Pump and Dump Detection Documentation

Sadrach Pierre

January 7, 2019

1 Data Collection

Raw trade data for 60 crypto coins were gathered from Binance exchange from 2018-6-01 00:00:00 to 2018-9-01 00:00:00.

1.1 Feature Engineering

Volume, price returns, price volatility and number of trades were resampled from raw trade data every 30 minutes. The 30 minute window is a tunable parameter which can be found/changed in solidus-ml-analytics/Configuration/ windowassumptions.json. The maximum value in price returns is selected along with its timestamp. The corresponding volume, number of trades and volatility are selected for the given timestamp of maximum return.

In order to label pump and dumps accurately three 30 min steps forward after the maximum value in return are taken. The 30 min step size can also be changed. It is found in solidus-ml-analytics/Configuration/ windowassumptions.json as well. The values are stored in a list and the minimum is selected. The dump is defined as follows: if $R_{min} < 0$ we define R_{dump} as,

$$R_{dump} = R_{max} + |R_{min}| \tag{1}$$

if $R_{min} > 0$ we define R_{dump} as,

$$R_{dump} = R_{max} - R_{min} \tag{2}$$

1.2 Labeling Data

The training data for the KNN algorithm is labelled using a tunable parameter called TARGET-THRESHOLD found in solidus-ml-analytics/Configuration/modeltuning.json. If $N_{trades} * R_{max} >=$ TARGET-THRESHOLD it is given a label "1" for pump and dump and "0" otherwise. We assign TARGET-THRESHOLD a value of 60,000. This is an assumption based on initial exploratory data analysis, where on average normal market behavior has a value several orders of magnitude smaller than 60,000.

1.3 K-Nearest Neighbors Algorithm + Rule-Based Dump Detection

The input for the K-NN model is a list of features [returns, volatility, volume, trades]. Manipulated point have features with large values and normal market points have relatively small values. K-NN takes an input of [returns, volatility, volume, trades] and calculates the distance of the feature input from the labelled training data. Votes are cast and the class with the most votes has the label that we assigned to the input. Finally, if K-NN returns a label of "1" for pump and dump and $R_{dump} > 0$ we label the feature input as a legitimate pump and dump.