Using a Model of Fraudulent Trader for Fraud Detection

Peter Fratrič[1,*], Sander Klous[1] and Tom van Engers[1,2]

[1]Informatics Institute, University of Amsterdam, the Netherlands
[2]Leibniz Institute, TNO/University of Amsterdam, the Netherlands

Abstract

The technological revolution brought by the internet, high performance computing, and artificial intelligence has fundamentally changed and continues to alter the landscape of finance. These innovations, if used with a malicious intent, can seriously destabilize the financial market. For this reason, countermeasures in the form of new detection methods are needed. In this study, we propose a novel detection framework that uses a model of fraudulent behavior to detect fraud from observed data. A similarity measure is defined to decide if the recorded actions of a monitored trader are matching actions of the fraudulent agent. We illustrate the framework on a simple form of manipulative trading in a simulation environment of a market consisting of two exchanges. This demonstrative case study is inspired by a price manipulation scheme that occurred on the Bitcoin market in 2017/18, where such simple forms of manipulation were observed. Simulation results outline vulnerabilities in markets, where uneven distribution of liquidity is present, as this can be exploited by pump-and-dump scheme.

Keywords

Computational finance, Fraud detection, Pump-and-dump scheme, Cryptocurrency

1. Introduction

Market fraud has been and continues to be a pressing issue of modern trading systems. Due to relatively low number of observed instances and often high complexity of the fraudulent behavior, the intrusion of the market is a challenging task to detect. Fraud can have wide consequences on every socio-economical system. For instance, the issue of market manipulation is especially present on cryptocurrency market [1] and other relatively immature markets. Although advances using statistical or modern machine learning solutions for the purpose of monitoring the market behavior have been achieved [2], the question of accurate and cost-effective market monitoring remains unsolved.

In this study, we aim to address the question of detection on a more basic level. In principle, every fraud is a manifestation of a behavioral scheme, and as such needs to be approached in this way by considering a specific behavioral model. Agent-based simulations of economic and social systems are gaining prominence in economy and finance. One might wonder if agents designed to violate norms in these models could potentially be used to aid fraud analysts. Moreover, to design suitable policies by accessing consequences of fraud on the market using

AMPM 2022: 2nd Workshop in Agent-based Modeling & Policy-Making, December 14th, 2022, Saarbrücken, Germany
*Corresponding author.
p.fratric@uva.nl (P. Fratrič); klous.sander@kpmg.nl (S. Klous); t.m.vanengers@uva.nl (T. van Engers)

the simulation environment. Assuming availability of a model of a fraudulent entity, we focus in this study on the research question of how can such a model be applied for fraud detection.

1.1. Related research

Financial fraud has been intensely studied through decades [3]. In addition to tax evasion, money laundering, or credit car fraud, the interest in study of market manipulation [4] has increased due to the rise of *pump-and-dump* schemes on the cryptocurrency market [5]. In the area of fraud detection, models based on deep neural networks appear to be successful [2], as they are able to model correlations in higher dimensions. Although the performance of these models on certain validation datasets was relatively high, the issue of explainability and the legal groundings of these models remains a significant drawback in wider applicability in the socio-economic domain.

Agent-based models appear to be a suitable tool for compliance modelling [6]. In the economic setting, agent-based models are more flexible compared to traditional mathematical models, because they can model various nuanced psychological aspects of individual traders [7, 8]. These become relevant once an agent with the intent to take advantage of these aspects enters the market. Application in the area of financial crime was limited to evaluation [9], or generation of synthetic data [10]. Detection algorithms have been implemented as part of the agent-based model [11], or in a distributed detection setting [12], but to our knowledge, no work has been done in exploring how the model of a fraudulent agent can be used directly for detection of fraudulent behavior.

1.2. Contribution

The goal of this study is to illustrate how a model of a fraudulent agent can be used to detect manipulative trading from an observed sequence of trading actions, and motivate further research on agents learning noncompliant behavior from a simulation environment. We discuss how this methodology can be integrated into current practice of market surveillance, assuming that a model of a noncompliant agent is already available.

The secondary contribution of this study is the implementation of a pump-and-dump market manipulation scheme in a simulated market environment, that was observed on the Bitcoin market in 2017/18 [13]. Recent agent-based analysis [14] suggested that the manipulation scheme takes advantage of uneven distribution of liquidity, and the presence of trend-following traders. We present new computational evidence that these two vulnerabilities of the Bitcoin market significantly contributed to the success of the price manipulation efforts.

2. Simulation-detection framework

Consider a socio-economic system where traders can exchange their assets, and there exists a rule describing what type of behavior is considered malicious. This rule is not enforced unless a detection method is present to identify noncompliant behavior. Assume a model of this system calibrated to observed data to reproduce standard behavior of the real system. In this system model, a parametric model of noncompliant agent is included, together with a similarity

measure that measures the similarity between the fraudulent agent and observed data of the monitored trader. Depending on the similarity, the system monitoring authority can decide to intervene.

2.1. Class of noncompliant behaviors

Let π denote the policy function of the fraudulent agent. The agent observes the current state of the system, denoted by vector $\mathbf{s}$. The state refers to publicly available variables, eg. public information on a stock exchange, or possibly transactions on a blockchain.

The model of the fraudulent agent will typically have a number of parameters θ, that define a subclass of fraud among the class of fraudulent behaviors, eg. a pump-and-dump scheme with parameters defining the frequency or amount of purchased assets, a time threshold after which the price is dumped etc. Note that these parameters are not coefficients of a statistical model, eg. a Bayesian network, or a neural network. This means the π_θ defines a class of noncompliant behaviors parametrized with θ, where for each choice θ the behavior remains noncompliant.[1]

2.2. Similarity score of fraudulent agent

Let us consider a window of observations J, indexing the set of all states $S = \{\mathbf{s}_j | j \in J\}$ and the set $data$ that denotes the set of actions a_j taken by the monitored trader for $j \in J$. The similarity measure between $data$ of the monitored trader and the fraudulent agent π_θ for given observed states of the system is defined as:

$$sim[data, \pi_\theta \mid S] = \frac{1}{|J|} \sum_{j=1}^{|J|} m[a_j, \pi_\theta(\mathbf{s}_j)] \tag{1}$$

where m is a *matching function* between each action of the monitored trader and fraudulent agent. This measure needs to be defined by (human) expert, and may depend on application domain. We provide an example of the matching function later. For now, let us assume that the function m is equal to 1 if the observed action of the monitored trader and the supposed action of the fraudulent trader are exactly the same, and is equal to zero if they are as different as possible. Note that the window expressed through J needs to be wide enough to include all relevant evidence. If the fraudulent behavior is not entirely contained in the window, the score still provides a valid indication of suspicious activities.

2.3. Inference

The fraudulent agent observes the observable state of the system $\mathbf{s}_t$ at time t. The policy function π_θ maps the observable state and the internal attributes of the agent into an action. Since π_θ defines a class of noncompliant behaviors, one would be interested to find such parameters θ that maximize the probability that actions of some monitored trader correspond to the actions of the fraudulent agent, ie. to localize the instance in the set $\{\pi_\theta | \theta \in \Theta\}$ most similar to the

[1] Note that not all choices of θ need to be profitable for the agent.

monitored trader, where Θ is a feasible region. In other words, we solve the optimization problem

$$\theta^* = \arg\max_{\theta \in \Theta} sim[data, \pi_\theta \mid S] \tag{2}$$

Compared to standard statistical models, where likelihood function is of central importance, what we gain by developing a causal model we loose during the inference process. Since agent-based models do not necessarily come with a mathematical functional expression, the inference needs to be performed without this assumption. The only assumption we make is the possibility to sample (execute) the model. So-called *likelihood-free* methods were developed to perform inference on simulation-based models [15]. These methods typically require a *similarity measure* between observed and generated data, and a prior distribution defined over the feasible region Θ from which parameter values θ are sampled.

2.4. Model-predictive intervention

Once a solution to equation (2) is found, the (human) compliant analyst can decide to intervene or not. This would be done by comparing the similarity measure (1) to a predefined intervention threshold δ. During the inference process, an implicit assumption is made that π is reasonably close to a *true* model of considered fraudulent behavior. This assumption is highly relevant for calibration of an intervention threshold for the market surveillance system, because a model of noncompliance that produces high similarity score for every observed behavior would have too high false positive rate. For this reason, every model needs to be validated on data generated in the simulation model of the socio-economic system, or on observational data of the system.

Performing monitoring of a trader can be done using the whole data history of the trader, or in some predefined window. In our framework, with a model of fraudulent agent available, it is possible to make rolling calculation of the score for the best estimated parameters. If the fraudulent behavior is in progress, it is very common that there is insufficient evidence for authorities to take action, ie. for an extended amount of time the behavior can be, although suspicious, but still, fully compliant. For a realistic model of the fraudulent agent and correctly defined similarity measure with calibrated intervention threshold value, it is desired that the similarity score will be high for suspicious behavior and will surpass the threshold value only after the fraudulent scheme is approaching its final steps. Ideally, right before the scheme concludes is where it is needed to intervene. If a model of compliant behavior that is similar to fraudulent behavior is available, then one can set δ to be higher than the similarity score produced by the compliant behavior, but is lower than the score of the fraudulent behavior. This can prevent unfair interventions on individuals that are compliant, but only slightly differ from fraudulent instances of a particular fraud scheme.

3. Modelling pump-and-dump scheme

The prototypical example used to illustrate the proposed framework is based on a real event of market manipulation that occurred on the Bitcoin market in 2017/18. This case was initially analyzed in [13], where clustering methods were used to identify relevant addresses, and statistical methods were used to provide evidence that flow of cryptocurrency through these

addresses was highly correlated with price increase. The market manipulator had access to virtually unlimited amount of Tether, a so-called stable coin supposedly backed by dollar, that was issued by *Tether limited* and was used to create artificial demand on the Bitcoin market. This case of market manipulation was in-depth investigated in [14], where an order book market model was developed with several simple trading agents, including a market manipulator agent. Since *Tether limited* was bound to release audit statements proving that every issued Tether is backed by one dollar, the manipulation scheme had to engage in massive selling at least once per month, although evidence suggested that this liquidation process occurred roughly every two months. This market behavior was likely even more impactful due to uneven distribution of liquidity on various exchanges, so that the price pumping on a less liquid exchanger could propagate through the whole system. Conversely, dumping the assets on a more liquid exchange would result in a lower market impact.

Taking the discussion above into consideration, we develop a model of an agent executing a pump-and-dump scheme in a simulation model of a market. In order to test if uneven distribution of liquidity plays a significant role, we consider a model of two exchanges, where one exchange is less liquid. Since the illustrated methodology is independent of the Bitcoin market and can be applied to any type of trading system, we will refrain from talking specifically about Bitcoins, and will talk about the assets in general.

3.1. Market model and response agents

We consider a simple order book market model based on [16]. An order book is a trading mechanism with a bid side and an ask side that lists buy and sell orders, respectively. Each order consists of limit price, asset amount to be bought or sold, and an expiration date. A trading day τ is discretized into T_{tic} time steps during which each agent can issue a buy or sell order. Let us denote by $p(t)$ the price of the traded asset and O_t the state of the order book at the time step t. In our model, two orders are matched in the order book if the limit price of the top buy order is higher or equal to the limit price of the top sell order. The new asset price on the exchange is then calculated as an average between the two limit prices. To extend the idea to a market consisting of two exchanges, the market price $p(t)$ of the asset at time t is calculated as an average weighted by daily traded volume of both exchanges $p(t) = \frac{v_1(t)p_1(t)+v_2(t)p_2(t)}{v_1(t)+v_2(t)}$, where $v_1(t), p_1(t)$ and $v_2(t), p_2(t)$ are daily volume-price pairs on exchange one and two, respectively.

Two types of so-called *response agents* are trading in the simulation environment. These agents are intended to model the response to the manipulation of the price. The simplest type of agent is a *Random agent*. This agent issues a sell or buy order with equal probability every time step on each exchange. At both exchanges $i = 1, 2$ the random agent calculates the limit price as $p(t-1) \cdot N(\mu_i, \sigma_i)$ for sell orders and $\frac{p(t-1)}{N(\mu_i,\sigma_i)}$ for buy orders, where $p(t-1)$ is the market price from previous time step. The asset amount is a random value drawn from an exponential distribution with rate parameter λ_e and the expiration time is drawn from a Poisson distribution with parameter λ_p. The values of all parameters are listed in Table 1. The second type of agent trading on both exchanges is *Chartist agent*. This agent is active with 0.5 probability. Chartist issues a buy order if the current price $p(t-1)$ is higher than 7-day market price average and sells otherwise. All other parameters are the same as for the random agent.

Parameter values of these agents are listed in Table 1. Note that due to different values of

Table 1
Parameters of the agents. The first five parameters belong to the response agents, λ_e belongs both to the fraudulent and the response agents, and the last three parameters belong to the fraudulent agent only. The response agents' parameters are similar as in the initial study [14], where they were calibrated on market data.

Parameter	μ_1	μ_2	σ_1	σ_2	λ_p	λ_e	p_{FA}	h	u_{max}
Value	1.01	1.05	0.01	0.05	1.0	5.0	0.20	55	0.1

pair μ_1, σ_1 compared to pair μ_2, σ_2, there is lower agreement about the price of the asset on the second exchange.

3.2. Fraudulent agent

The observable state of the system is defined by a vector $\mathbf{s}(t) = (O_1(t), O_2(t), p_1(t), p_2(t), p(t))$, where $O_1(t)$ and $O_2(t)$ are order books of exchanges one and two, respectively. The fraudulent agent observes the state $\mathbf{s}(t)$ and makes a decision according to policy π_θ. The function π_θ implements a simple trade-based pump-and-dump scheme that takes advantage of uneven distribution of liquidity on exchanges. The agent performs a sequence of aggressive buy orders such that the price impact due to low liquidity is maximized. At time h the price of the asset is dumped by a sequence of sell orders, but this time issued so that the price impact is minimized by targeting the more liquid ask order book. Two attributes of the agent are the capital balance $C(t)$ and the amount of assets $A(t)$ at time t. The agent aims to execute a pump-and-dump scheme such that the capital balance is positive and the amount of assets is zero.

In more detail, the agent decides to issue a buy order before day h with probability p_{FA}. The limit price, the amount of buy orders, and the expiration time is decided as in the case of random agent. To choose which exchange to target, the fraudulent agent will estimate the liquidity by calculating the immediate cost of buying or selling 10 units of its asset. If the agent is buying, then the less liquid exchange is targeted. Conversely, the agent is selling on more liquid exchange. The selling process is initiated after time threshold h is reached. At each time step a sell order is issued of $U \cdot A(t)$ asset amount, where U is a random variable uniformly distributed on the interval $[0, u_{max}]$.

In the simulation environment, parameters can be identified that make the manipulation scheme profitable in given market conditions. Chosen parameters for the fraudulent agent are listed in Table 1. An example of market price influenced by agent's actions is shown on Figure 1a.

3.3. Action matching function

Consider a trader being monitored in the real market producing a sequence of trading actions. The actions recorded into the set $data$ have a form: `(order type, exchange ID, asset amount, limit price, expiration time)`. By inspecting π_θ, it is easy to see that certain parameters of the agent do not have to be estimated using likelihood-free inference. For example, by having data of the monitored trader and known distributions used by the policy π_θ the

information about the distribution of limit prices or expiration times can be obtained using standard distribution estimation methods, thus we omit measuring similarity with respect to these parameters.[2]

Let $a_\pi = \pi_\theta(\mathbf{s}_t)$ be the action of the fraudulent agent given observable state $\mathbf{s}_t$ at time t, then we define:

- Order type match: $I_{ot}(a, a_\pi) = 1$ if the order types of the monitored trader and the fraudulent agent are equal; zero otherwise.
- Exchange match: $I_{ex}(a, a_\pi) = 1$ if the exchange choice of the monitored trader and the fraudulent agent are equal; zero otherwise.
- Amount distance: $g(x_a, x_{a_\pi}) = e^{-(x_a - x_{a_\pi})^2}$ for the asset amount components x_a and x_{a_π} of a and a_π, respectively.

Summing up the above quantities, the action matching measure is defined as $m(a, a_\pi) = w_{ot} I_{ot}(a, a_\pi) + w_{ex} I_{ex}(a, a_\pi) + w_g g(x_a, x_{a_\pi})$, where $w = (w_{ot}, w_{ex}, w_g)$ are weights associated with each summand. These weights can be used by the analyst if some patterns are suspected to be more significant. The sequence of orders and selected exchanges provide more information in our specific example, therefore we set the wight vector to $w = (0.4, 0.4, 0.2)$.

3.4. Inference using Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) [17] is a likelihood-free method capable to estimate parameters of a model if the likelihood function is unknown. The main idea of ABC is based on Bayes theorem $P(\theta|D) = \frac{P(D|\theta)}{P(D)} P(\theta)$ where the prior $P(\theta)$ is chosen by selecting a particular distribution G. Since the likelihood is not possible to evaluate, and we are only interested in the relative posterior plausibilities of θ, the normalizing constant $P(D)$ can be also ignored. All ABC methods approximate the likelihood by simulations that are compared to observed data. The main premise of the method can be therefore expressed as $P(\theta|D) \simeq \rho_\epsilon(D, \hat{D}) G(\theta)$ where $\hat{D}$ is a sample of considered parametric model. The model is executed for parameters drawn from the prior distribution G, and the simulation outcome is compared to observed data using a distance measure ρ for a given acceptance threshold ϵ.

In our case, the observed data are $D := data$. The simulated data are obtained by evaluation $\pi_\theta(\mathbf{s})$. The prior distribution G is defined over the feasible set Θ by the domain expert. Since the similarity measure is defined to be equal to the (maximum) value one when the observed and generated action are identical, we set $\rho := 1 - sim[data, \pi_\theta|S]$. As discussed in the previous subsection, some parameters of the fraudulent agent are excluded from the likelihood free inference, and therefore we let $\theta = (h, u_{max}, \lambda_e)$.

We use the simplest form of ABC algorithm, which is the *rejection sampler*. To infer the parameters of the fraudulent agent, the algorithm consists of four simple steps:

1. Sample θ from prior distribution $G(\theta)$.
2. Simulate a dataset $\hat{D}$ using $\pi_\theta(S)$.

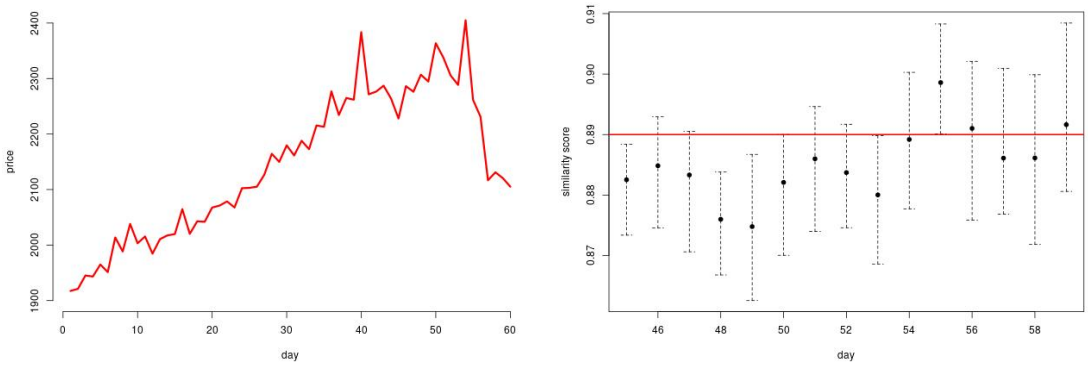
[2] We omit information about these parameters completely only for the sake of simplifying the example, since the focus is on the likelihood free inference. In practice, the information should be included in the scoring function, possibly with smaller impact depending on prior expectation of its significance.

3. If $sim(data, \hat{D}|S) > \epsilon$, accept θ, and reject otherwise.
4. Return to step 1 unless termination criterion is met.

The termination of the algorithm is ensured by defining maximum number of iterations. At the end of the sampling process, histograms of parameter values are generated.

4. Results

Applying inference to random, chartist, and fraudulent agent By solving (2) for each agent, the parameters produced similarity scores much higher for the fraudulent agent than for the random agent and the chartist agent. Therefore, the model can successfully differentiate between fraudulent and non-fraudulent agents, which provides baseline evidence that the model and the similarity measure are correctly defined[3]. In the case of the fraudulent agent, the ABC algorithm converged for threshold $\epsilon = 0.8$ only in parameter h, while the distributions of parameters u_{max} and λ_e stayed roughly uniform, which was confirmed by low p-values of Kolmogorov-Smirnov test for uniformity. The parameters that did not converge for predefined number of iterations suggest that the amount distribution parameter is not an important component of the similarity measure, as was initially suspected.



(a) Manipulated market price

(b) Similarity to pump-and-dump scheme

Figure 1: The market price on the left resembles a typical pump-and-dump pattern. On the right, the similarity estimates of the synthetic monitored trader to a fraudulent agent during the last days of the pump-and-dump scheme.

Deciding the intervention threshold In general, assuming that we know what type of compliant behavior is the most similar to the fraudulent agent, we can determine the threshold values by computational experiment. Any pump-and-dump scheme has a minimum number of

[3]Clearly, in practice this testing would be done on real data of both compliant and fraudulent traders, but can be also tested on various models of compliant agents to investigate which decision mechanisms tend to trigger false positives.

days the price pumping process takes. Since the similarity measure (1) is an average over the number of recorded actions, we can set without loss of generality $h = 55$ days[4].

Consider a scenario where each day parameters of the monitored trader are estimated. Let us set ϵ to be the 95th percentile of the scoring distribution. On Figure 1b we can see the average value of the similarity score calculated by sampling from the posterior. One can see that the values of the score tend to be higher after the price dumping process starts on day 55. Obviously, the intervention threshold δ should be higher than the similarity score of an agent that is unreasonably buying assets on less liquid exchange, which an example of a compliant agent similar to the fraudulent agent. Once the monitored trader starts the price dumping process, the threshold δ should be low enough to identify the similarity as significant enough to trigger a response from the authorities. This is why we set $\delta = 0.89$, as can be observed on Figure 1b. Until day 54 the trader appears to be compliant, but on the 55th day the selling process stars, which means pump-and-dump scheme ought to be detected. The red line is the value of intervention threshold δ. To prevent false alarms, the detection threshold is set slightly higher than the score of the compliant behavior. Credible intervals give us information about the certainty of observed fraudulent behavior. For instance, the monitored trader on day 54 has slightly less than 50% chance to be fraudulent, but on 55th day the certainty is much higher.

5. Conclusion

Design of agents capable to conduct a specific type of behavior in a real or simulated environment is a central research question of artificial intelligence. In our study, we further motivate this research by proposing a framework in which the model of a fraudulent agent can be used for fraud detection. The proposed framework is illustrated on a market model where an agent executing a simple pump-and-dump scheme is present. We use the agent to generate synthetic data, and then to test the framework using the same model for detection. We also demonstrate how the intervention threshold used by the market monitoring authorities can be decided by considering similar but compliant sequences of actions. Moreover, our market model exemplifies the vulnerability of markets where uneven distribution of liquidity is present on exchanges, and provides additional evidence that this uneven distribution can be used to enhance pump-and-dump schemes. Although this study was focused on trade-specific fraud, the proposed methodology seems to be applicable to different areas. Abstraction of the methodology to general norm learning agents along with the extension to coalition forming can be regarded as future research directions.

Acknowledgments

This work was partly funded by the Dutch Research Council (NWO) for the HUMAINER AI project (KIVI.2019.006).

[4]Value equal to 55 is roughly the same value the Bitcoin market manipulator have used in 2017/18 to sell enough Bitcoins before the date of publishing the end of month audit statements.

References

[1] J. Xu, B. Livshits, The anatomy of a cryptocurrency pump-and-dump scheme, in: 28th USENIX Security Symposium (USENIX Security 19), 2019, pp. 1609–1625.

[2] J. Nicholls, A. Kuppa, N. A. Le-Khac, Financial cybercrime: A comprehensive survey of deep learning approaches to tackle the evolving financial crime landscape, IEEE Access 9 (2021) 163965–163986.

[3] A. Reurink, Financial fraud: a literature review, Journal of Economic Surveys 32 (2018) 1292–1325.

[4] F. Allen, D. Gale, Stock-price manipulation, The Review of Financial Studies 5 (1992) 503–529.

[5] F. Victor, T. Hagemann, Cryptocurrency pump and dump schemes: Quantification and detection, in: 2019 International Conference on Data Mining Workshops (ICDMW), 2019, pp. 244–251.

[6] E. van Asselt, S. Osinga, H. Bremmers, Simulating compliance behaviour for effective inspection strategies using agent based modelling, British Food Journal (2016).

[7] B. Ross, L. Pilz, B. Cabrera, F. Brachten, G. Neubaum, S. Stieglitz, Are social bots a real threat? an agent-based model of the spiral of silence to analyse the impact of manipulative actors in social networks, European Journal of Information Systems 28 (2019) 394–412.

[8] S. D. Silver, M. Raseta, A. Bazarova, Dynamics of phase transitions in expectations for financial markets: An agent-based, multicomponent model, Journal of Behavioral Finance 23 (2022) 92–105.

[9] C. Yagemann, S. P. Chung, E. Uzun, S. Ragam, B. Saltaformaggio, W. Lee, On the feasibility of automating stock market manipulation, in: Annual Computer Security Applications Conference, ACSAC '20, New York, NY, USA, 2020, p. 277–290.

[10] E. A. Lopez-Rojas, S. Axelsson, A review of computer simulation for fraud detection research in financial datasets, in: 2016 Future Technologies Conference (FTC), December, IEEE, 2016, pp. 932–935.

[11] J. Brito, P. Campos, R. Leite, An agent-based model for detection in economic networks, in: Highlights of Practical Applications of Agents, Multi-Agent Systems, and Complexity: The PAAMS Collection, Springer International Publishing, Cham, 2018, pp. 105–115.

[12] V. Gowadia, C. Farkas, M. Valtorta, Paid: A probabilistic agent-based intrusion detection system, Computers & Security 24 (2005) 529–545.

[13] J. M. Griffin, A. Shams, Is Bitcoin Really Untethered?, Journal of Finance 75 (2020) 1913–1964.

[14] P. Fratrič, G. Sileno, S. Klous, T. van Engers, Manipulation of the Bitcoin market: an agent-based study, Financial Innovation 8 (2022).

[15] F. Hartig, J. M. Calabrese, B. Reineking, T. Wiegand, A. Huth, Statistical inference for stochastic simulation models–theory and application, Ecology letters 14 (2011) 816–827.

[16] M. Marchesi, S. Cincotti, S. M. Focardi, M. Raberto, The genoa artificial stock market: Microstructure and simulations, in: Heterogenous Agents, Interactions and Economic Performance, Springer, 2003, pp. 277–289.

[17] M. Sunnåker, A. G. Busetto, E. Numminen, J. Corander, M. Foll, C. Dessimoz, Approximate Bayesian Computation, PLoS Computational Biology 9 (2013).