Ethereum Price Prediction

Matthew Chen, Neha Narwal, Mila Schultz
{mchenja, nnarwal, milafaye}@stanford.edu

Overview

Ether is a young cryptocurrency, and the predictability of Ether prices is not well-known. We assess the performance of several models which predict the directionality of Ether price changes. Our results show that a SVM using price features performs well on this task.

Dataset & Feature Selection

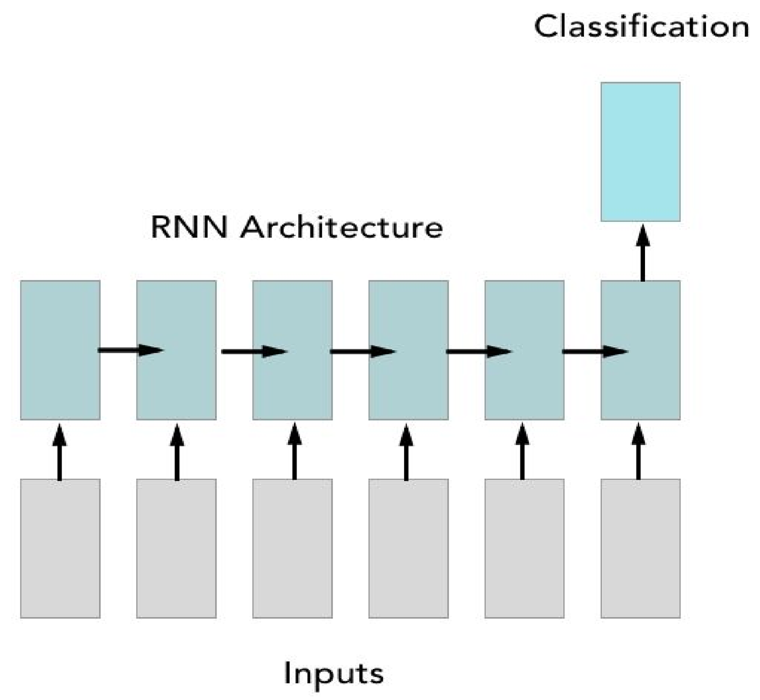
Original dataset includes 19,473 Ether prices (sampled hourly).
Prices before 2/26/17 are removed due to the drastic behavioral change after that date.
The price feature is 6 price points, labeled with **`sign(`**7^{th} price - 6^{th} price**`)`**.
The dataset is split into train/test/dev datasets by following a 80%/10%/10% split.

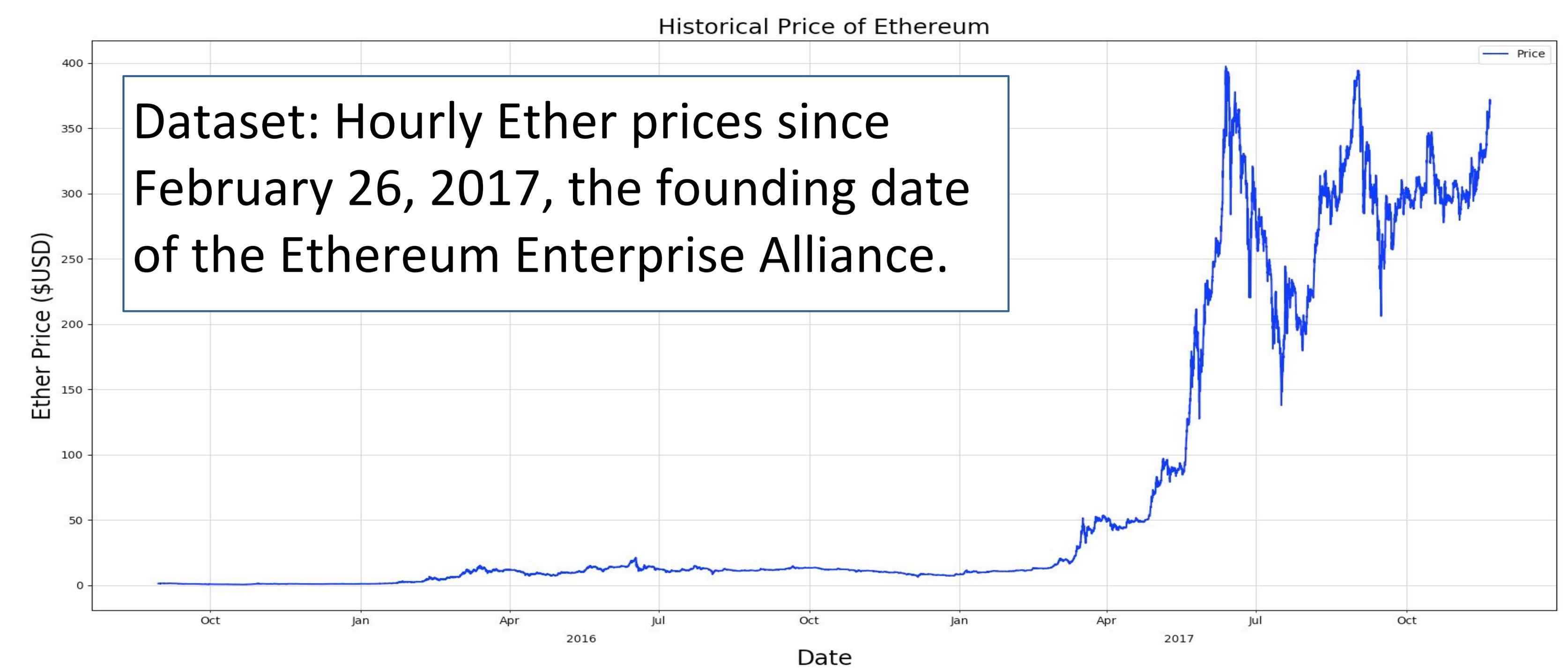
Summary Statistics

Sample Size	6383
Minimum	14.47
1st Quartile	85.23
Median	255.93
3rd Quartile	302.09
Maximum	397.31
Mean	210.42
Standard Dev.	115.44

RNN with LSTM

- The model takes batches of short series of price data as input, and uses the last prediction frame as output.
- The model is trained using softmax cross-entropy loss
- We experimented with batch size, learning rates, number of epochs, and activation functions
- If the signal is encoded in the most recent data points, LSTM may not have a large advantage over other methods
- We are still refining the model's architecture and performance



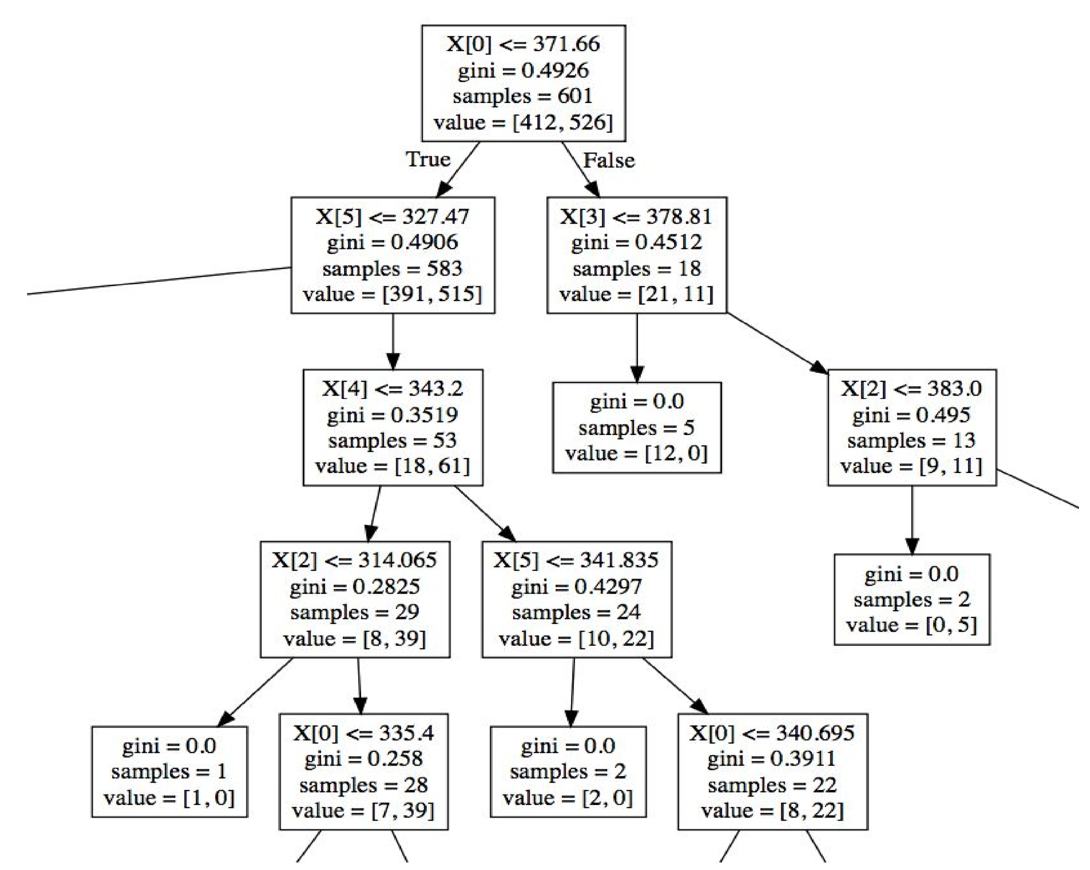


Results

- The best-performing SVM model achieves a classification accuracy of 96.1%, using a Radial Basis Function (RBF) kernel initialized with default parameters. While this performance is good, we expect performance to increase after hyperparameter tuning, which may involve tuning the kernel parameters and/or scaling the mean and variance of the training dataset.
- The Random Forest-based model also performs well, with a comparable accuracy of 94.2%. Random Forests perform well with data that has not been normalized and do not require involved hyperparameter tuning. The Random Forest can also learn different behavior for high prices vs. low prices, which could help its performance.
- The RNN with LSTM will benefit from additional tuning. We expect to modify the architecture and input additional features before the publication deadline.
- Logistic regression underperforms the SVM classifier, which suggests that the data is not linearly separable (relative to the price feature). Alternate feature selection may improve performance.
- The Naive Bayes classifier performs comparably to the baseline, which classifies entirely based on the prior possibilities. The most logical explanation is that the training data is not normally distributed; alternatively the feature points may not be conditionally independent.

Model Performance

Algorithm	Train Accuracy	Test Accuracy	F1 Score
Support Vector Machine	91.7%	96.1%	0.96
Random Forest	96.7%	94.2%	0.94
RNN with LSTM	55.7% (preliminary)	58.2% (preliminary)	-
Logistic Regression	58.7%	56.7%	0.53
Naive Bayes	55.8%	55.8%	0.40
Momentum Baseline	55.5%	52.9%	0.56
Naive Baseline	55.8%	55.8%	0.72



Part of a decision tree from Random Forest Classifier

Future Work

- We will refine the RNN model in an effort to improve the performance
- We may be able to train on other cryptocurrency price data to make our model more robust
- Model expressivity can be increased by adding more market information, such as transaction cost, market capitalization, daily opening and closing prices, and other features.
- We may be able to compare accuracy of these classifiers for another dataset that contains daily data

Conclusion

- The SVM-based model achieves a higher-than-expected accuracy with relatively little tuning, suggesting that the prediction task for Ether is less complicated than for comparable securities such as Bitcoin or common stock. This may change in the future as additional agents seek to invest more money in non-Bitcoin cryptocurrencies.
- The RNN-based model does not overfit on the training dataset, suggesting that it is limited by its configuration rather than by the model's expressive power. More work is needed.
- Logistic Regression and Naive bayes underperform. These algorithms are poor choices for this classification task, possibly due to their inherent assumptions that limit their applicability.

Acknowledgements

We would like to thank our professors and TAs who helped with our project.
The data we used was obtained from etherchain.org.