Stock Price Prediction with LSTM Network

Siyang Jing[1], Jiacheng Tian[1], Jiyu Xu[1], and Yuhui Huang[1]

[1]Department of Computer Science, University of North Carolina at Chapel Hill, NC 27599, USA

Summary: The time series of stock prices are non-stationary and nonlinear, making the prediction of future price trends much challenging. To learn long-term dependencies of stock prices, we first perform unsupervised learning to extract and construct useful features, then build a deep Long Short-Term Memory (LSTM) network to generate the prediction. The experiments on real market dataset demonstrate that the proposed model outperforms other four baseline models in the mean square error.

Motivation

- Application of deep learning has increased dramatically, as a result of exponentially enhancing computing power.
- Availability of high frequency trading data has increased.
- Interest in developing models using deep learning with high frequency trading data has grown as performance of current models become unsatisfactory.
- Previous studies have mostly focused on stock price prediction at a low frequency.

Introduction

- In the financial industry, stock price prediction has constantly been a popular field of research.
- stock markets have been proved to be predictable in some scenarios according to many widely accepted studies.
- It's interesting to focus solely on past trading patterns while other features are available.

Data

- We are currently using 30 days of minute-level price data of 50 stocks, in total of 11700 time steps.
- Our aim is to obtain data of 10 years for model training.
- Each stock, at each time step, has 5 features, open/high/low/close price and volume. In total, each time step contains 250 features.
- We aim to predict the close prices of the 50 stocks at next time step.
- We use the first 29 days as training, and the last day as test.

Method

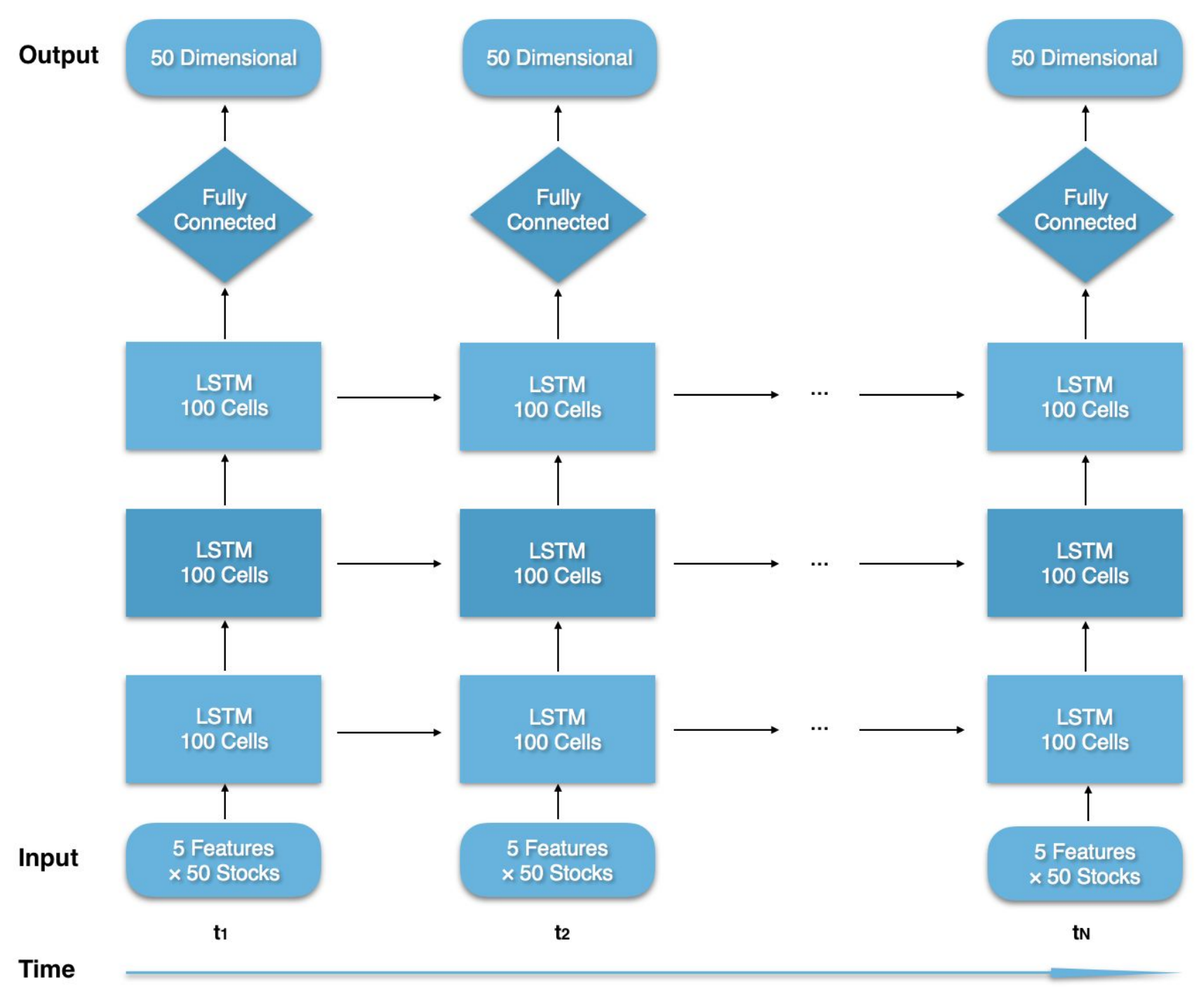
LSTM

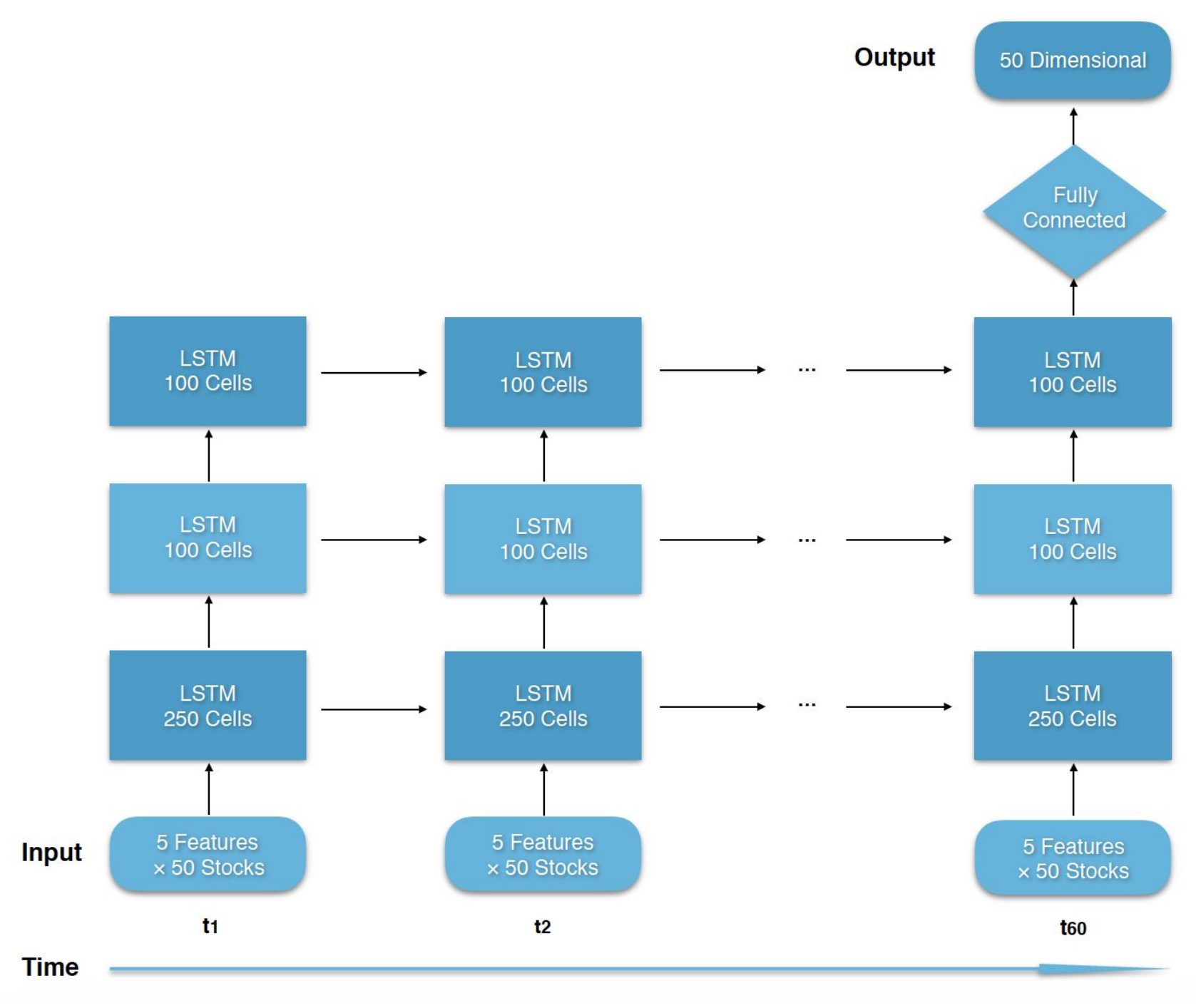
$$\begin{pmatrix} \mathrm{i} \\ \mathrm{f} \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$
$$h_t = o \odot \tanh(c_t)$$

- A Long short-term memory (LSTM) unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. Therefore, LSTM networks are well-suited for time series data.

- We propose two approaches to formulate the problem.
- Many to one approach. (right figure) The output is explicitly dependent on the previous 60 time steps (i.e. 1 hr). To make each prediction, the model has to run 60 time steps.
- Many to many approach. (left figure). The prediction is made at each time step, and the model can deal with sequence of arbitrary length.
- Experiments show that the many to many approach is roughly the same as many to one approach, but it's also far less computationally expensive and more flexible. Therefore, we proceed with the many to many approach. After a grid search, we found that a network of 3 LSTM layers, each with 100 units achieves best performance.
- We also thought of using unsupervised feature extraction before the network, but it does not give any improvement.





Experimental Results

- To evaluate our model performance on the test set, we employ the Walk Forward Validation strategy, where we do not retrain the model on each time step, but make the last tested data point available to the model.
- We define three error metrics:

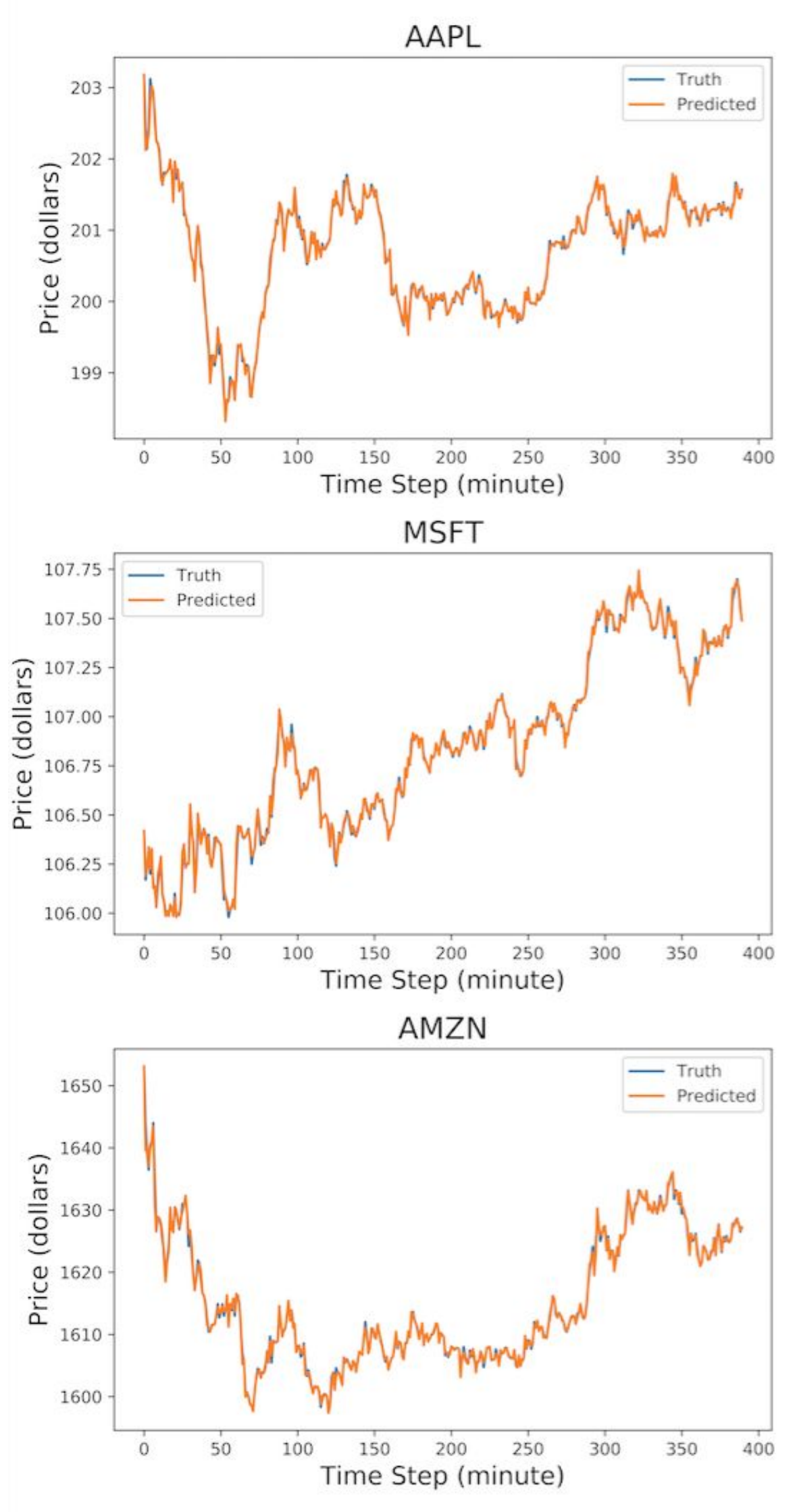
Mean square error (MSE): $mean((Y' - Y)^2)$ The square is element wise

Mean relative error (MRE): $mean(abs\left(\frac{Y'-Y}{Y}\right))$ The division is element wise

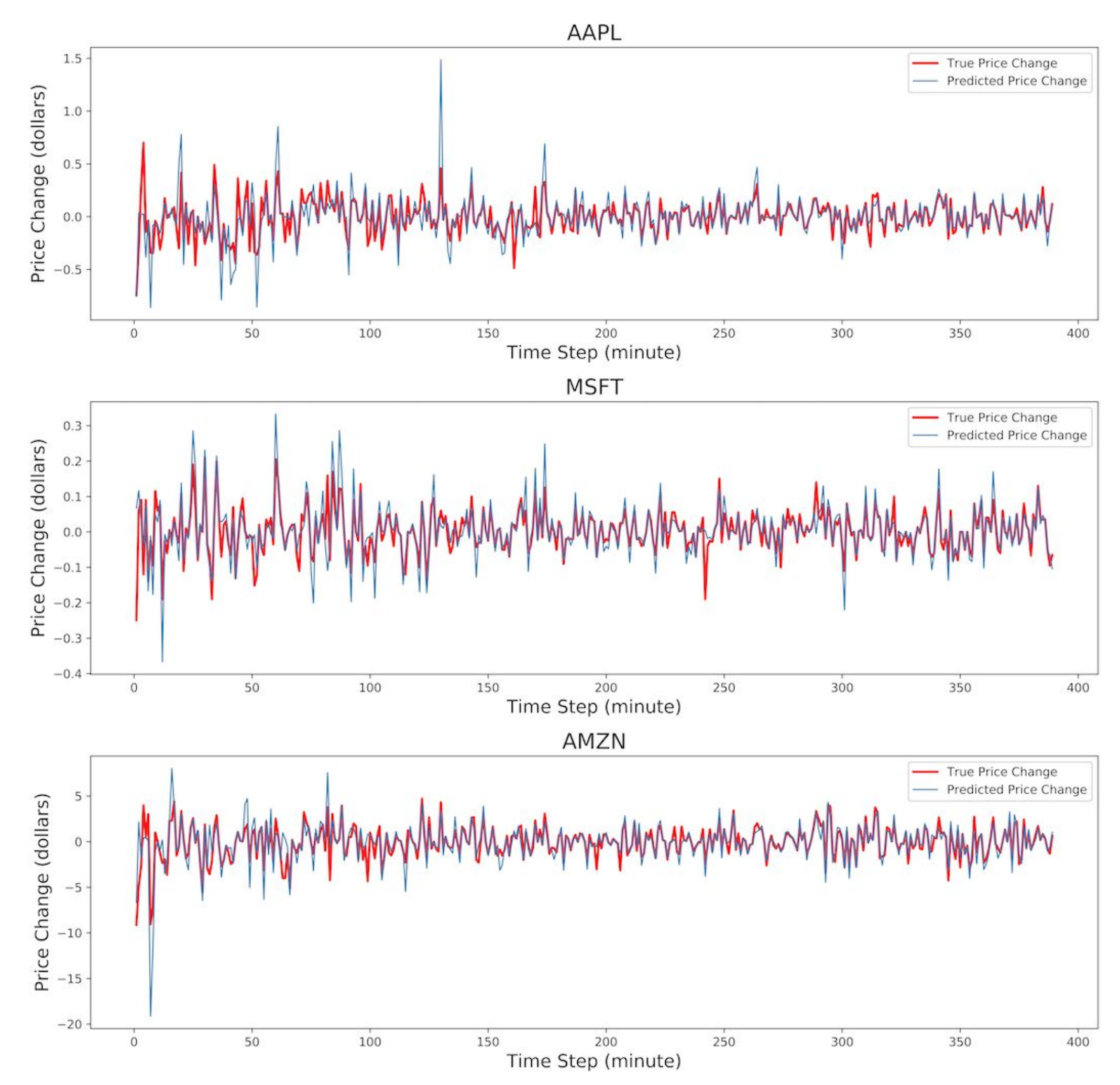
Trend Error (TE): $\mathrm{mean}(I[Y' \times Y < 0])$ I is indicator variable, × is element wise

- In the figures below, we plot the predicted and true price and returns for three stocks, apple, microsoft, and amazon.

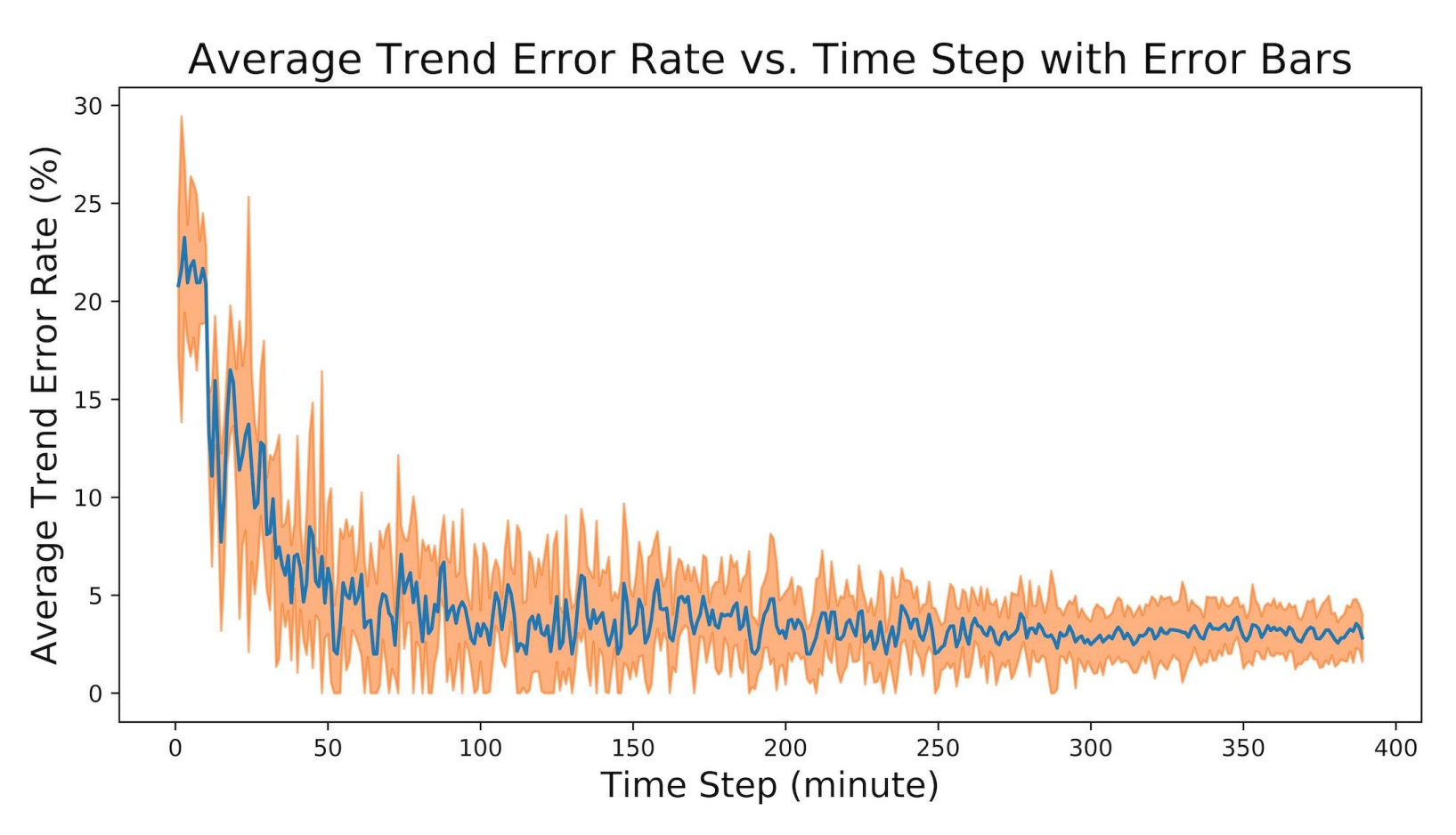
Predicted vs Actual Price over Time



Predicted vs Actual Return over Time



- The figure to the right shows how the trend error varies with time.
- The error significantly decreases after 50 time steps, and stablizes around 3-4% after 250 time steps.



- We compare our method with two popular statistical methods ARIMA and GARCH, the method from [1], and a simple feed forward network of 4 layers, each with 100 neurons. The table below shows the results.

METHOD	MSE	MRE (10^{-4})	TE (%)
PROPOSED	0.51	3.32	12.48
ARIMA	7.24	47.13	33.54
GARCH	10.56	68.74	35.12
CHONG 2017	3.89	25.32	31.37
FNN	2.17	14.13	22.96

Conclusions

- The results show that we beat the other methods on all 3 metrics.
- We achieve a trend error rate of 12.48%, which means we can correctly predict whether the stock price is going up or down at next minute most of the times.
- The performance is consistent for all stocks after a short period of time.

References:

[1] Chong, Eunsuk, Han, Chulwoo, and Park, Frank C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. 2017

[2] Zhong, Xiao and Enke, David. Forecasting daily stock market return using dimensionality reduction. 2017

[3] Chourmouziadis, Konstandinos and Chatzoglou, Prodromos D. An intelligent short term stock trading fuzzy system for assisting investors in portfolio management. 2016