Predicting Ethereum Next-Day Price Movement

Andrej Elez[1], Eastan Giebler[2], Koren Gilbai[3], Andrei Mandelshtam[4]

{aelez[1], eastan[2], kgilbai[3], andman[4]}@stanford.edu

Department of Computer Science, Stanford University

Stanford
Department of Computer Science

Abstract

This project analyzes the price dynamics of Ethereum (ETH), a cryptocurrency enabling the emerging ecosystems of decentralized finance (“defi”) and Web3. Individual investors who have increasingly adopted the cryptocurrency may lack the infrastructure and resources of large hedge funds and trading firms. These investors would benefit from an accurate tool to predict changes in the next-day price of ETH. Furthermore, knowing which variables are most determinative of the future price of ETH is relevant to members of the Ethereum community who are thinking about how to improve or otherwise change the protocol configuration.

We constructed LSTM and GRU models to predict the next day price of ETH given past price and other feature data. The models achieved, in the best case, a root mean squared error of 0.042763 on the test set.

Stanford University

Dataset and Features

To retrieve data from the blockchain, we utilized the open-source API “Blockchain ETL” and ran SQL queries on the public “crypto_ethereum" dataset hosted on Google BigQuery.

The dataset includes **2463** examples, each corresponding to a single, discrete day between 08/08/2015 and 05/05/2022. We extracted a total of 118 features, and used **17** real-valued features in the the models whose results are shown here. Our hand-designed features include:

- Number of active public keys: a proxy for engagement on the Ethereum ecosystem.
- Hash rate and number of new miners: related to the production of new blocks, affecting the supply of ETH.
- Velocity (“rate of turnover”) and Total Value Locked (volume of “defi”): reflect the adoption of Ethereum's practical utilities and financial products.

Index	Feature Name
1	ETH Price
2	ERC-20 Unique Address Count
3	Average Txn Fee (USD)
4	Transaction Count
5	Gas Used
6	Hash Rate
7	Uncle Count
8	Velocity, Current Supply, 1 Yr
...	...
17	Block Difficulty

Baseline Models

Logistic Regression

Standard logistic regression on the classification task (next-day price ⇅)

1. With unmodified input features, logistic regression trained poorly
2. Adding feature engineering improved performance. Accuracy only slightly better than average - **53.5%**
3. Simple neural network has an accuracy of **55.5%**

Logistic regression lacks the expressivity to classify next-day movement at a rate significantly better than a coin toss.

Methods

Feature Engineering

1. Add intercept
2. Min-max Normalization $$x' = \frac{x - min(x)}{max(x) - min(x)}$$
3. Z-norm standardization $$x' = \frac{x - average(x)}{\sigma}$$

Recurrent Neural Networks

- Necessary to use a highly nonlinear model
- A time-dependent approach is also desirable
- Recurrent neural networks come to the rescue - each timestep admits its own cell state:

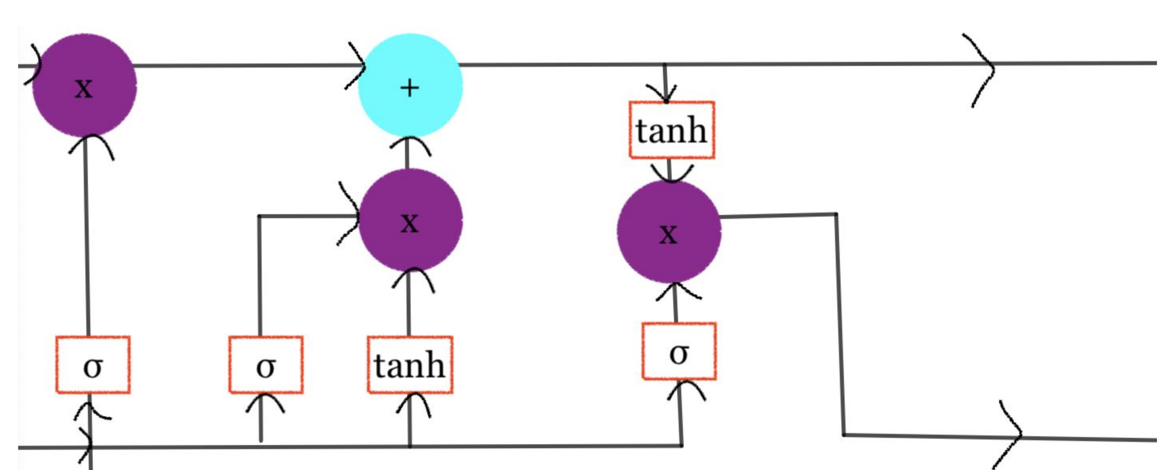


Fig: Long-Short Term Memory Time Cell

- Two types of RNNs investigated - GRUs (Gated Recurrent Units) and LSTMs (above)
- GRUs work similarly but combine the actions of forget and input.

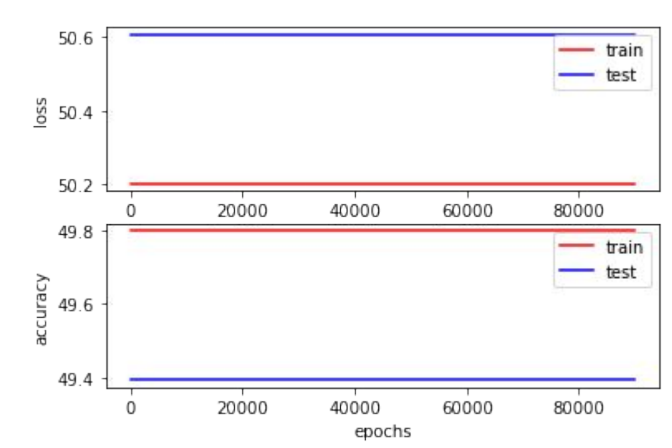


Fig **1**: Logistic Regression with unmodified features

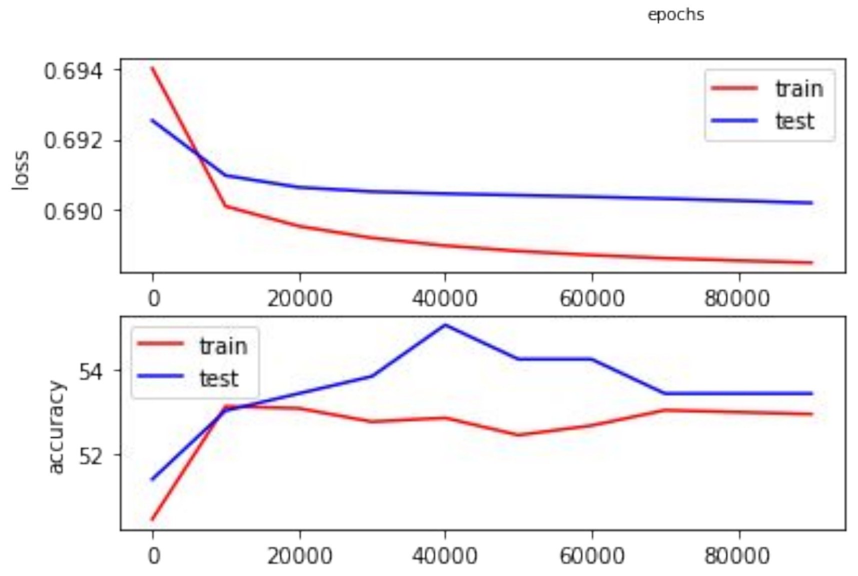


Fig **2**: Simple Log. Reg. with feature engineering

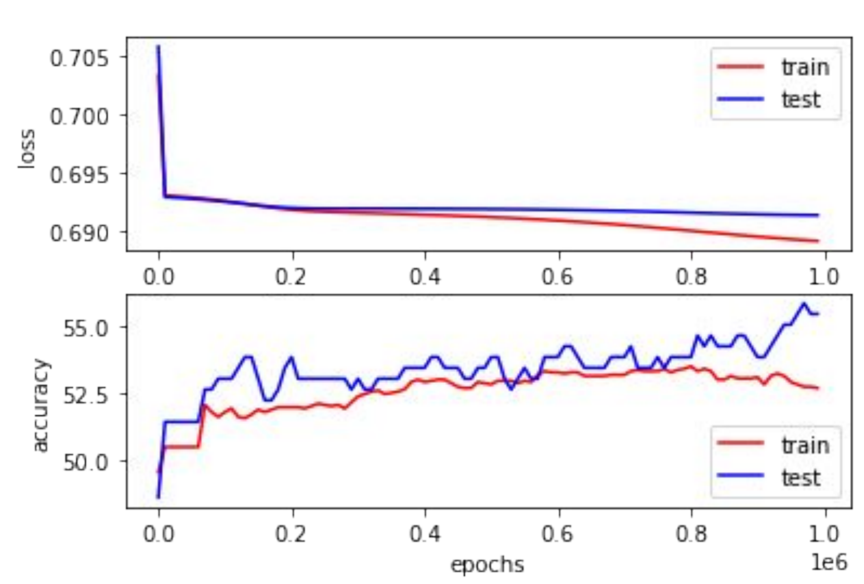


Fig **3**: 3 layer NN (log reg with 2 hidden layers)

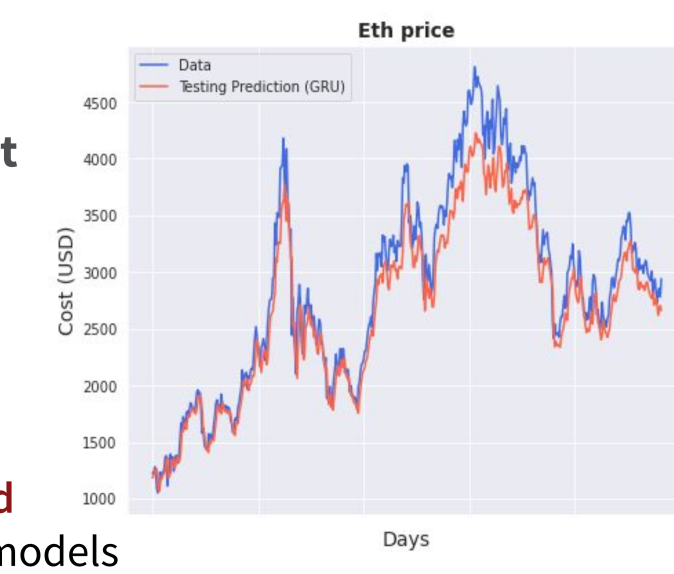
Results

Model	Training RMSE (10^-3)	Testing RMSE (10^-3)
Single Feature (Price) LSTM	4.229	67.936
Single Feature (Price) GRU	4.427	52.768
Multivariate LSTM	3.604	42.763
Multivariate GRU	3.983	52.473

Price Only LSTM

Multivariate LSTM

Price Only GRU



Multivariate GRU

Discussion

- GRU trains faster and **performs better** than LSTM on **less training data**
- LSTM is more accurate with a **larger input dataset**

Hyper Parameters

Lookback:

- Much higher for single feature RNN models
- Suggests that the **single feature RNNs needed more time series data** than the multivariate models to improve prediction accuracy
- As the # of features increases, less expressive meaning is needed from the time series memory

RNN Layers:

- Additional RNN layer improved performance for single feature RNN models
- **Single feature** models needed **more RNN layers** to extract more meaningful patterns from the past price feature for the regression task

Learning Rate:

- **GRU converges more rapidly than the LSTM**, the smaller learning rates for the GRU models prevents divergence to suboptimal solutions.

Parameter	Single Feature LSTM	Single Feature GRU	Multivariate LSTM	Multivariate GRU
Lookback (days)	50	50	10	10
Hidden Dimension	32	32	48	48
# of RNN layers	2	2	1	1
Epochs	1000	1000	2000	2000
Learning Rate	0.005	0.002	0.002	0.001

Conclusions and Future Work

- Our multivariate LSTM algorithm was the highest performing, which we expected because of the timelines inherent in price prediction problems

Future

- Hyperparameter Tuning: perform an exhaustive grid search using Scikit-learn's grid search package
- Directly examine the model parameters to better understand the important features
- Include more features from the scraped data

Similar Works:

[1] Han-Min Kim, Gee-Woo Bock, and Gunwoong Lee. Predicting ethereum prices with machine learning based on blockchain information. Expert Systems with Applications, 184:115480, 2021.
[2] Mohammad J. Hamayel and Amani Yousef Owda. A novel cryptocurrency price prediction model using gru, lstm and bi-lstm machine learning algorithms. AI, 2(4):477–496, 2021.