University of Essex

Department of Computer Science and Electronics Engineering

Report: Part 1

Subject: CF969 Big-Data for Computational Finance

Title: Ascertaining price formation in cryptocurrency markets with Deep Learning

Registration number: 2201829

Supervisor: Dr. Panagiotis Kanellopoulos

Date of submission : 21/04/2023

Word count: 1449

Introduction

The main objective of this document is to summarize and evaluate the article *"Ascertaining price formation in cryptocurrency markets with Deep Learning"* (Tables and figures in this document are referred from this paper) [1]. The following sections are divided into five i) A brief introduction about the contents of the paper ii) Methods used for the implementation of the model iii) Analysis of the suitability of the approach iv) Result and Analysis of experiment v) Strong and weak point of the paper.

Aim of the paper

The aim of this experiment is to predict the direction of the mid-price movement of upcoming ticks in the cryptocurrency market using high-frequency data. The characteristics of the cryptocurrency market have been analyzed and proposed a deep learning method has to find useful patterns from the limit order book. The authors chose 8 cryptocurrencies to live tick-level data that has been monitored and used both machine learning and statistical techniques for predicting live predictions. The experiment gave a promising result in the end with an accuracy of 78% on the prediction of the live exchange rate of US dollars vs Bitcoins

Methodology

A. **Dataset**:

The dataset used for the experiment is the live data recorded via WebSocket through the GDAX exchange WebSocket API. The data contained in the subset includes level-2 order book updates, orders submitted to the exchange and the ticker data. A total of 40,951,846 level-2 data, 61,909,286 order flow data records and 128,593 ticker data were available for the study[1].

B. **Architecture of proposed trading system**

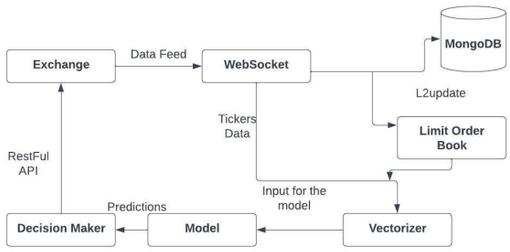

Figure 1 . Overview of Trading system

The design is more focused to predict the mid-price movement, especially cryptocurrencies. This model will be a part of the trading system as in Figure - 1. The following are the required components :

i) *WebSocket* - used to subscribe to the receive and exchange live data including the limit order book update, tickers and order flow.

ii) *Tickers* - It contains the best bid, the price and best ask, thus will show the change of real-time price.

iii) *MongoDB* - Prices and volume information on the limit order book are updated in real time using data from exchanges like GDAX. The data was stored locally in MongoDB, an adaptable non-relational database, and capable of storing in JSON format as documents.

iv) *Limit order book* - The local copy of the limit order book from the database can be reconstructed using level-2 limit order updates and can be used for calculating the order imbalance and can provide quantified features of the limit order book.

v) *Vectorizer* - it will finalize the input to the model by combining the information, data parsing and extracting features from the local limit order book and ticker data. It will reshape the features and pass them to the LSTM model.

vi) *LSTM model* - The architecture of the model features a neural network with two LSTM layers. The first layer has fully connected neurons, while the second is a softmax output layer managing upward or downward probabilities. These two layers capture non-linear data features, and the fully connected neuron layer acts as the decision layer with input from the last LSTM layer.
The neural network is designed to be simple as possible, as every millisecond is crucial in tick data. Reducing layers will significantly cut computational complexity and data processing time.

vii) *RMSprop optimizer* - A stochastic gradient descent optimizer, which is used to train the neural network. It will update the model parameters based on the output from the loss function and will evaluate the model performance of the dataset. The learning rate will be divided by the exponentially weighted average during training.

C. **Multi-label prediction**

Authors have replaced binary target prediction with a four-target prediction with four softmax layers and transformed the original two-class classifier into the four-class classifier. Considering the fees used by Coinbase Pro, authors also used ±0.2% of transaction fees as the threshold to differentiate the large and small changes. Below are the four classes we used for multi-label classification.

i) Significant increase - $(+0.2\%, +\infty)$

ii) Significant decrease - $(-0.2\%, \infty)$

iii) Insignificant increase - $(0, +0.2\%]$

iv) Insignificant decrease - $[-0.2\%, 0)$

D. **Walkthrough Training**:

Two walkthrough training methods used in the experiment are as follows.

i) *Walkthrough with stable retain frequency* - In this method model will be retrained in a fixed interval of time. Depending on the accuracy of the data and trading strategy the interval will vary. This will help the model acquire new features and increase efficiency and performance.

ii) *Walkthough with dynamic retrain frequency* - In this method the model will be retrained when a new peak is reached, using Maximum Drawdown, the biggest observed loss from a peak to a trough in the portfolio. It will help to reduce the chance of overfitting and waste of computational resources. The hyperparameter used for this method is Modified Maximum Drawdown.

$$\text{Modified MDD} = \frac{\text{Max Accuracy Value - Min Accuracy Value}}{\text{Min accuracy Value}}.$$

Suitability of the approach

The methodology they have adopted for this experiment is suitable as they have adopted different strategies to increase the accuracy and reduce the chance of failure. Some of the interesting factors they have considered as follows:

i) They have used live data for a long time span and historical data for modelling the trading system.

ii) The data selected for modelling has no bias as historical trading contains all the available transaction data and most of the available currency pairs in the cryptocurrency market.

iii) The experiments are not affected by policy impact, bear or bull market and other factors.

iv) They have wisely selected the data by including bull-market condition, high transaction volume condition, low-transaction volume condition, bear market condition etc.

v) Due to the use of live data from Coinbase Pro, bad connections may result in missing values. By cross-referencing data with third-party service providers, this issue has been fixed.

Result and Analysis

The study aimed to implement a model capable of predicting the mid-price movement of upcoming ticks using multi-label prediction. The following are important findings from the experiment:

i) Model has been trained with data size and the result shows that accuracy is not improving with increasing the size of the data. After removing the currency pair with fewer data and accuracy improvement, the model has been again trained and accuracy has improved. After training individual currency pairs, authors experimented with a universal model with all currency pairs and the result shows a significant improvement in accuracy and came to the conclusion that the universal model will give the best accuracy and performance.

Table 1 . Result of data size effect

	Model accuracy on testing set			
Sample size	AVG	Selected	Universal	Universal Selected
10%	0.60990	0.62419	0.64899	0.66417
20%	0.63578	0.67134	0.66962	0.68286
50%	0.67263	0.70823	0.69180	0.70057
70%	0.67252	0.73500	0.71111	0.73126
85%	0.66429	0.73902	0.71533	0.74140

ii)Autoencoder helped to lower noise in the data and address the universal predictive model's declining performance on live data. It reduced overfitting, increased speed and improved better capturing of sequential data than PCA by feeding the intermediate output of the autoencoder to LSTM. The following table shows the effect of the autoencoder on performance improvement.

iii) The model's performance can be stabilised and accuracy on real data can be increased using walkthrough training. "Walkthrough with stable retain frequency" outperforms "Walkthrough with dynamic retrain frequency" but considering the number of retraining and computation resources later performed better.

Table 2 . Performance improvement after integrating auto encoder

Classification report of autoencoder as denoiser				
	Precision	Accuracy	F1-score	Support
up	0.72	0.86	0.78	5,265
down	0.81	0.65	0.72	4,927
avg / total	0.77	0.76	0.75	10,192
Classification report of autoencoder as reducer				
	Precision	Accuracy	F1-score	Support
up	0.79	0.78	0.79	5,265
down	0.77	0.78	0.77	4,927
avg / total	0.78	0.78	0.78	10,192

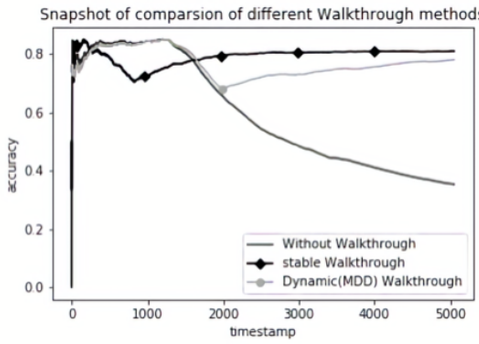

Figure 2 . Comparison of the result of different walkthrough methods

Strength and Weakness

Some of the strong points I have identified in this project have been described below:

i) They have designed the system by taking the excellent measurement to reduce the chance of failure either in the way the data is collected and also in improving the performance of the model. So the chance of failure will be less. Also expecting to get a stable result.

ii) They have tried different scenarios for training the model with different data sizes and identified the best scenarios for training the model(eg: universal deep learning model)

The paper has some potential weak points that may impact the robustness and generalizability of the findings.

i) The study has a limited scope of experimentation, as it only used one deep learning model and focused on a single dataset from a specific time period. This may reduce the generalizability of the results to other models, datasets, and time periods.

ii) The statistical analysis is limited, as the paper mentions that the experiment was repeated 20 times for a stronger statistical guarantee but does not provide detailed statistical measures.

iii) The study also does not explicitly discuss the limitations of the research or provide practical guidance for implementing the identified optimal walkthrough methods in real-world scenarios.

References

[1] Fan Fang et al. "Ascertaining price formation in cryptocurrency markets with DeepLearning". In: *arXiv preprint arXiv:2003.00803* (2020).