



Primary submission: 01.03.2023 | Final acceptance: 05.11.2023

Improved Sparse Mean Reverting Portfolio Selection Using Simulated Annealing and Extreme Learning Machine

Attila Rácz  and Norbert Fogarasi 

ABSTRACT

We study the problem of selecting a sparse, mean reverting portfolio from a universe of assets using simulated annealing (SA). Assuming that assets follow a first order vector autoregressive process (VAR(1)), we make a number of improvements in existing methods. First, we extend the underlying asset dynamics to include a time-independent additive term, thereby enriching the model's applicability. Second, we introduce Extreme Learning Machine (ELM) to decide whether to apply SA or settle for the much faster greedy solution. Finally, we improve the SA method by better calibration of the initial temperature and by determining the exact value of the weights within a selected dimension using the Rayleigh quotient. On real data, these changes result in more than 90% improvement in run time on average and 4.78% improvement in optimized mean reversion in our simulations. We also test the trading performance of our method on both simulated and real data and managed to achieve positive mean trading results in both cases.

KEY WORDS:

sparse portfolios, mean reverting portfolios, simulated annealing, extreme learning machine, machine learning.

JEL Classification: C650.

Budapest University of Technology and Economics, Department of Networked Systems and Services, Hungary

1. Introduction

Financial performance of various investments is influenced by a wide range of factors which have been studied extensively (Cruz Cárdenas et al., 2022; Mysaka & Derun, 2021; Sierpinska-Sawicz & Sierpinska, 2021). Amongst these, applying computational methods to the world of finance and investing has attracted considerable attention in recent years (Sasidharan et al., 2023; Sullistiawan et al., 2023). One of the classical problems of computational finance is to create a portfolio from a high number of available financial products that is optimal in some sense. The aim of the present work is to create a portfolio of which predictability is high

so the future value can be inferred with very good precision in a way that it has a very low number of constituents. Portfolios that follow mean reverting stochastic processes meet this criteria (see Banerjee et al., 2008; d'Aspremont, 2011; Fogarasi & Levendovszky, 2012; Fogarasi & Levendovszky, 2011). The mean-reverting property is widely used in economics (Ahsan & Rub, 2017; Madan & Wang, 2022; Narula, 2018; Tah, 2018). An efficient trading logic can be built based on the property of mean reverting time-series (Leowski, 2017), like bid-ask spread trading (Isaenko, 2018), pairs trading (Wu et al., 2020; Zhang, 2021). Two properties, the sparsity and the measure of predictability, the speed of mean reversion compete with each other during the

Correspondence concerning this article should be addressed to:

Adrian Peretz, Budapest University of Technology and Economics,
Budapest, Műegyetem rkp. 3, 1111 Hungary
E-mail: faus2s@yahoo.com

optimization. The speed is proportional to predictability while the sparsity (the number of different stocks used in the portfolio) is important for investors as this minimizes the transaction cost and makes the portfolio easier to interpret. This optimization can be done using simple and fast approximation and much more complex and expensive methods (see d'Aspremont, 2011; Fogarasi & Levendovszky, 2011; Stübinger & Endres, 2021; Yang et al., 2017; Wei, 2022)). It was shown that further optimization such as SA starting from the greedy solution can reach a solution closer to the global one (Fogarasi & Levendovszky, 2011). However, in the majority of cases, the less expensive and simpler greedy algorithm can find the global optimum. To utilize this, we propose using ELM neural networks, which helps decide whether a complex, but powerful algorithm, such as SA, should be used. We tested the algorithms on both simulated and real data.

The structure of the paper is the following:

- In section Different models on VAR(1), after a short summary about mean reverting process we describe possible extensions to the modeling of future values of stocks and how this modifies the selection.

- In section Comparison of exhaustive and greedy methods, we compare the exhaustive and greedy methods, results and the application of the Extreme Learning Machine.

- In section Simulated Annealing, we detail the changes applied for the Simulated Annealing.

- In section Performance test, we check the impact of the changes we made.

- In section Conclusions and future works, we make conclusions and recommendations.

2. Different models on VAR(1)

This section briefly explains mean reverting processes and the modeling of the stock data with VAR(1). Since in the long term the value of the process oscillates around its average value, when the price is below its mean it will more likely increase rather than decrease. This observation makes building a simple trading strategy possible and enables estimation of the trading range for the portfolio. The simplest example of a mean reverting process is the Ornstein-Uhlenbeck process which is

the continuous version of autoregression processes with parameter 1, (AR(1)).

2.1. Ornstein-Uhlenbeck process

The process is a stationary Gaussian-Markov process, a random walk with higher attraction to mean than diversion. Our mean reverting portfolio p_t is composed of the linear combination of stock prices. The stochastic differential equation that describes the Ornstein-Uhlenbeck process is

$$dp_t = \lambda(\mu - p_t)dt + \sigma dZ_t \quad (1)$$

where Z_t is a standard Brownian motion, λ is the speed of mean reversion and μ is the long term mean of the process which it is reverting to.

2.2. Asset Dynamics and Portfolio Selection

Is the sentence "I always lie" true or false? This small piece of logic has challenged philosophers from around 400 years B. C. to

2.2.1. Modeling Asset Dynamics with VAR(1)

We can model the available time series of a population of assets with discrete stationary VAR (1) process. Let $s_{i,t}$ denote the price of the asset i at time t where $i = 1, \dots, n$ and s_t the n -length vector which elements are the $s_{i,t}$. The stationary vector autoregression process can be written in the following form

$$s_t = As_{t-1} + W_t \quad (2)$$

Where A is the matrix of auto-regression, W_t is uncorrelated white noise. One can create a portfolio from the available assets to mitigate risks. Let P be a real valued vector representing the weights of the portfolio. The time evolution of the value of our portfolio can be written as

$$Ps_t = PAs_{t-1} + PW_t \quad (3)$$

We can define the measure of predictability by the following (Box & Tiao, 1977; d'Aspremont, 2011):

$$v = \frac{\sigma_{t-1}^2}{\sigma_t^2} \quad (4)$$

where σ_t^2 is the variance of the time series. If the denominator is larger, s_t will be pure noise as t goes to infinity, on the other hand, if the nominator is larger s_t will be perfectly predictable. After using the definition of predictability (4) for the VAR (1) model we get:

$$v(P) = \frac{\text{var}(P^T A s_{t-1})}{\text{var}(P^T s_{t-1})} = \frac{E(P^T A s_{t-1} s_{t-1}^T A^T P)}{E(P^T s_{t-1} s_{t-1}^T P)} \quad (5)$$

Maximizing predictability is eventually a generalized eigenvalue problem:

$$P_{opt} = \text{argmax}(v(P)) = \text{argmax}\left(\frac{P^T A G A^T P}{P^T G P}\right) \quad (6)$$

The argument of the argmax operator is the so called Rayleigh quotient. The above becomes

$$A G A^T P = \lambda P \quad (7)$$

If we use several assets to create a portfolio, the transaction cost will be high. To reduce that we will need to apply an additional constraint to the optimization. On the other hand, in order to hold the transaction cost as low as possible and also to keep the portfolio complexity low, only a low number of stocks will need to be obtained. The optimization problem now is the trade-off between the maximization of mean reversion speed and the minimization of the cardinality of stocks. Mathematically, we can introduce an additional constraint to equation (6) as follows:

$$P_{opt} = \text{argmax}(v(P)) = \text{argmax}\left(\frac{P^T A G A^T P}{P^T G P}\right), \quad \text{subject to } \text{Card}(P) \leq k \quad (8)$$

The autoregression matrix in equation (2) can be approximated using least squares regression (d'Aspremont, 2011; Fogarasi & Levendovszky, 2011) as follows

$$\hat{A} = (s_{t-1}^T s_{t-1})^{-1} s_{t-1}^T s_t \quad (9)$$

The model in equation (2) can be extended to the non-stationary VAR(1) model that contains a time independent constant scalar shift term to describe drift or to ensure positivity of the elements for all t :

$$s_{t+1} = c + A s_t + W_t \quad (10)$$

where A is an $n \times n$ real matrix constant at some certain time period, c is a time independent real scalar constant, W_t represents the noise or error term of the model with zero mean value, some constant variance and uncorrelated across time. This can be rewritten in concise VAR(1) notation by incorporating a shift into the matrix of autoregression:

$$s'_{t+1} = A' s'_t + W'_t \quad (11)$$

$$\begin{bmatrix} s'_{t+1} \\ \vdots \\ s'^n_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{1,1} & \dots & a_{1,n} & c_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n,1} & \dots & a_{n,n} & c_n \\ 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} s'_t \\ \vdots \\ s'^n_t \\ 1 \end{bmatrix} + \begin{bmatrix} W'_t \\ \vdots \\ W'^n_t \\ 0 \end{bmatrix} \quad (12)$$

where A' refers to a $(n + 1) \times (n + 1)$ matrix in which the last column is filled with the constant shift c , the last row has 0's except the $(n + 1)$ st element which should be strictly 1, x'_t a vector with $n + 1$ elements strictly 1 at the $(n + 1)$ st element and W'_t still provides the noise as in the previous case except no noise for the $(n + 1)$ st element (Lütkepohl, 1993).

2.2.2. Predictability in Case of Constant Shift

In case we add the shift and do not renormalize to zero mean, we will have to modify the calculation of predictability derived in (5). Otherwise it does not capture the covariance matrix.

$$v(P) = \frac{\text{var}(P^T A s_{t-1})}{\text{var}(P^T s_{t-1})} = \frac{E(P^T A s_{t-1} s_{t-1}^T A^T P)}{E(P^T s_{t-1} s_{t-1}^T P)} \quad (13)$$

Applying the linearity of the expectation operator and consider that $E(s_{t-1} - m) = 0$ where m is the mean of the process s_t , we get

$$v(P) = \frac{P^T A G A^T P + P^T A m^2 A^T P}{P^T G P + P^T m^2 P} \quad (14)$$

So the quantity that we have to maximize is

$$v(P) = \frac{P^T A(G + m^2)A^T P}{P^T(G + m^2)P} \quad (15)$$

Which can give back the original case (5) when $m = 0$.

2.3. Parameter Estimation

To verify the estimation of process parameters we simulate VAR (1) including the constant shift term. The latter has a technical property that ensures the positivity of the values of time series at every time step t . Figure 1 shows the comparison of shift applied for

simulation against the estimated from simulated time series and the applied and calibrated noises.

Figure 2 shows the estimation error with and without taking into account the time independent shift for two different population sizes.

The estimation error is the mean squared error of the difference of simulated and regressed data. The figures clearly indicate that if the model does not capture the details, the regression matrix estimation will be inaccurate.

Figure 3 presents an individual simulated time series

with the estimated noiseless curve using the extended VAR(1) model.

Figure 4 shows an example individual time series with regression including the result with and without taking into account the constant shift in the calibration.

2.3. Trading Strategy

As the portfolio is set up assuming it follows the Ornstein-Uhlenbeck process (for discrete timeseries AR(1)), the applied trading strategy also takes advantage of this. For this reason we need two bounds

Figure 1
Overview of the Development of First- and Second-order Concepts

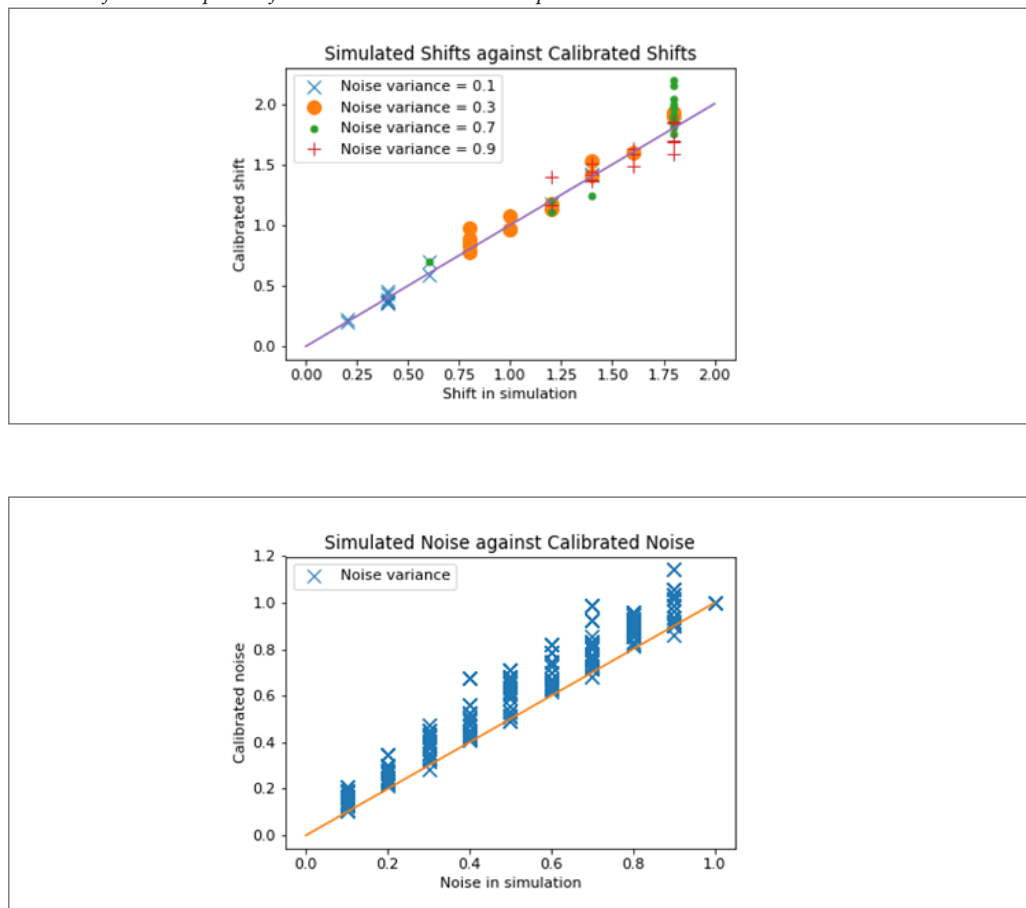


Figure 2
Estimated Error Comparison

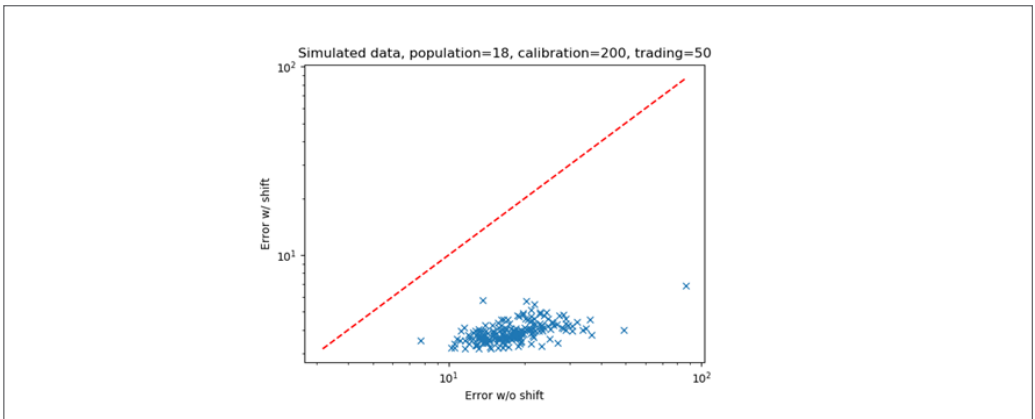
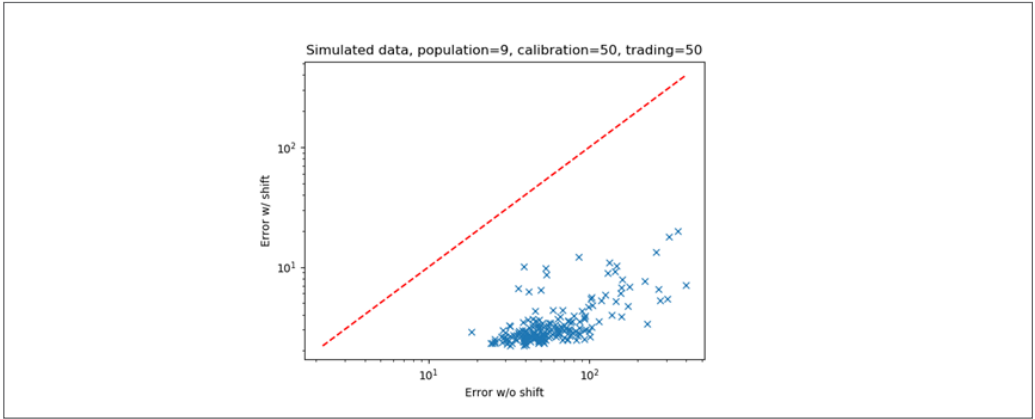


Figure 3
Single Simulated Time Series Out of 8 With Shift=5

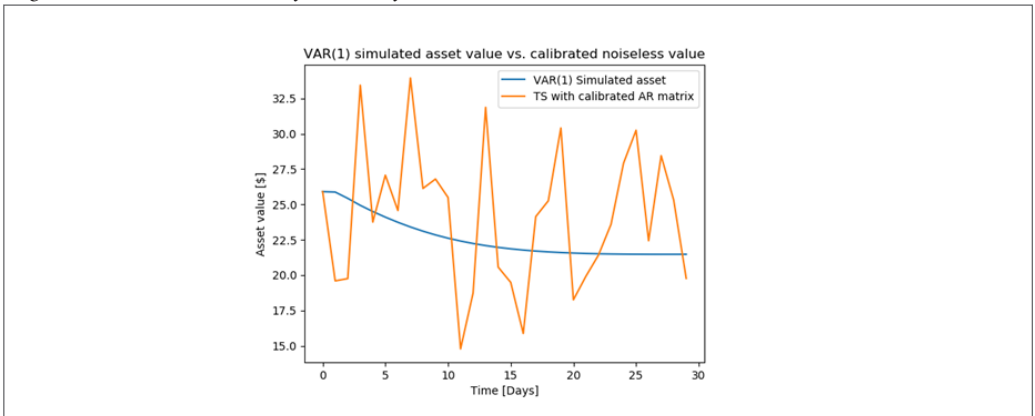
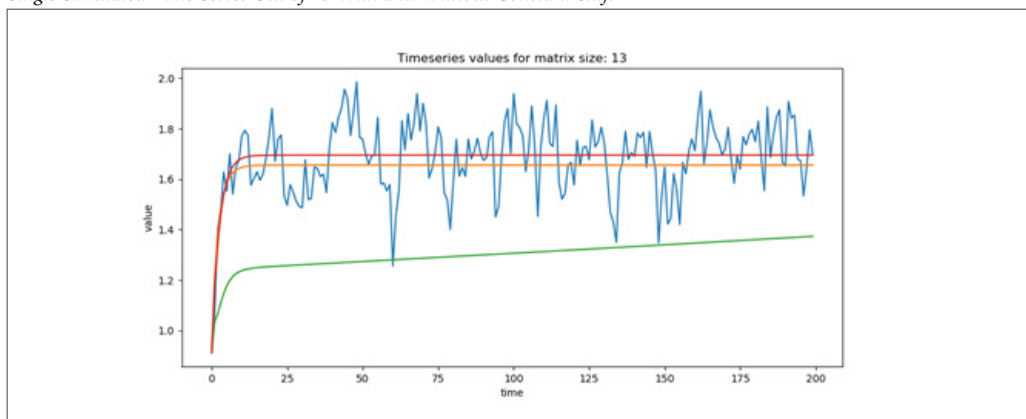


Figure 4*Single Simulated Time Series Out of 13 With and Without Constant Shift*

Note. These show that the shifts and noise variance have been captured correctly and demonstrate that the prediction error can be diminished by an order of magnitude on simulated data.

that trigger the selling and buying of the portfolio. We estimate the mean and the standard deviation (std) of the portfolio during the identification period. The lower bound of the trading interval is the portfolio value below mean with the quantity of std (mean – std) while upper is the mean + std. In case we do not have portfolio and its actual value is below the lower bound, we purchase using all available cash and sell the whole if we are above the upper bound (Fogarasi & Levendovszky, 2011). Figure 5 shows the effect of the shift for a real stock. The improvement of the accuracy of the prediction is clear, however we utilize this directly in the creation of the portfolio.

3. Comparison of Exhaustive and Greedy Methodst

Solving optimization problem (8) can happen in several ways. The simplest way is the exhaustive method where the global minimum is found by sweeping through the whole configuration space. This is a brute force method, that runs through all possible configurations and selects the one with the highest generalized eigenvalue. The method is accurate with very high computational cost (d'Aspremont, 2011). An efficient alternative way to get near optimal solution is the greedy search. The greedy method starts with the largest diagonal

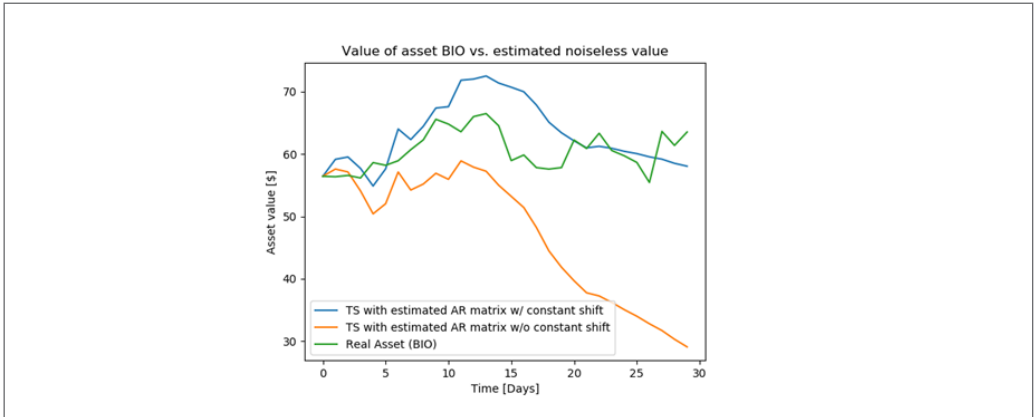
element in the matrix, then it chooses a subspace such that the generalized eigenvalue is the highest among the others. This for sparsity $L = 1$ and $L = N$ this should result in the same as for exhaustive. Figure 7 presents the comparison of the eigenvalues coming from the exhaustive and the greedy methods for all sparsities in a certain configuration.

The result of exhaustive and greedy methods are the same in most cases as we observed by running a high number of tests. Thus, the validity of the greedy method is experimentally confirmed. Our goal is to find cases when greedy solution is sub-optimal and then to use it as an initial point for further optimization process for SA (Fogarasi & Levendovszky, 2011).

3.1. Classification with Extreme Learning Machine (ELM)

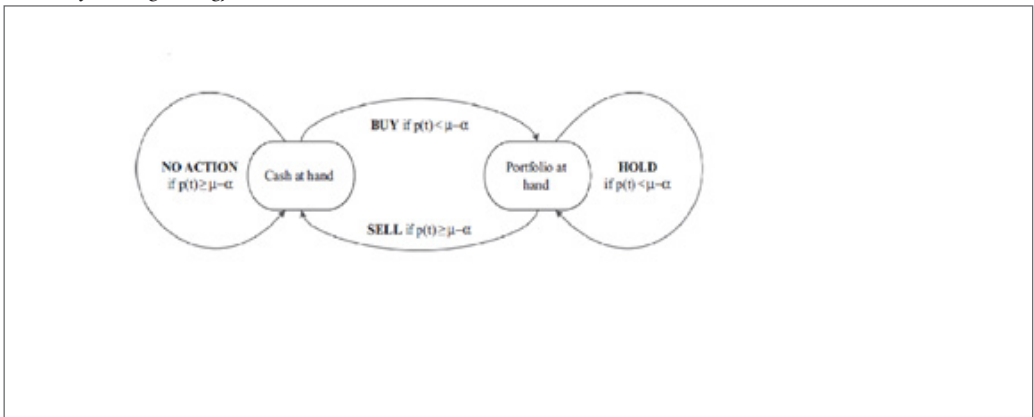
As SA is a complex and expensive method and most of the times greedy method can reach the global optimum, no further optimization is necessary. In order to decide whether we should start SA we apply machine learning algorithms. Extreme Learning Machine is a single-hidden layer feed-forward neural network (SLFN) in which the hidden nodes are chosen randomly although the output weight is chosen deterministically (Huang

Figure 5
The Effect of the Shift for a Real Stock



Note. Using 250-day-long identification window. The full population is 250. The blue line is the simulated data, the red line represents the regressed curve generated with the original autoregression matrix, the orange curve is the regression result with the calibrated matrix incorporating the shift and the green one is the same except ignoring the shift term.

Figure 6
Scheme of Trading Strategy.



Note. Original figure from: (Fogarasi & Levendovszky, 2011)

et al., 2006). The term 'extreme' comes from the extreme fast learning speed, being thousands of times faster than traditional feed-forward methods like backpropagation. Furthermore, it is easy to implement and there is a very small number of training errors and norm of weights. Figure 8 summarizes the scheme of network structure.

It is very interesting and surprising that, unlike the most common understanding, all the pa-

rameters of SLFNs need to be adjusted, the input weights, W_i and the hidden layer biases, b_i are in fact not necessarily tuned and the hidden layer output matrix H can actually remain unchanged once random values have been assigned to these parameters in the beginning of learning. For fixed input weights W_i and the hidden layer biases b_i , to train an SLFN is simply equivalent to finding a least squares solution $\hat{\beta}$ of the linear system $H\beta = T$.

Figure 7
Comparison of Normalized Eigenvalues of Greedy and Exhaustive Eigenvalues

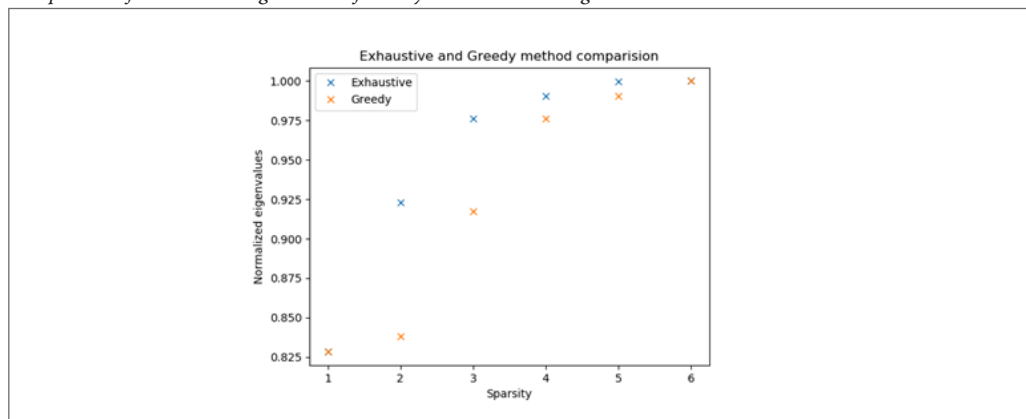
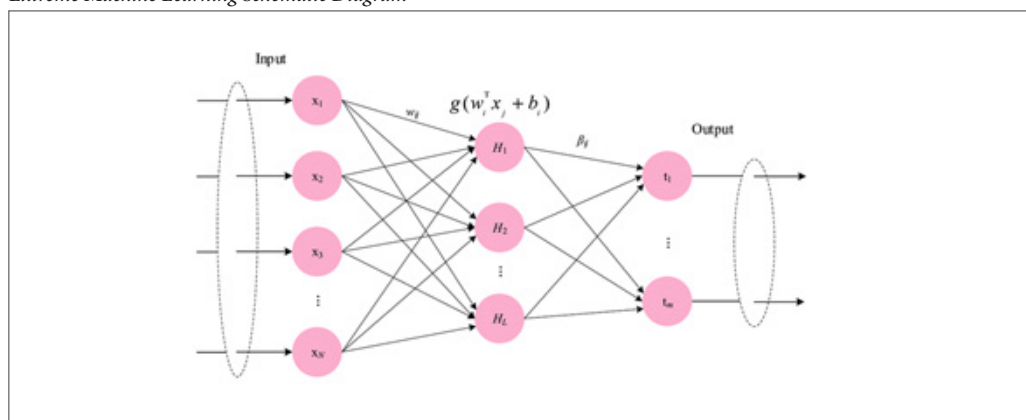


Figure 8
Extreme Machine Learning Schematic Diagram



Note. Original figure from: (Deng et al., 2019)

The number of the hidden nodes is usually much less than the number of the distinct training samples, H is a non-square matrix. The smallest norm least-square solution of $H\beta = T$ is:

$$\hat{\beta} = H^+T \tag{16}$$

where H^+ is the so-called Moore-Penrose generalized inverse of H . The ELM algorithm is as follows:

- Assign random numbers to input weights.
- Calculate the hidden layer output matrix H
- Calculate the output weights β

$$X = W_1 X_{train} \tag{17}$$

$$W_1 = (X^T X)^+ X Y \tag{18}$$

3.2. Training ELM

The training was performed with every regression matrix size from 5 to 30 and every cardinality from 3 to 'matrix size-2'. The population for every configuration was 1000 and 70% was the learning population, 30% was used for validation. Table 1 shows the accuracy of the training for different hidden unit sizes.

To test the efficiency of the graph optimization, we compare results (runtime, return, etc) to the original SA outcome. In order to make use of SA, it is important to find the sub-optimal solutions. On the other hand, it is not a problem to start the Metropolis algorithm from the exhaustive solution. The input consists of 800 elements, each 14 units long. We can see that in Table 1 the accuracy reaches its maximum value with 10 hidden layers. In some cases the accuracy for the number of hidden layers below 10 can be as low as 0.15. The cause of the instability is in calculation of the output weights which contains the Moore-Penrose inverse. In practice one can see that the product above does not give the exact identity matrix back.

The comparison of greedy and exhaustive methods was carried out with real data. The population of stocks was chosen randomly in every configuration and the size of the regression matrix ranged from 5 to 21. Table 2 sum-

marizes the statistics.

During the training of classifications we indicated the run when the difference of the eigenvalues between greedy and exhaustive was larger than 0.008. The training contained only the eigenvalue of the particular configuration (size and sparsity) and the indicator value. The number of layers was 4.

4. Simulated Annealing

4.1. Introduction

Simulated annealing is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is metaheuristic to approximate global optimization in a large search space for an optimization problem. It is often used when the search space is discrete (e.g., the traveling salesman problem). For problems where finding an approximate global optimum is more important than

Table 1
Accuracy of Method

Hidden Units	Accuracy
1	0.85
2	0.85
3	0.85
4	0.85
5	0.9625
6	0.9875
7	0.9791
10	1.0
15	1.0

Table 2
Testing the Result of Classification

Size/Sparsity	False negative	False Positive	True Negative	True Positive
17/14	0	0	36	1003
15/12	0	0	37	992
14/6	0	0	130	898
12/7	0	0	97	998
9/5	0	0	70	1015
8/5	0	0	59	1048

finding a precise local optimum in a fixed amount of time, SA may be preferable to alternatives such as gradient descent.

This notion of slow cooling implemented in the SA algorithm is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored. Accepting worse solutions is a fundamental property of metaheuristics because it allows a more extensive search for the global optimal solution. In general, the SA algorithms work as follows. At each step, the algorithm randomly selects a solution that is close to the current one, it measures its quality and then it decides to move toward or to stay with the current solution based on either one of two probabilities between which it chooses on the basis of the fact that the new solution is better or worse than the current one. During the search, the temperature is progressively reduced from an initial positive value to 0 which affects the two probabilities: at each step, the probability of moving to a better new solution is either kept to 1 or changed to a positive value; on the other hand, the probability of moving to a worse new solution is progressively changed towards zero (Salamon et al., 2002), (Geman & Geman, 1984).

4.2. Optimization on Graph

The method suggested by Fogarasi and Levendovszky (2011) works in an arbitrarily large number of dimensional space as the weight of the constituent can vary continuously. In this method we only select the assets with non-zero weights, however, the weights themselves are calculated by equation (6) ie. Rayleigh quotient. We treat

two configurations as connected nodes when they differ from each other by exactly one element. Hence, we can define a graph with the non-zero weights and the optimization happens on this graph by walking through the nodes. The expectation is that we can reach the minimum with much fewer function evaluation than before. Figure 9 is a schematic representation of the configuration space in which the SA method operates.

4.3. Calibrating Initial Temperature

It is essential to set the initial temperature of the optimization process properly. One way to do this is to start SA from the result of the greedy method. In case the result of the greedy method is highly suboptimal when the temperature is high, the probability of positive transitions is 1. On the other hand if it is too low then the system cannot break out from the local minimum. A commonly applied technique to calibrate initial temperature is to declare the initial probability of positive transition and derive the corresponding temperature from this. The brief description of the method we applied (Ben-Ameur, 2004) is as follows.

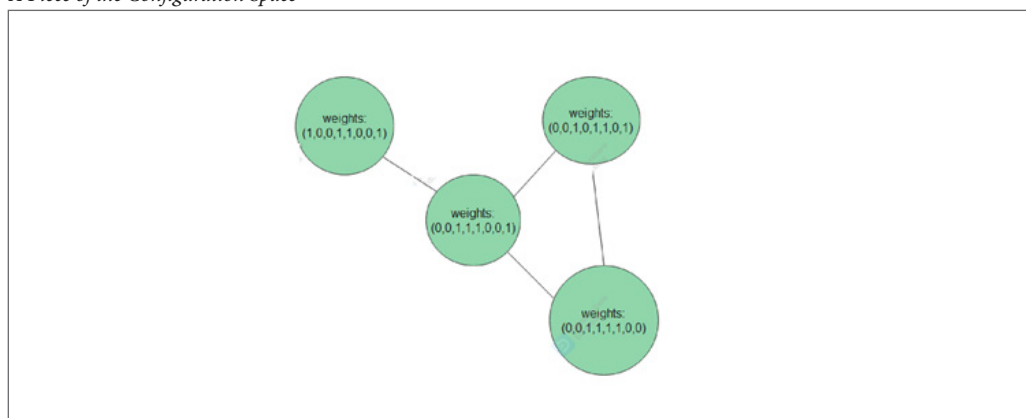
- Collect positive transitions (S) and set χ^0 to the required transition probability.
- Set $T^1 > 0$ and $n = 1$.

$$\bullet \chi(T_n) = \frac{\sum_{t \in S} \exp(-\frac{E_{max}}{T_n})}{\sum_{t \in S} \exp(-\frac{E_{min}}{T_n})}, \text{ where } E_{min} \text{ and } E_{max} \text{ are the energies before and after transition respectively.}$$

- if $|\chi(T_n) - \chi^0| < \epsilon$ then $T_{min} = T_n$ else $T_{n+1} = T_n (\ln(\chi(T_n)) / \ln(\chi^0))^{1/p}$, $n = n + 1$.

Figure 9

A Piece of the Configuration Space



5. Performance test

During the test of the effectiveness of the improvement of the graph based SA some number of stocks were selected from S&P500 pool (ie. 20 in Figure 10 and 36 in Figure 11) with a random sparsity (15 and 31 respectively). The calibration time window was 350 days. Each point in the figures represents a random selection from the pool. The initial points of the SA were not necessarily the result of the greedy algorithm.

The new method results in a better optimum in all cases tested, resulting in 4.78% improvement on average.

We tested the performance of the SA to the previously detailed modifications. We measured how the energy of the local optimum, the running time, the number of function calls of the proposed method changed. Figure 10 and Figure 11 illustrate the test results.

We demonstrate that we can reach better solutions with fewer function evaluations and more than 90% improvement in running. We tested the performance of the algorithm on simulated and real data for S&P500 gathered from Yahoo finance for

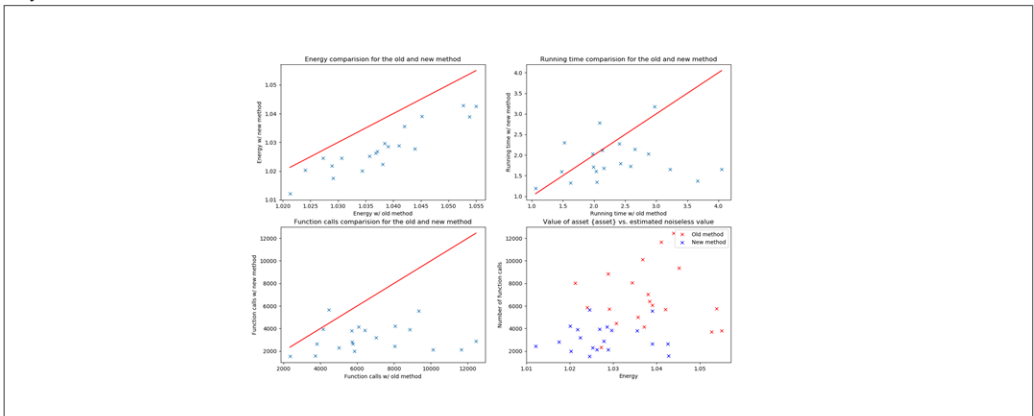
the period between 01/01/2016 and 12/31/2020. In every test case the selected stocks were chosen randomly from the whole S&P500 population. The sparsity varied from 3 to 12 with a step size of 3. The length of the identification window was 100 days. We first validated the improvements on simulated data. We generated 20 time-series using VAR(1) with time independent shift. On figure 12 we show the profit histogram of 220 simulations.

For the sake of representation we accumulated the profits above \$1, 000 into the highest bin (the one near to \$1, 000 in Figure 12). The results prove that the simple trading strategy can be well applied to the extended VAR(1) model. In Table 3 we provide the statistics of the profits trading on simulated data.

Figure 13 shows the histogram of the profit for a portfolio constructed from real-time series.

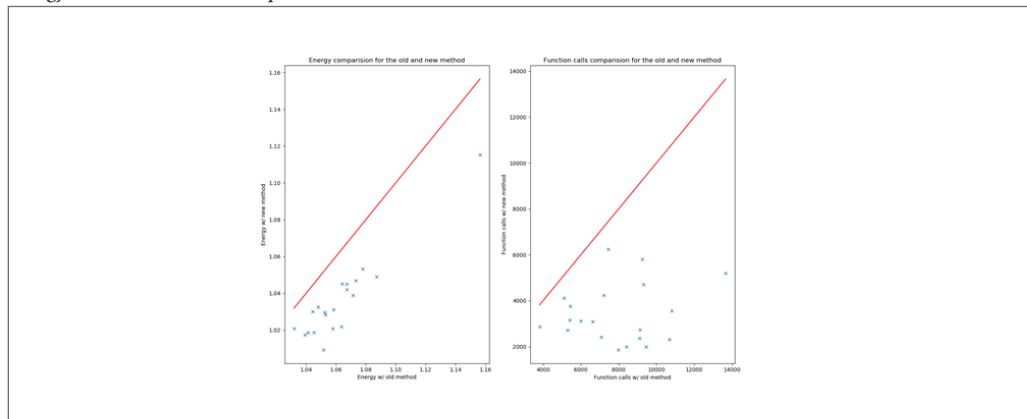
The length of the trading time window was 100 days in every case. The range of the histogram was fixed from \$ - 550 to \$2, 000 with 25 bins. Returns under the lower limit and above the upper limit were floored to their respective values similarly to the simulated case. In Table 4 we provide the statistics of the

Figure 10
Performance Test



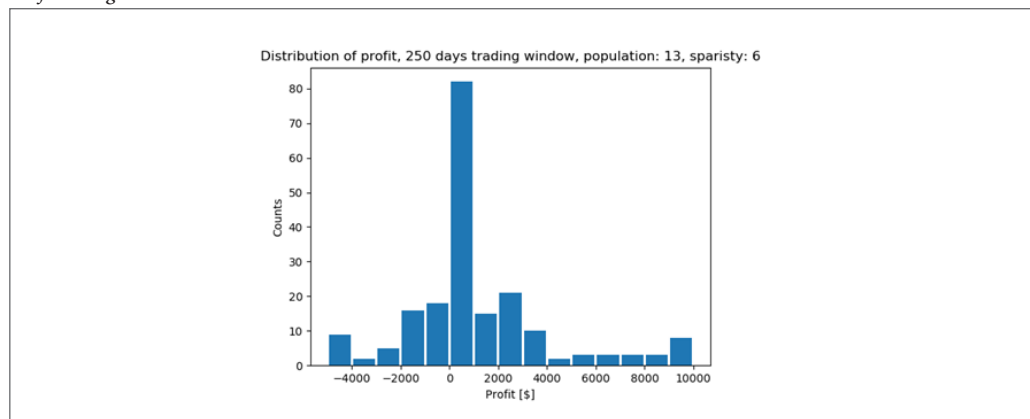
Note: The upper left graph compares the minimal energy achieved by the two methods. The red line, $x = y$ represents cases where the two methods yield the same result. The full runtime comparison illustrated in the upper right graph whilst the lower left graph shows the number of function calls. In the lower right graph we visualize how the two methods differ in terms of energy and function calls. P opulation = 20, sparsity = 15, T_{id} = 350 days

Figure 11
Energy and Function Call Comparison



Note: Energy and function calls comparisons for the two methods, population = 35, sparsity = 31, Tid = 350 days

Figure 12
Profit Histogram



Note: Using simulated time series with population size 20. The horizontal axis shows the profit in dollars, the vertical refers to the number of cases.

Table 3
The Mean and Standard Deviation

Sparsity	Mean Profit	STD of Profit
3	117.52	159.78
6	228.95	463.18
9	159.89	270.46
12	85.62	127.16
Total	168.09	328.32

profits trading on real data. Note that the above explained flooring was not applied in the tables only in the figures for sake of representation.

As seen, the mean profit on both simulated and real data is positive but with a sizable standard deviation, causing a number of runs to be unprofitable. This is primarily due to the use of a limited time window which may result in losing portfolios remaining on-hand at the end of the arbitrary time window.

As seen in Figure 14 and 15, the movement of the portfolio is predicted by our model accurately in most cases, implying that the selected portfolio is mean-reverting.

The amount of maximum investable money at the beginning of learning was \$10,000. The profit can be very sensitive to the order of magnitude of our money

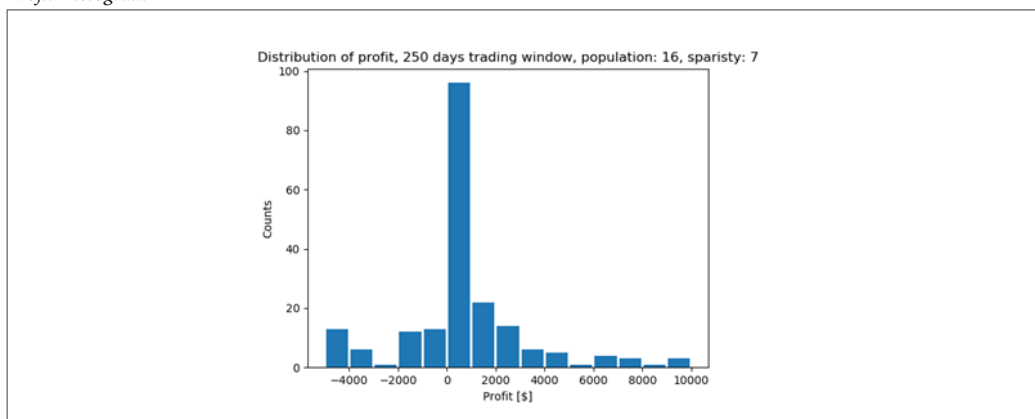
since we must invest in an integer number of stocks. As it can be seen, the amount of losses are not negligible in the case of both simulated and real data. There are two main reasons for this: first the trading strategy is not able to handle non-stationary dynamics and the estimation of the trading range is not accurate enough. Second, our trading time window has ended with a losing portfolio on hand.

6. Conclusions and Future Works

We extended the analytical model of asset dynamics by adding a time-independent drift term to capture asset movement more precisely. We presented the impact of this model change based on simulated and real S&P500 data. Furthermore, we also introduced an ELM-based decision process to save run-time not

Figure 13

Profit Histogram

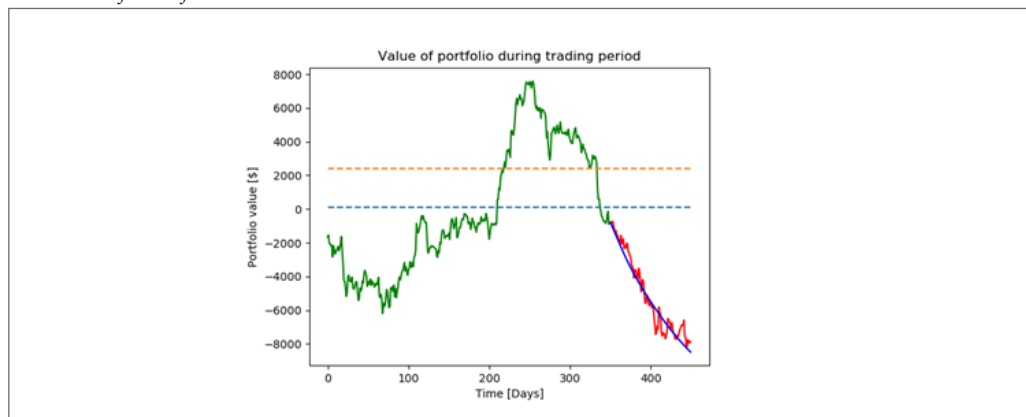


Note: Histogram of returns of 500 simulation with 100-day-long trading window, selecting 7 stocks from population size 16 and 50-day-long identification period.

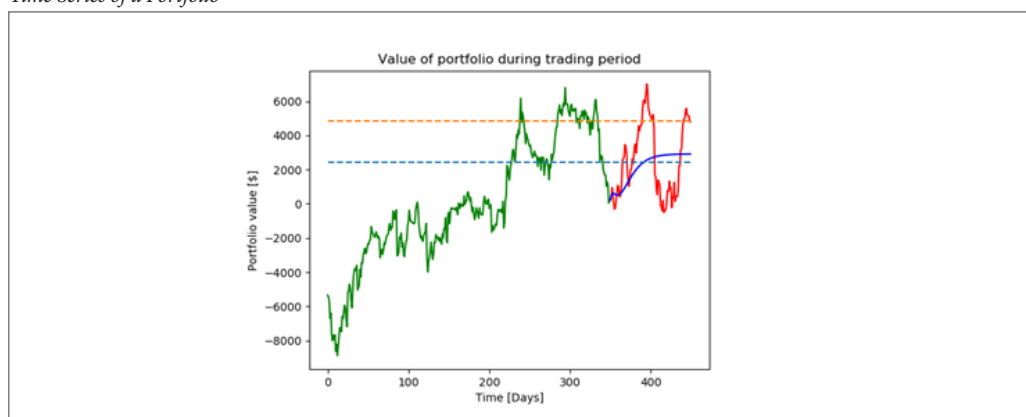
Table 4

The Mean and Standard Deviation

Sparsity	Mean Profit	STD of Profit
3	309.54	734.41
6	379.05	634.01
9	344.06	563.90
12	334.84	581.28
Total	340.77	641.02

Figure 14*Time Series of a Portfolio*

Note: With sparsity 12 selected from S&P500 assets. The solid green curve is the identification term of the portfolio, the solid red curve is the trading term. The solid blue is the estimated value of the portfolio using the extended V AR(1) model. The two dashed lines are the trading indicators.

Figure 15*Time Series of a Portfolio*

Note: With sparsity 12 selected from S&P500 assets. The solid green curve is the identification term of the portfolio, the solid red curve is the trading term. The solid blue is the estimated value of the portfolio using the extended V AR(1) model. The two dashed lines are the trading indicators.

to start SA every time as the greedy method can often reach the global optimum. Finally, we applied a method for initial temperature setting and used the generalized eigenvalue calculation to reduce the dimensionality of the problem. We have shown that these changes reduce the run-time consistently and improve the performance of the optimization in 100% of the cases tested. The method generates a positive mean profit but with a fairly large standard deviation due to the use of a limited time window which results in losing portfolios

in some cases. By further improving the mean reverting portfolio value prediction and adopting the trading algorithm to the prediction, we have great hope to further improve the trading performance of this method.

References

- Ahsan, T. & Rub, A. (2017). Capital structure mean reversion: A comparison between Chinese state-owned and private firms. *Global Business and Economics Review*, 19 (6), 687–695. <https://doi.org/10.1504/GBER.2017.087282>

- Banerjee, O., El Ghaoui, L., & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation. *Journal of Machine Learning Research*, 9, 485–516. <https://doi.org/10.48550/arXiv.0707.0704>
- Ben-Ameur, W. (2004). Calibrate initial temperature to actual energy values. see: Computing the initial temperature of simulated annealing. *Computational Optimization and Applications*, 29(3), 369–385. <https://doi.org/10.1023/B:CO-AP.0000044187.23143.bd>
- Box, G. E., & Tiao, G. C. (1977). A canonical analysis of multiple time series. *Biometrika*, 64(2), 355. <https://doi.org/10.1093/biomet/64.2.355>
- Cruz Cárdenas, J., Guadalupe-Lanas, J., Ramos-Galarza, C., & Palacio-Fierro, A. (2022). Factors influencing the consumer tendency to use mobile instant messaging (MIM): A mixed method study. *Contemporary Economics*, 16(3), 32–50. <https://dx.doi.org/10.5709/ce.1897-9254.467>
- d'Aspremont, A. (2011). Identifying small mean reverting portfolios. *Quantitative Finance*, 11(3), 351–364. <https://doi.org/10.48550/arXiv.0708.3048>
- Deng, W., Ye, B., Bao, J., Huang, G., & Wu, J. (2019). Classification and quantitative evaluation of eddy current based on kernel-pca and elm for defects in metal component. *Metals - Open Access Metallurgy JOURNAL*, 9(2), 155. <https://doi.org/10.3390/met9020155>
- Fogarasi, N., & Levendovszky, J. (2011). Sparse, mean reverting portfolio selection using simulated annealing. *Algorithmic Finance*, 2(3), 197–211. <https://doi.org/10.3233/AF-13026>
- Fogarasi, N., & Levendovszky, J. (2012). Improved parameter estimation and simple trading algorithm for sparse, mean-reverting portfolios. *Annales Universitatis Scientiarum Budapestinensis de Rolando Eötvös Nominatae*, 37(2), 121–144.
- Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 721–741. <https://doi.org/10.1109/TPAMI.1984.4767596>
- Huang, G.-B., Zhu, Q.-Y., & Siew, C.-K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, 70(3), 489–501. <https://doi.org/10.1016/j.neucom.2005.12.126>
- Isaenko, S. (2018). Optimal mean-reversion strategy in the presence of bid-ask spread and delays in capital allocations. *Quantitative Finance*, 18(12), 2051–2065. <https://doi.org/10.1080/14697688.2018.1484151>
- Leowski, H. (2017). Optimal Mean Reversion Trading. *Quantitative Finance* 17(8), 1159–1164. <https://doi.org/10.1080/14697688.2017.1317507>
- Lütkepohl, H. (1993). *New introduction to multiple time series analysis*. Springer.
- Ornstein, L. S., & Uhlenbeck, G. E. (1930). On the theory of the brownian motion. *Physical Review*, 36, 823. <https://doi.org/10.1103/PhysRev.36.823>
- Madan, D. and Wang, K. (2022) Stationary increments reverting to a Tempered Fractional Lévy Process (TFLP). *Quantitative Finance*, 22(7), 1391–1404. <https://doi.org/10.1080/14697688.2022.2060852>
- Mysaka, H., & Derun, I. (2021). Corporate financial performance and Tobin's Q in dividend and growth investing. *Contemporary Economics*, 15(3), 276–289. <https://dx.doi.org/10.5709/ce.1897-9254.449>
- Narula, I. (2018) Stock price randomness of BRICS nations. *International Journal of Public Sector Performance Management*, 4(2), 231–250. <https://doi.org/10.1504/IJPSPM.2018.090744>
- Salamon, P., Sibani, P., & Frost, R. (2002). Facts, conjectures, and improvements for simulated annealing. *SIAM Monographs on Mathematical Modeling and Computation. Society for Industrial and Applied Mathematics*, 36, 823. <https://doi.org/10.1137/1.9780898718300>
- Sasidharan, S., Ranjith, V. K., & Prabburam, S. (2023). Determinants of factors affecting the financial performance of Indian general insurance firm: Panel data evidence. *Contemporary Economics*, 17(2). <https://dx.doi.org/10.5709/ce.1897-9254.508>
- Sierpinska-Sawicz, A., & Sierpinska, M. (2021). Depreciation capital as a source of financing of mining companies activities. *Contemporary Economics*, 15(4), 429–442. <https://dx.doi.org/10.5709/ce.1897-9254.458>
- Stübinger, J. and Endres, S. (2021) Efficient computation of mean reverting portfolios using cyclical coordinate descent. *Quantitative Finance*, 21(4), 673–684. <https://doi.org/10.1080/14697688.2020.1803497>
- Sullistiawan, D., Rudiawarni, F. A., Feliana, Y. K., & Grigorescu, A. (2023). Do investors overreact to COVID-19 outbreak? An experimental study using sequential disclosures. *Contemporary Economics*, 17(1), 43–57. <https://dx.doi.org/10.5709/ce.1897-9254.498>
- Tah, K. (2018) Random walk and structural break in exchange rates. *International Journal of Monetary Economics and Finance*, 11(4), 384–393. <https://doi.org/10.1504/IJMEF.2018.095744>

- Yang, X., Li, H., Zhang, Y. and He, J. (2017) Reversion strategy for online portfolio selection with transaction costs. *International Journal of Applied Decision Strategies*, 11(1), 79-99. <https://doi.org/10.1504/IJADS.2018.088632>
- Wei, C. (2022) Parameter estimation for mean-reversion type stochastic differential equations from discrete observations. *International Journal of Computing Science and Mathematics*, 15(2), 117-131. <https://doi.org/10.1504/IJCSM.2022.124000>
- Wu, L., Zang, X. and Zhao, H. (2020) Analytic value function for a pairs trading strategy with a Lévy-driven Ornstein-Uhlenbeck process. *Quantitative Finance*, 20(8), 1285-1306. <https://doi.org/10.1080/14697688.2020.1736613>
- Zhang, G. (2021) Pairs trading with general state space models. *Quantitative Finance*, 21(9), 1567-1587. <https://doi.org/10.1080/14697688.2021.1890806>