FINANCIAL MARKET PRICING ESTIMATION USING DEEP LEARNING TECHNIQUES WITH REGRESSION BASED SMOOTHING

Haydn Anderson, Chris Nguyen
Washington State University
haydn.anderson@wsu.edu, chris.m.nguyen@wsu.edu

1 INTRODUCTION

With the rise of modern financial technology, more and more people have access to financial instruments than before. Instead of the field being dominated by hedge funds and quantitative researchers, retail investors have claimed a market share larger than ever before. This has created a rush of retail investors attempting to gain an edge over other investors to fully take advantage of new market opportunities.

Many new investors have no knowledge or access to advanced financial computation software or tools, leading to a knowledge gap that we will be attempting to take advantage of with our trend prediction model for financial markets. A problem with these emerging markets is the high level of volatility, which in turns adds more noise to the data. This is a problem for many models as over-fitting the noise gives inaccurate predictions, to solve this we will be calculating the slope of a sliding window within a batch of candles to smooth out the noise.

2 LITERATURE REVIEW

In the paper “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis” by Mojtaba Nabipour, Pooyan Nayyeri, Hamed Jabulani, Shahab S. and Amir Mosavi uses continuous and binary data in their approach to predicting stock market trends. In short continuous data is using input data that were computed through formulas. Binary data is converting the continuous data value of indicators’ nature and properties then using the following methods for the prediction models Decision Tree, Random Forest, Adaboost, XGBoost, SVC, Naïve Bayes, KNN, Logistic Regression, and ANN with 2 deep learning algorithms. The entire article is based on the use of continuous and binary data in determining trends. While ours is more centralized on the use of long short-term memory neural networks the learning rate

In “Predicting the direction of stock market prices using random forest” Luckyson Khaidem, Snehanshu Saha, and Sudeepa Roy Dey approach the idea of predicting the stock market by using the random forest as a learning rate for prediction. Extracting features to get the technical indicators that was used as parameters to help in forecasting stock market directions. The difference between our approach and this one is that this one is centralized on using random forest while our project we use are more focus on using LSTM as the basis of our model.

In the paper “Predicting the Trends of Price for Ethereum Using Deep Learning Techniques”, Deepak Kumar and Dr S. K. Rath are comparing the performance between two different deep learning models, the multi-layer perceptron and the long short-term memory network. They also similarly tested the performance of their networks on different time frames of data, such as daily, hourly and minute intervals. With our approach we are using different time frames as features in our model to increase the amount of visibility the network has to different data intervals. We also are using custom-built features to help our network learn better. Their work complements our choice of using a long short-term memory network as the basis of our model.

3 DATASET

Our data set is 800,000 1 minute candles (1m) of 3 different types of cryptocurrencies against the US dollar, specifically XMRUSDT, BTCUSDT, ETHUSDT. The preprocessing performed on the data involved normalization and calculating candles of higher time frames. To transform the market data into a higher time frame involves combining numerous one minute candles to the desired size. Each candle contains a high, low, open, close, volume and timestamp value. We also created a custom regression feature that calculates the slope of a window of candles. Our regression feature has a smoothing function that filters out the noise of the market. A higher window size filters out extraneous noise from the data due to how the slope is calculated.

4 BASELINE

We will be calculating the slope using linear regression of a sliding window of 12 one hour candles within a batch of 36 candles. Since our sliding window has a size of 12 candles, we use the last 12 candles to calculate the slope to prevent the slope taking future values into account. Our baseline will be the slope of 24 one hour candles compared to the slopes of the next 5 one hour candles outside of the 24 one hour candles. The candles were never used in the regression calculations using the sliding window, and are considered future values. We will be comparing the baseline slope to the 5 slopes of the one hour candles outside of the 24 candle batch using mean squared error and root mean squared error.

5 MAIN APPROACH

Our model will be featuring a long short-term memory network (LSTM). We will be providing inputs in the form of normalized candlestick data and our slope data calculated with linear regression in the form of features. We normalized our inputs using a normalization technique called Min-Max scaling, this converts all the input values into a range between (0 - 1).

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Since we used a set of candlesticks to calculate the slope for each candle, it added a smoothing function to our slope data that filters out the noise of small price movements. For example, let $C = (c_1, ..., c_s)$ where C represents the set of candlesticks with a size of s and each candlestick c_s contains a high, low, open, and close value. The set of candlesticks C will be referred to as a batch, and a window will be referred to as a set of candlesticks that is a subset of X with a size of l that is less than or equal to the batch size s.

To isolate a window of candlesticks with size l from the set of candlesticks C, let $w_i = (c_i, ..., c_{i+l})$, where w_i represents a sub set of candles from the set C from the index i to the index $i + l$, and i is the window index. The amount of candlestick windows to isolate can be calculated by subtracting the batch size s from the window size l.

For example, in our baseline we used a window size $w = 12$ and a batch size $s = 36$; we continued generating candlestick windows until $i = s - l$ and each window is calculated as followed.

$$\begin{gathered} w_1 = (c_1, ..., c_{1+12}) \\ w_2 = (c_2, ..., c_{2+12}) \\ \vdots \\ w_i = (c_i, ..., c_{i+l}) \end{gathered}$$

Let $W = (w_1, ..., w_{s-l})$, where W is the set of all windows in the batch C. For each element w_i, we used linear regression to calculate the slope of the window.

Instead of traditionally using the candlestick close values as inputs into our model, we will also be using the calculated slope values as inputs in our LSTM neural network. We will be performing multivariate time series forecasting on our data in an attempt to predict the slopes of the next 5 candles outside the initial batch C.

6 Evaluation Metric

We will be using mean squared error (MSE) and root mean squared error (RMSE).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Actual_i - Predicted_i)^2}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Actual_i - Predicted_i)^2$$

7 Results Analysis

Our baseline resulted in a mean squared error (MSE) of 44.89 and a root mean squared error (RMSE) of 6.70. Our LSTM neural network resulted in a MSE of 0.41 and a RMSE of 0.639. We tested our model against the noisy candlestick data as well, which resulted in a significantly higher MSE ranging from 66.55 to 248.99, and an RMSE ranging from 8.16 to 15.78. Predicting the slopes of our candlestick data proved to be effective, as MSE and RMSE were much lower. The sliding window method we used to calculate the slope has a built-in smoothing function to filter noise out. Using a window of 12 candlesticks allowed the slope to be unaffected by small changes in the data, making the LSTM neural network perform better on smoothed data.

References

Nabipour, Mojtaba, et al. ”Predicting stock market trends using machine learning and deep learning algorithms via continuous and binary data; a comparative analysis.” IEEE Access 8 (2020): 150199-150212.

Khaidem, Luckyson, Snehanshu Saha, and Sudeepa Roy Dey. ”Predicting the direction of stock market prices using random forest.” arXiv preprint arXiv:1605.00003 (2016).

Kumar, Deepak, and S. K. Rath. ”Predicting the trends of price for ethereum using deep learning techniques.” Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, Singapore, 2020. 103-114.