

# Final Report

Ishmam Kabir and Troy Witmer

2025-05-01

## 1 Objective

Our initial Objective was to use openars to predict future data points on a candlestick graph based on historical data. This uses open, high low, and close data points on each candlestick to predict the next few. This project uses stock data because of its historical availability, accuracy. However due to unforeseen circumstances we could not implement all candlestick data such as high, low, and open, instead we settled for close and changed from a candlestick graph to a line graph. Due to errors in the python version of OpenNARS our program was cut short and was not able to be fully implemented.

## 2 Problems

During our development we ran into two big problems, first was the prediction rate vs the rate we were sending data to NARS, second was the babbling feature was the only data we were receiving due to issues in the source code. These problems appeared to be separate at first with our minds focusing on figuring out how we could get NARS to execute a sensorimotor command everytime we sent a sentence to it. After hours of failing to come up with a solution we stumbled across the source code for the SensorimotorChannel which contained bypasses, which according to the comments stated an issue with the Reasoner. After much time debugging, we were unable to find a fix for the issue. As a result of the Reasoner being broken all commands sent from the SensorimotorChannel were random due to the babbling feature.

Here is snippets found inside the source code of OpenNARS Python.

```
1 // cheating
2 // since something is wrong the reasoner , such that it
   cannot generate sub-goals correctly , I have to cheat .
3
4 reactions , ret = self.input_buffer.buffer_cycle(inputs ,
   memory)
5 for each in reactions:
```

```

6     self.reactions.push(each, self.reaction_evaluation(each
    , memory))
7
8 return ret
9 // original
10 // return self.input_buffer.buffer_cycle(inputs, memory)
11
12
13 // cheating
14 // since something is wrong the reasoner, such that it
    cannot generate sub-goals correctly, I have to cheat.
15 // this means there are reactions with no corresponding
    goals
16 if reaction.goal is None:
17     return 0.7

```

### 3 Results

Dispite the issues we still wrote most of the code required to complete our Objective. Our how we choose truth values, what we sent to NARS

