STOCK PRICE MANIPULATION DETECTION USING GENERATIVE ADVERSARIAL NETWORKS

Presented by Team 2

INTRODUCTION

Market has bad players who gain profit by deceiving other players through market manipulations. The field faces **3 challenges**:

1. Distinguishing manipulations from normal trading behaviours.
2. The small amount of damage from a few easy-to-expose manipulations.
3. The data is partially observable, due to buyers/sellers privacy.

Proposed solution: train deep neural network, since manipulation cases are irregular trading behaviours and it can learn without labels.

RELATED WORK

The related research focuses on trade-based manipulations that studied theoretical and empirical work:

1. **Large investors** have **large probability** to take part in stock manipulation.
2. An investor who has i**nsider information** of a firm also has **high probability** to be a manipulator.
3. **Manipulator** may gain his profit by making **takeover bid** for stock price manipulation (sell the stock at the expected price).
4. **Liquidity, returns, and volatility** affected the stock price **during manipulation periods**, where stocks with low volatility and value had high chances to be manipulated.

<u>*Examples of used machine learning techniques and other methods*</u>:

1. Data mining techniques vs. logistic regression and discriminant analysis → **data mining performed better**.
2. K-nearest neighbour vs. adaptive dynamic model → **both performed well**.

STOCK MANIPULATION PATTERNS

- **Illegal** strategies to influence stock prices for personal gain.
- **Examples**: Information-based, Action-based, and Trade-based.
- **'Pump-And-Dump'**: Inflating stock prices through misleading positive statements, then selling off once the price is high.
- **Variables involved in stock price manipulations**: Cancelled volumes and matched volumes.
- **Impact**: Distorts market integrity, misleading investors, and affecting market stability.

ANOMALY DETECTION USING GANS

GAN Framework

Utilizes two neural networks, a Generator and a Discriminator, in a competitive setup.

Discriminator

Learns to differentiate between real and generated data.

LSTM Integration

Incorporates Long Short-Term Memory networks to effectively process time-series data.

Generator

Creates synthetic data mimicking real stock trading patterns.

Anomaly Identification

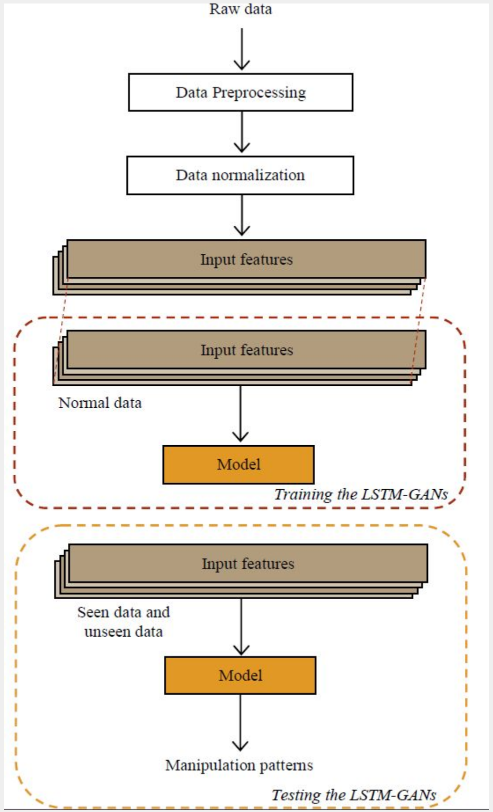
Anomalies in stock data are treated as 'generated' data, revealing manipulation patterns.

Advantages

Enhanced detection capabilities, adapting to evolving manipulation techniques.

MODELS

- **Architecture**: Combines GANs with LSTM networks.
- **LSTM Usage**: Processes time-series data, essential for analysing stock market trends.
- **Generator Model**: Creates data sequences simulating potential manipulation patterns.
- **Discriminator Model**: Distinguishes between actual market data and generated sequences.
- **Training Process**: Iterative adjustment of both models to improve detection accuracy.
- **Objective**: To effectively identify subtle and complex manipulation tactics in stock market data.



EXPERIMENTS DATA

The real manipulation cases are limit and rare to obtain. Unable to get enough real data, the authors switched to injecting synthetic manipulation patterns to the real dataset, which isn't available today.

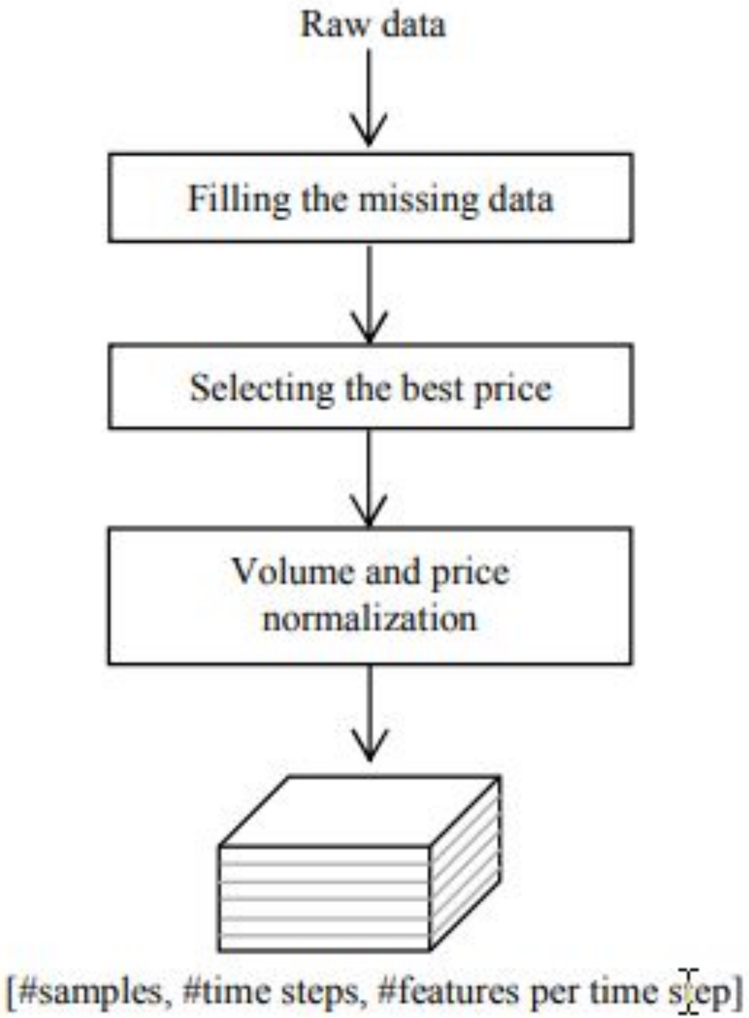
Paper's data: 22 full trading days of the biggest companies, with no reported market manipulations.
Our data: Bitcoin Dataset – midpoint, spread, amount sold, cancelled amounts, best bid price and best ask price.

The authors focus on the format of data called limit order book (LOB). The LOB tables were set in five depths with single second sampling period.

The stock volumes were normalised by a common logarithm (base 10) and then with Z-score. The normalised Z-score as shown in the following equation will help adjust the value to have a normal distribution.

TABLE I. INPUT FEATURES

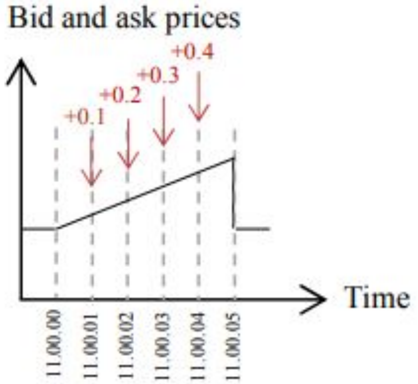
Column	Variable explanation
1	best bid volume
2	second best bid volume
3	third best bid volume
4	fourth best bid volume
5	fifth best bid volume
6	best ask volume
7	second best ask volume
8	third best ask volume
9	fourth best ask volume
10	fifth best ask volume
11	best bid price
12	best ask price
13	matched volume
14	Cancelled volume (bid side)
15	Cancelled volume (ask side)



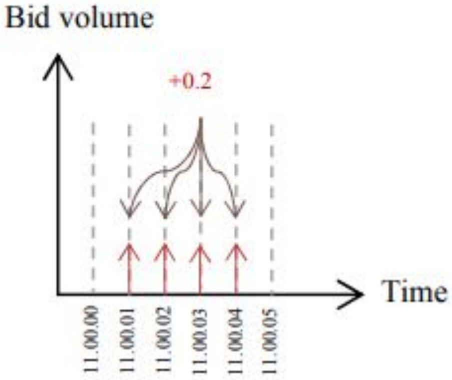
EXPERIMENTS

1. The LSTM-GANs were trained on sequential data.
2. A random noise was then fed to a dense layer and the LSTM layers of the generator.
3. This outputted a sequence which tried to mimic the original time-series.
4. The data was then reshaped and entered into the discriminator for the classification task.
5. The output from the discriminator was ‘1’ for real and ‘0’ for generated data.

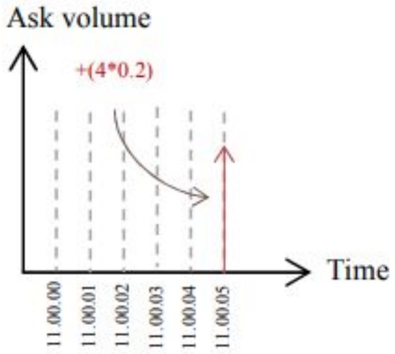
The authors proposed a manipulation scheme that lasted for 5 seconds.
Pumping period: first four seconds,
Dumping period: one second.
Constant rate for bid and ask: 0.1% per second during the pumping period.
In the dumping period, the bid orders were cancelled, and the ask orders were quickly executed.



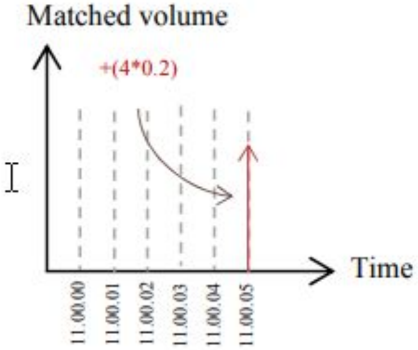
(a) Bid and ask prices in pumping period.



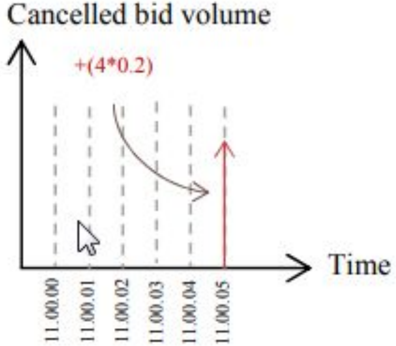
(b) Bid volume in pumping period.



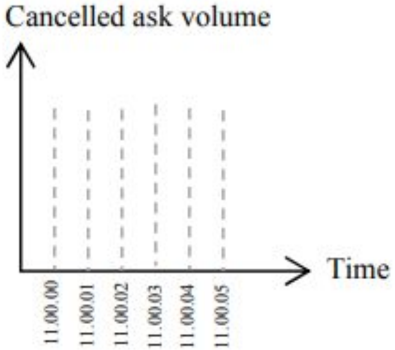
(c) Ask volume in dumping period.



(d) Matched volume in dumping period.



(e) Cancelled bid volume dumping period.



(f) Cancelled ask volume in dumping period.

EXPERIMENTS RESULTS

Inputs	Model accuracy
1. seen normal cases	80.05%
2. unseen normal cases	73.09%
3. seen normal cases + manipulative pattern	61.46%
4. unseen normal cases + manipulative pattern	68.10%

CONCLUSION

The authors' experiment showed that the LSTM-GANs were effective for detecting stock price manipulations. The model performed well with both seen and unseen data. This is the first research to apply deep generative adversarial networks on the problem of detecting stock price manipulations.

Contributions of the paper:

- The paper proposed an **unsupervised methodology** employing a combination of LSTM and GANs for stock price manipulation detection.
- The proposed LSTM-GANs were **effective** for detecting stock price manipulations.

However:

- The effectiveness of the networks **needs to be evaluated** on different datasets with real manipulative cases.