error.

This process can be repeated any number of times, and the ”ML Blackboard start” button can be clicked again as well. Expect subsequent training to be faster than the initial one, and that once converged naturally (not stopped prematurely), the ML Blackboard process will have found a sole victor and the other models that were initially there will not be trained on. If you wish to train on them again, you can run the reset.py script as previously discussed, though this will also reset our prior knowledge of our best contender, or if you want you could manually move the models from src/invalid_models to src/valid_models to keep the knowledge of our best model. Feel free to close and rerun the GUI at any point in the process.

The Blackboard has been preloaded with two naive models, for the sake of exemplifying the product in a controlled manner.

5.3 Using the Hardware System

To use the hardware system, start by connecting the array power source to a 120V supply. Assuming installation was done correctly, once connected to a 120V supply, turn the power switch on to provide power to the array. Once the hardware system is powered on, use tweezers to insert the desired particle for levitation. The particle inserted should be levitating and can be controlled by closing the "up switch" or "down switch" to move the particle vertically, or close the "reset switch" to return the particle to the centre.

6 Troubleshooting

6.1 Server Initializer Errors

Error: MATLAB Simulation Failed
This error means that the selected MATLAB simulation file is not compatible with the specified parameters. The user must ensure that the specified parameters is compatible with the MATLAB simulation file and run the test again to be able to launch the server.

6.2 Server Errors

Server sided errors are displayed in the console of the remote host. Here is a list of potential errors:

Error: Failed to Launch Server
This error can occur for many reasons; manual modifications to the local database, invalid specifications, corrupted launch_server.bat file. The solution to this would be to reconfigure the server, and launch it.

6.3 Client Errors

Client sided errors are displayed in the console of the users machine. Here is a list of potential errors:

Error: MATLAB Engine Error
If you see this error, either the specified inputs or the chosen MATLAB simu-

lation files are not compatible, thus, cannot produce a result from MATLAB Engine.

Error: Connection Refused Error
If you see this error, that means the client could not connect to the server. This is typically because the server is not launched and the client is trying to connect.

6.4 Machine Learner Errors

Ensure that you have run reset.py and confirmed the reset in the terminal before running the main loop.

MOST IMPORTANT TROUBLESHOOT: Ensure that you are running the files from WITHIN the MobiCharged folder, such that MobiCharged is our parent folder and we are in the same directory as the src folder. We were having issues with absolute pathing, and noted that if you run it with src as your root file you will encounter pathing errors. This will be remedied and standardized at launch.

A possible pseudo-error state is that the Blackboard is taking an eerily long time to converge. And as such the option to predict does not open for some time. With the two given models, and the likely small database size we will be providing for marking, you should expect convergence within 5 minutes. If this does not occur, check the terminal. If the training errors being reported there are infinite or NaN, then our input data is ill-defined and it is unlikely that the Blackboard will converge. If this happens, reach out to us and we will go about re-generating some data that plays nicer with our naive machine learning models.

It is unlikely that we will be generating great deals of data, so do not expect high accuracy from these models. If, however, we do have a very full database, there is a possibility of a memory error occuring, in which case you can once again reach out to us and we will slash down our database, as I think this is easier than asking you to do something on your end.

7 Frequently Asked Questions and Answers

Question: What can be levitated?
Answer: An object whose density is less than 4g/cm (cubed). This includes

liquids and solids. Objects of uniform shape (spheres) are more stable.

Question: How long can the array run?
Answer: The array can run as long as any of the electrical components remain at a reasonable temperature. Results have show this to be greater than 2 hours.

Question: How many clients can connect to the server?
Answer: There is no limitation on the number of clients that connect to a server. The server was tested with over 25 clients connected at once and experienced little to no loss in performance.

Question: Can I run more than one client on a machine?
Answer: Yes, you can run an unlimited amount of clients on one machine, or many machines.

Question: What happens if the server or client are interrupted during data transmissions?
Answer: The Server-Client application was designed to prevent any loss or corruption in data, thus, should be fine once either the client or server are restored.

References

[1] Hartley, C. (2023, February 18). User agreement template. TermsFeed. Retrieved April 5, 2023, from https://www.termsfeed.com/blog/sample-user-agreement-template/
Any other sources refer to documentation provided by Mobilite-Power.