Detection of pump and dump in a micro-cap stock using an LSTM network.

Acknowledgement: I wish to extend my gratitude to my advisor, Mr. Yiqiao Yin, who helped me throughout the process of writing this research paper. I would also like to thank Mr. Krishnan Neelakantan and Mr. Anish Teli for helping me understand stock market related aspects of my paper.

Abstract: This paper implemented an LSTM network to detect price manipulation in a micro-cap stock called Ram Minerals and Chemicals Ltd. The data collected from the Bombay Stock Exchange was manually labeled and utilized to train a simple Recurrent Neural Network (RNN) model to ensure the data was well labelled, and then an advanced LSTM model to produce an average validation loss of 0.207 in a time series cross-validation experiment.

Introduction

Pump and dump is a form of stock manipulation where the price of stocks with a small or micro capitalization is spiked by the use of fraudulent messages which attract misinformed investors, and when the price is high the fraudsters sell their shares, thus gaining a profit.[1] This leads to a massive loss for investors and a drop in the price of the

[1] "Pump and Dump Schemes | Investor.gov." *Investor.gov*, 2022, www.investor.gov/introduction-investing/investing-basics/glossary/pump-and-dump-schemes.

stock as well.[2] Thus, this paper aims to identify such cases of stock manipulation in a micro-cap stock called Ram Minerals and Chemicals Ltd. through the use of three explanatory features: close price, rolling volatility and trading volume.

These three features were chosen to explain a pattern that is visible in most pump and dump stocks. This pattern starts at the pumping section, where the scammers increase the close price by trading small amounts of shares in an illiquid stock. The liquidity of a stock is “how rapidly shares of a stock can be bought or sold without substantially impacting the stock price.[3] Thus, for an illiquid stock even buying low volumes increases the close price at a fast pace. Therefore, during the pump section, there is usually a low trading volume, high rolling volatility and high close price. But in the dumping section, the demand for the stock is now high, thus the fraudsters can sell their stock at high volume and increase the supply, which in turn decreases the close price but increases the rolling volatility, as the change in price is very fast. Thus, the model aims to analyze these two sections of a pump and dump scheme.

---

[2] “Pump and Dump Schemes | Investor.gov.” *Investor.gov*, 2022, www.investor.gov/introduction-investing/investing-basics/glossary/pump-and-dump-schemes.

[3] “Liquidity (or Marketability) | Investor.gov.” *Investor.gov*, 2022, www.investor.gov/introduction-investing/investing-basics/glossary/liquidity-or-marketability#:~:text=A%20stock’s%20liquidity%20generally%20refers,shares%20when%20you%20want%20to.

Experiment Design and Modeling

Close price and number of shares are included in the BSE (Bombay Stock Exchange) data provided, but rolling volatility is a derived feature. It is the standard deviation of the close price in a rolling window of 60 dates.

To separate the training and validation set time-series cross-validation is used to include the importance of time in the data. In time series cross-validation k examples from the data are taken as the training set, and the $k + 1th$ example is used as the validation set, and for the next iteration the $k + 1th$ example joins the training set and the validation set becomes the $(k + 2)$ example. Through this, the model learns the time dependency, unlike in a normal validation split or k-fold cross-validation.

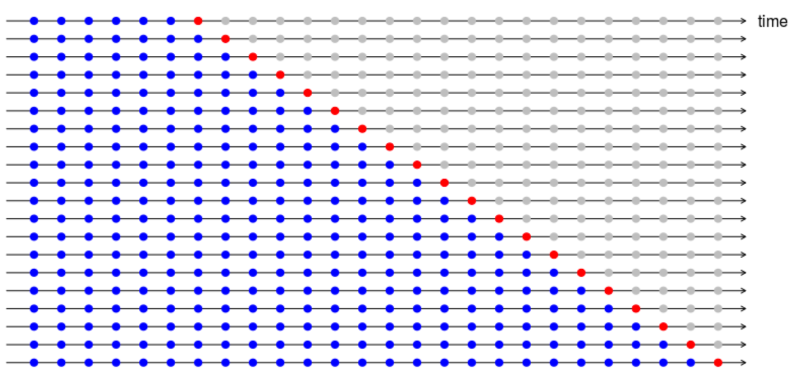


Figure 1: Time series cross-validation[4]

[4] Hyndman, Rob J, and George Athanasopoulos. "5.10 Time Series Cross-Validation | Forecasting: Principles and Practice (3rd Ed)." *Otexts.com*, 2015, otexts.com/fpp3/tscv.html.

A simple neural network, called an Artificial Neural Network, is a machine learning model, which has an input layer, a set of hidden layers, and an output layer. The input layer is a set of neurons each connected to a neuron in a hidden layer. To include non-linearity in the model, each output of the hidden layer is connected to an activation function, which is usually either a tanh function, a reLu function or a sigmoid function.

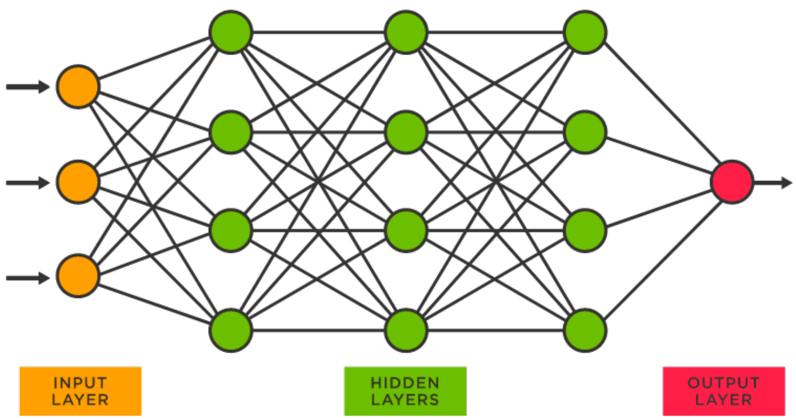


Figure 2: Architecture of an Artificial Neural Network[5]

Activation functions increase the complexity of the model, thus making the predictions more accurate but also leading to the model fitting exactly against the training dataset, which is called overfitting.[6] To reduced overfitting the model must be penalized, therefore ridge regularization must be used. Ridge regularization is an addition or a

[5] "What Is a Neural Network?" *TIBCO Software*, 2021, www.tibco.com/reference-center/what-is-a-neural-network.

[6] IBM Cloud Education. "What Is Overfitting?" *Ibm.com*, 3 Mar. 2021, www.ibm.com/cloud/learn/overfitting.

penalty for the loss function which helps reduce overfitting to the data, thus allowing more accurate predictions. In the case of ridge or L2 regularization, the penalty is the sum of all the weights squared multiplied by lambda, which is a variable that can be manually changed to manipulate the impact of regularization.[7]

A neural network can be explained by a set of equations[8]:

$$\hat{Y}_i = \sigma(w_1 i_1 + w_2 i_2 + w_3 i_3 + b)$$

To minimize the weights for each layer in a simple neural network a gradient descent algorithm called backpropagation is used, where after propagating forward once, the derivatives of the functions are used to minimize the loss function.

$$L(Y_i, \hat{Y}_i) = -\frac{1}{n}\sum_{i=1}^{n}(Y_i \cdot log\hat{Y}_i + (1 - Y_i) \cdot log(1 - \hat{Y}_i)) + \lambda \sum_{i=1}^{n} w^2$$

$$\nabla = \frac{\delta}{\delta w} L(Y_i, \hat{Y}_i)$$

$$w_s = w_{s-1} - \eta \nabla$$

[7] Bhattacharyya, Saptashwa. "Ridge and Lasso Regression: L1 and L2 Regularization." *Medium*, Towards Data Science, 26 Sept. 2018, towardsdatascience.com/ridge-and-lasso-regression-a-complete-guide-with-python-scikit-learn-e20e34bcbf0b.

[8] "Neural Networks: Structure | Machine Learning Crash Course | Google Developers." *Google Developers*, 2020, developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/anatomy.

η is the learning rate of the algorithm and is used to calculate the step in the Y direction after one epoch of the gradient descent algorithm. In the first step, after calculating the loss function for the predicted Y value, $\hat{Y}_i$ and the actual Y value, Y_i, the derivative of the loss function is calculated, and as $L(Y_i, \hat{Y}_i)$ has $\hat{Y}_i$, the chain rule is used to calculate the derivative in respect to w_i. Then, using the gradient vector the direction for the steepest descent is identified and the new w_i is calculated. This is done for all the weights involved as well as the bias in the linear function.

However, the baseline model for the classification of RMCL as pump and dump is a simple Recurrent Neural Network (or RNN), which advances the simple neural network by allowing the model to classify time series data, data which have a time interval between them.

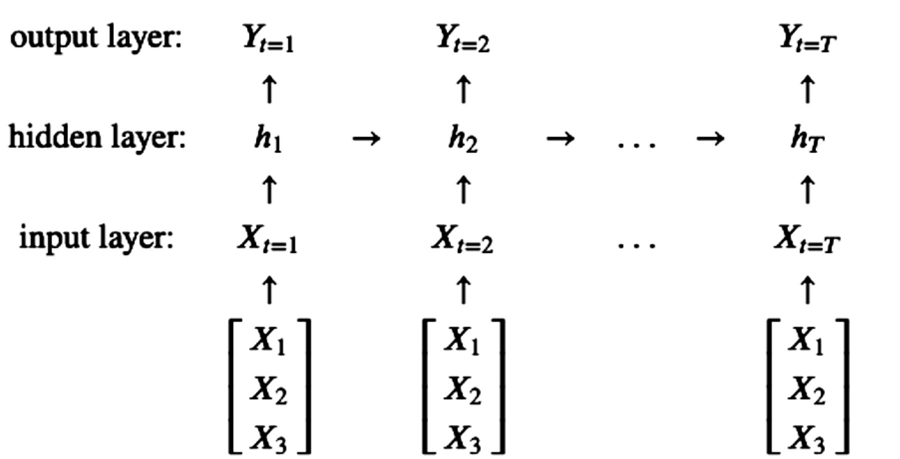


Figure 3: Architecture of the simple RNN

For the classification of RMCL, the input layer is an input vector of 3 dimensions, including the rolling volatility, close price, and the No. of shares. This input vector goes through a simple neural network with hidden layer h_1 which connects to an output for the first input only,$Y_{t=1}$. Since time series data must be taken into account by the model, an input of $X_{t=2}$ is added to another hidden layer, which is connected to the previous timestamp through the previous hidden layer. This ensures the time intervals of the data are taken into account when classifying the stock. Yet to train the model, instead of a normal back propagation algorithm a time series backpropagation algorithm must be used.

$$L(Y_{i,t=1}, \hat{Y}_{i,t=1}) \ = \ -\frac{1}{n}\sum_{i=1}^{n}(Y_{i,t=1} \cdot log\hat{Y}_{i,t=1} + (1 - Y_{i,t=1}) \cdot log(1 - \hat{Y}_{i,t=1})$$

$$C(Y, \hat{Y}) \ = \ \sum_{t=1}^{T} L(Y_t, \hat{Y}_t)$$

$$\nabla C\left(Y, \hat{Y}\right) = \frac{\delta}{\delta t}\frac{\delta}{\delta w} \ C(Y, \hat{Y})$$

$$w_s \ = \ w_{s-1} - \eta \nabla C\left(Y, \hat{Y}\right)$$

Therefore, in a time series back propagation algorithm by including the loss function for different time stamps, and summing them up in the cost function, and then including the derivative in respect to time for the gradient of the cost function, the algorithm includes the time-series sequence of the data.

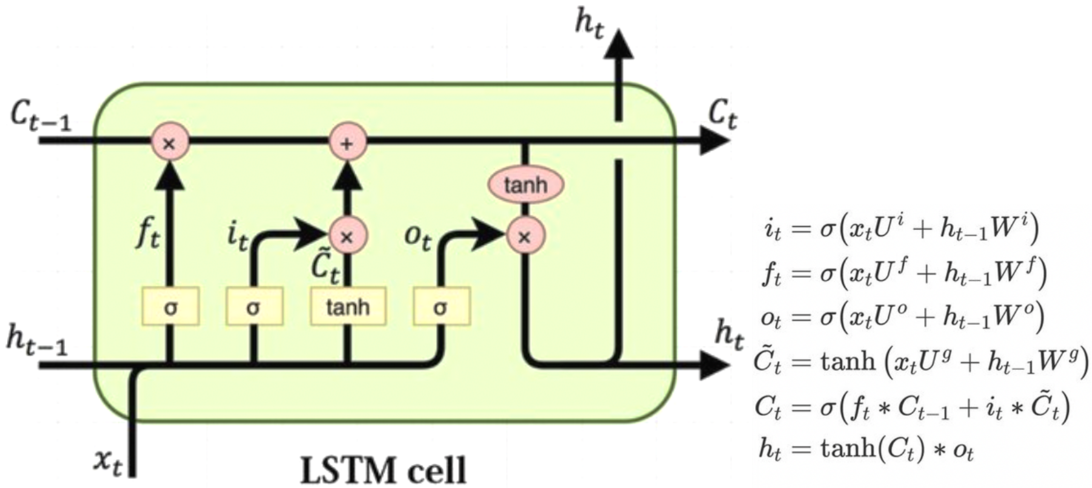


Figure 4: Architecture of the LSTM cell [9]

In the Long Short-Term Memory (or LSTM) model the simple RNN cell is replaced by the LSTM cell where the cell now includes multiple neural networks or ‘gates’, helping the model learn long-term dependencies.[10] This is not possible in an RNN due to the structure of the cell, which only has a connection to the previous hidden layer, thus through back propagation, the gradients of the initial time stamps will be much smaller than the final time stamps, this is known as the vanishing gradient problem.[11] The LSTM

[9] Varsamopoulos, Savvas, Koen Bertels, and Carmen G. Almudever. "Designing neural network based decoders for surface codes." *arXiv preprint arXiv:1811.12456* (2018).

[10] “Understanding LSTM Networks -- Colah’s Blog.” *Github.io*, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

[11] “Understanding LSTM Networks -- Colah’s Blog.” *Github.io*, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

solves this issue through the use of three gates, the forget gate, the input gate and the main gate.[12] The forget gate allows the model to delete unwanted data through a sigmoid activation function with an input of x_t , and the previous hidden layer, h_{t-1} . While the addition of new data is handled by the input gate, which has the same structure as the forget gates but with weights U^i and W^i, and the layer that contains all the possible candidates for the new value, which is the tanh activation function of the current input, x_t and the previous hidden layer, h_{t-1} . Thus, through multiplying these two layers the model includes all the updates to the cell. Then, the final gate or the main gate is the addition of the previous two gates, thus ensuring the new C value includes the deletions and the additions to the data, while the new hidden layer is the tanh of the new C value multiplied with the output layer, which is the sigmoid function for the linear function of x_t and h_{t-1} .

The LSTM model uses the same gradient descent algorithm as the RNN model.

Application

[12] "Understanding LSTM Networks -- Colah's Blog." *Github.io*, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

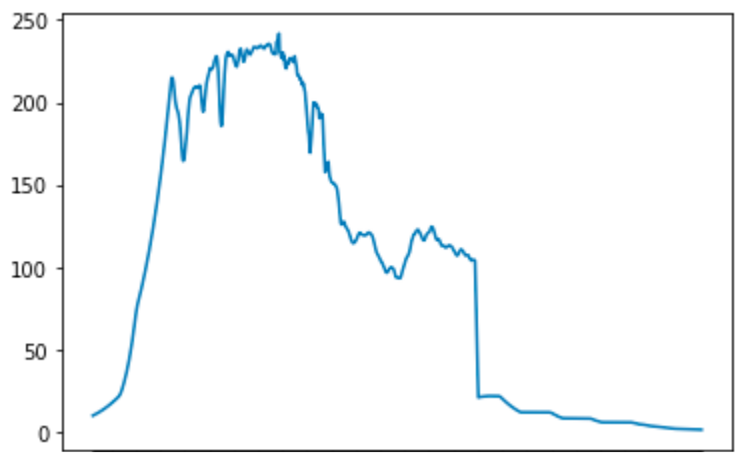


Figure 5: Graph of the close price of the stock

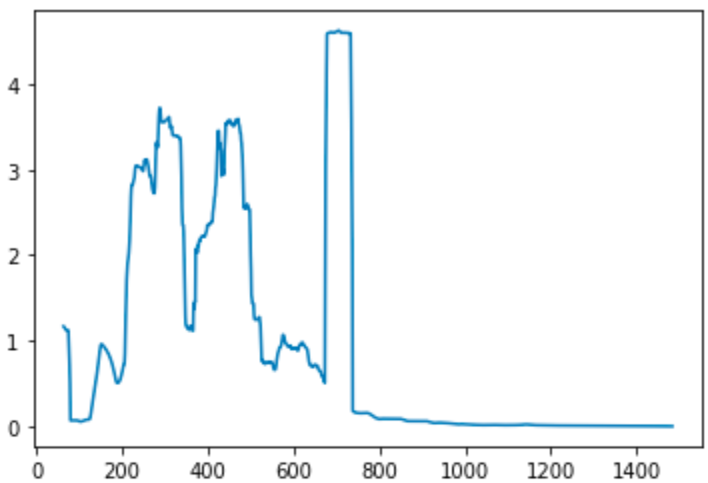
Ram Minerals and Chemicals Ltd. is a micro-cap company, with a market capitalization of Rs. 10.08 crores.[13] The company has been barred from the Indian stock market by the Securities and Exchange Board of India (SEBI) for raising the price of the stock from Rs. 2.20 to Rs. 219.55 between December 2013 to December 2014, which is a clear indication of stock manipulation.[14] This is also visible in figure 5 as there is a sudden rise in the price of the stock over a short period of time. Thus, this gives two major indicators of a pump and dump scheme, the number of shares traded or the trading

---

[13] "Ram Minerals and Chemicals Share Price : Live NSE/BSE Stock Price Today." *Business Standard*, 2020, www.business-standard.com/company/ram-minerals-41288.html.

[14] PTI. "Sebi Bars 12 Entities from Securities Market for Manipulating Share Price." *The Economic Times*, Economic Times, 4 June 2019, economictimes.indiatimes.com/markets/stocks/news/sebi-bars-12-entities-from-securities-market-for-manipulating-share-price/articleshow/69655267.cms.

volume and the volatility of the close price of the stock. To include these features in the model the close price, the number of shares sold, and the rolling volatility over a window of 60 days was taken into consideration. This allowed the data cleaning process to be easier as well, as the labels for the data were created through the use of the rolling volatility and the number of shares sold. The data cleaning process began with a big window of 5 days, where the rolling average of 5 days was calculated for the data.



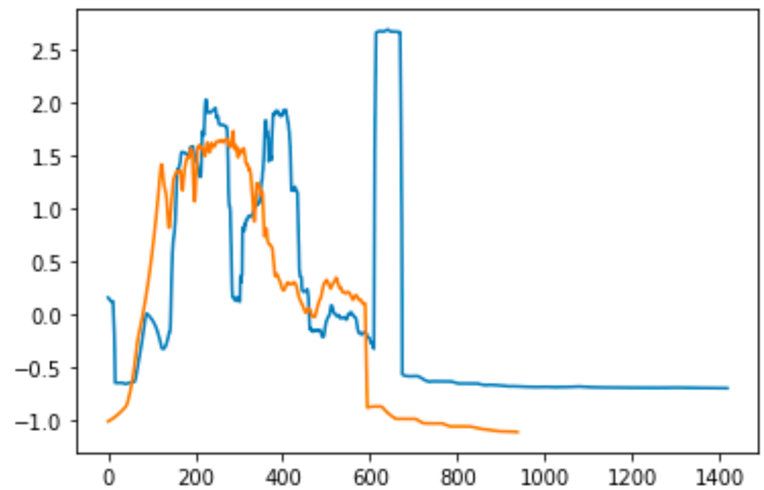


Figure 6 and 7: Plot of the rolling volatility; Plot of the close price and rolling volatility

After establishing the big window, the labels of the data had to be created for the model to train over. Figure 6 shows the first step to creating the labels of the data. By making a rolling window over 60 days and getting the rolling standard deviation for the data, the rolling volatility is made. Then, the values that are greater than the mean of the rolling volatility plus one standard deviation are labeled as the initial signal for stock manipulation. This allows the sudden rises and drops in price to be classified as pump and dump as the moments where the volatility is high are when the price is rising fast. This can be seen when the two figures are graphed together, as in figure 7 it is clear that when the price is rising at the beginning, the rolling volatility is high, and when the price suddenly drops towards the end, the rolling volatility increases again.

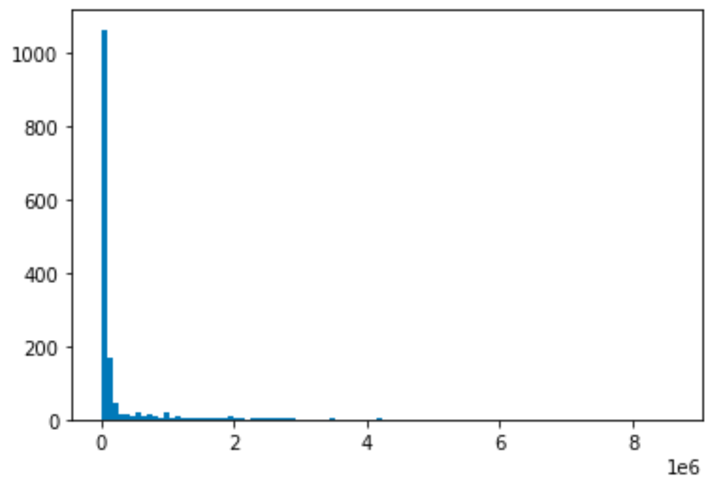


Figure 8: Histogram of the trading volume of the stock

The trading volume must also be taken into consideration when creating the labels for the data. Figure 8 is a histogram of the trading volume of the data, and in the diagram clear outliers are visible, both higher and lower than the mean. To establish the relatively high and relatively low volume, the mean of the no. of shares of the false signal was compared to the array with the no. of shares of the true signal to help generate the pump and dump true and false values. Any examples with a higher volume than the mean of the false signal or a lower volume than the mean of the examples with low volatility minus 0.15 times the standard deviation of the examples with low volatility were labeled True, while all the other examples were labeled False.

Models	Training error	Test error
1	0.56789976	0.4866789
2	0.47528237	0.16715588
3	0.56689459	0.0289225
4	0.54336625	0.20090245
5	0.54649913	0.12842482
6	0.55311239	0.39722967
7	0.53014624	0.18315436
8	0.5321492	0.20324875
9	0.45534143	0.07106766

Table 1: RMCL model results

For this experiment for Ram Minerals and Chemicals Ltd. a cross-validation is run. This cross-validation has an initial training set of 350 examples and uses 50 as an increment up till 750 examples are used as the training set. Thus, there are 9 validation sets in total. The model used to train the data in each of these validation sets is an LSTM-based RNN with a different number of neurons for each model; model 1 has 10 neurons and this is incremented by 50 every iteration until a final value of 460 neurons. The model is fit using an Adam optimizer, a binary cross-entropy loss function and run with 50 epochs and a batch size of 100. The output of the model used as the results of the experiment is the minimum training and validation loss for each cross-validation set.

An LSTM model was utilized for this experiment because stock returns are time-series data, thus there are long-term time dependency patterns in the data structure.[15] Therefore, the only way to capture these patterns and accurately classify pump and dump is through an LSTM model.

Conclusion

This paper developed a Long Short-Term Memory (LSTM) Network to predict pump and dump in a micro-cap stock called Ram Minerals and Chemicals Ltd. The results in table 1 show that through cross-validation the test error decreases as the number of neurons increases, and the experiment goes on, as model 1 has 10 neurons and a testing error of 0.567, but model 10 has a testing error of 0.071 and 460 neurons. Thus, the model clearly learns the pattern visible in the stock through an LSTM network. Even though this experiment has shown some success there are still many aspects of a pump and dump scheme that have not been taken into account such as the liquidity of the stock,

[15] "Understanding LSTM Networks -- Colah's Blog." *Github.io*, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

and a more advanced model such as a Generative Adversarial Network (GAN) can be used to increase the accuracy of the experiment.[16] Therefore, other researchers should continue to explore this field as pump and dump schemes have also started to creep into cryptocurrency and are a major threat to a person's financial safety.

References

Varsamopoulos, Savvas, Koen Bertels, and Carmen G. Almudever. "Designing neural network based decoders for surface codes." *arXiv preprint arXiv:1811.12456* (2018).

Chadalapaka, Viswanath, et al. "Crypto Pump and Dump via Deep Learning Techniques." *arXiv preprint arXiv:2205.04646* (2022).

Teema Leangarun, Poj Tangamchit, and Suttipong Thajchayapong. 2018. Stock price manipulation detection using generative adversarial networks. In 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pages 2104–2111.

Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. Lstm-based encoder-decoder for multi-sensor anomaly detection. arXiv preprint arXiv:1607.00148.

Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural

---

[16] Leangarun, Teema, et al. *Stock Price Manipulation Detection Using Generative Adversarial Networks*.

network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD ’19, page 2828–2837, New York, NY, USA. Association for Computing Machinery.

Wenjie Yang, Ruofan Wang, and Bofan Wang. 2020. Detection of anomaly stock price based on time series deep learning models. In 2020 Management Science Informatization and Economic Innovation Development Conference (MSIEID), pages 110–114.