Selection and LSTM based trading of sparse, optimal portfolios using the VAR(p) model

Attila Rácz
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
0000-0002-7432-5911

Norbert Fogarasi
Department of Networked Systems and Services
Budapest University of Technology and Economics
Budapest, Hungary
fogarasi@hit.bme.hu

***Abstract*—Given multivariate time series of market observable security prices, this paper explores the use of $VAR(p)$ model to select sparse portfolios that maximize predictability and uses $LSTM$ neural network based trading strategies to assess their profitability. Several previous papers have used the $VAR(1)$ model to show the viability of this method for selecting sparse, mean reverting portfolios and have shown its profitability. This work builds on this and shows that higher dimensional $VAR(p)$ models can achieve a better fit of the multivariate time series and therefore result in higher profits using $LSTM$ neural network based convergence trading algorithms.**

***Index Terms*—Seq2Seq LSTM, algorithmic trading, neural network, portfolio optimization, vector autoregression**

I. Introduction

Finding accurate techniques for predicting financial time series is of extraordinary interest among researchers. However, stock price forecasting is quite challenging because of the significant noise, non-linearity, and volatility [1]–[7]. This paper focuses on forecasting sparse optimal portfolios with neural networks. The technique, which was originally developed for natural language processing, was applied here is $LSTM$ networks [8]. This network has already been applied for forecasting financial time series [9]–[12]. $Seq2Seq$ [13] architecture was applied with $LSTM$ networks (which are able to predict longer time range) to use our predicted data to estimate the trading range to set trigger points to sell or buy the portfolio [14]. These portfolios were created using $VAR(p)$ model fitted on the natural logarithm of the stock price data and the weights of the portfolio are coming from the eigenvector related to the maximal eigenvalue of the linear combination of the regression and the auto-correlation matrices using $Simulated\ Annealing$ (see [15]). In our previous work [16] we tried to forecast sparse, mean reverting portfolios using $LSTM$ networks and built a trading strategy that utilizes this prediction. However, it was shown during the test that in most cases the trading ranges were estimated inaccurately, so few events occurred. The aim of this article is to build a neural network that is able to predict a proper trading range rather than the estimation of the long term mean. A more complex trading logic was built that incorporates both the $Seq2Seq\ LSTM$ network which predicts the range and the simple $LSTM$ which is able to estimate upcoming portfolio value very precisely.

The structure of the paper is the following:

- In section II, the techniques such as how to create optimized portfolio with $VAR(p)$ model, time series predictions with $LSTM$ networks and the trading strategy used in the paper are discussed.
- In section III, the performance of strategy is demonstrated.
- In section IV, some possible future works are mentioned.

II. Materials and methods

A. Modeling stocks dynamics with VAR(p)

In previous works the stocks' dynamics were modeled with $VAR(1)$ model as that is the discrete model of the mean reversion process. The aim of increasing the parameter of VAR model is to use more precise quantitative model for portfolio creation. A step toward higher predictability defined in 4 is to increase the lag parameter. Higher lag parameter p involves more regression and covariance matrices in the optimization as detailed in $Section$ II-B. To check the effect of higher order lags in the auto-regression, the $VAR(p)$ model was fitted for $S\&P500$ stocks and the mean absolute error (MAE) was used as measure for precision. In a previous work [16] couple of data models such as the log, $diff$ and $log-diff$ were investigated and was found that fitting the $VAR(1)$ model on log data (meaning that taking the natural logarithm of stock values) gave the best results among the 4 data models.

B. Creating portfolio using VAR(p) model

As Show in the previous section, increasing the parameter of VAR model provides more precise quantitative model for portfolio creation. In order to utilize the extended, generalized model of the stock dynamics, we need to calculate the general formula of predictability. The portfolio is scalar expressed as a product of the stock vector S_t and the weight vector w:

$$P_t = \sum_{i=1}^{n} w_i S_{t,i} = w^T S_t \quad (1)$$

The formula of $VAR(p)$ model ([19]):

$$S_t = A_{t-1}S_{t-1} + A_{t-2}S_{t-2} + \cdots + A_{t-p}S_{t-p} + \epsilon_t \quad (2)$$

https://doi.org/10.3311/WINS2024-017

Putting this into the portfolio formula:

$$P_t = w^T S_t = w^T(A_{t-1}S_{t-1}+A_{t-2}S_{t-2}+\cdots+A_{t-p}S_{t-p}+\epsilon_t) \quad (3)$$

The predictability is the maximization of the Rayleigh quotient which is the ratio of the predicted portfolio at time t and the actual value at t:

$$\nu(w) = \frac{var(w^T M_t)}{var(w^T S_t)} = \frac{\mathbb{E}[w^T M_t M_t^T w]}{\mathbb{E}[w^T S_t S_t^T w]} \quad (4)$$

where

$$M_t = A_{t-1}S_{t-1} + A_{t-2}S_{t-2} + \cdots + A_{t-p}S_{t-p} \quad (5)$$

is the regressed vector. The denominator of (4) can expressed by definition as

$$\mathbb{E}[w^T S_t S_t^T w] = w^T G_t w \quad (6)$$

where G_t is the covariance matrix. The numerator of (4) is by using (5)

$$\begin{aligned}\mathbb{E}[w^T M_t M_t^T w] = {} & w^T A_{t-1}G_{t-1}A_{t-1}^T w \\ & + w^T A_{t-1}G_{t-1,t-2}A_{t-2}^T w \\ & + w^T A_{t-2}G_{t-2,t-1}A_{t-1}^T w \\ & + w^T A_{t-2}G_{t-2}A_{t-2}^T w + \ldots\end{aligned} \quad (7)$$

which can be simplified to

$$\mathbb{E}[w^T M_t M_t^T w] = \sum_{i,j=1}^{p} A_{t-i}G_{t-i,t-j}A_{t-j}^T \quad (8)$$

where $G_{t-i,t-j}$ is the autocovariance matrix of the stock values S_t with lags i and j and A_{t-i} is the regression matrix with lag i. In case of $p = 1$ the original result can be got back (see [15], [16]). Finding the maximum value of $\nu(w)$ in (4) is the following generalized eigenvalue problem:

$$(\sum_{i,j=1}^{p} A_{t-i}G_{t-i,t-j}A_{t-j}^T)w = \lambda G_t w \quad (9)$$

Finding the maximal value of λ is done in the same way as in [15]–[18]. The eigenvector related to the maximized eigenvalue represents the weights of the constituents of the optimized portfolio. Note that some weights can be negative which means that related stocks will be shorted.

C. Vanilla LSTM to predict 1 step ahead

Recurrent neural networks are capable to regress future prices of the portfolio using historical values. Particularly $LSTM$ networks are well usable for processing and predicting time series [8].

This particular network has a single hidden layer with 4 hidden units and a dense layer on top of $LSTM$. The optimizer is the $Adam$ [20] and the applied loss function was the $mean\ squared\ error$, (MSE). the network was trained to predict future value using 4 historical values.

As it was previously mentioned this single layer vanilla $LSTM$ network is used only to predict one time step ahead precisely. This single step prediction is used as final step in the trading logic to make it more effective in terms of profitability. The trading logic detailed in Section II-E. However, in order to make more efficient trading strategies, longer time interval estimation is needed which is described in the following subsection.

D. Seq2Seq LSTM

An architecture developed originally in the field of natural language processing is called the $Seq2Seq$. A special class of these problems is called a modeling problem, where the input as well as the output are a sequence. A typical $Seq2Seq$ model has an encoder and a decoder part, both of them two different neural networks. The encoder network is trained to create a smaller dimensional representation from the input data. This representation is then forwarded to a decoder network which generates a sequence of its own that represents the output.

Both the encoder and decoder were created using two layers of $LSTM$ network with $tanh$ activation function. Parameters like the $number$ of $hidden\ units$, $batch\ size$ or the $learning\ rate$ were optimized during the training process. On top of the decoder $LSTM$ a $dense$ layer was applied.

E. Trading strategy

The trading strategy applies two separate neural networks, one vanilla $LSTM$ for step prediction and a $Seq2Seq\ LSTM$ with two hidden layers to capture the the time series internal structure to be able to predict precise long term future values. Let P_t be actual value of the portfolio, p_t the predicted value, p_{min} and p_{max} the minimum and maximum of the time series predicted by the $Seq2Seq\ LSTM$ neural network. The b_{max} and b_{min} be the upper and lower bounds of the trading range respectively, defined using p_{max} and p_{min} as:

$$b_{max} = p_{max} - R \quad (10)$$

$$b_{min} = p_{min} + R, \quad (11)$$

where

$$R = \alpha(p_{max} - p_{min}) \quad (12)$$

and $\alpha = 0.05$. The single and $Seq2Seq\ LSTM$ networks are applied in a two step trading strategy. In the first line of trigger the value of portfolio hits the appropriate barrier of trading range defined by $Seq2Seq\ LSTM$. In case of cash at hand, there is no event occur when $b_{min} \leq x_t$, but an event is triggered when $b_{min} > x_t$. In case of portfolio at hand no action when $x_t < b_{max}$ and the first trigger is hit when $b_{max} \geq x_t$. The second line of trigger eventually decides about trading event. Single $LSTM$ has very precise prediction for one step ahead and we utilize this property when a trading event is triggered. Single $LSTM$ checks if the predicted portfolio value moves away from trading range, in case of cash at hand $x_t < p_{t+1}$ if so then it waits 1 time step to trade. Similar logic applies when there is a portfolio at hand $p_{t+1} \leq x_t$. The trading range is updated every 5 steps. This is a heuristic number. The lower and upper bounds of the trading range are symmetric to the mid of the predicted long term portfolio prices. The schematic diagram of the trading logic can be seen on figure 1.

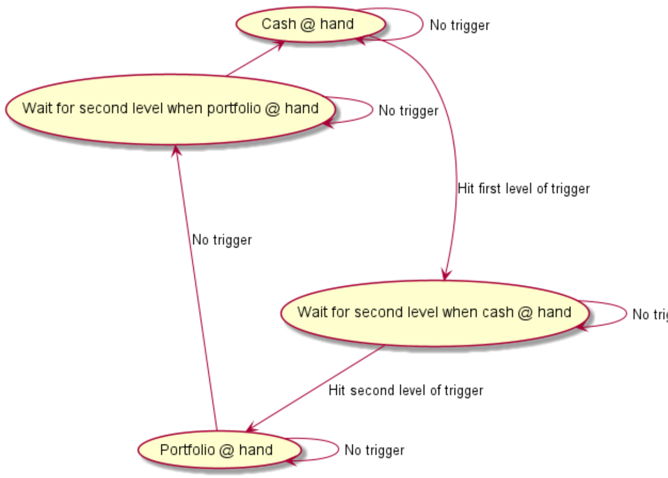

Fig. 1. Schematic diagram of trading logic.

III. RESULTS

In this work the fitting of $VAR(p)$ model was performed c the log data model of the closing values for the full availab $S\&P500$ stocks. The calibration time series length was 23 days and the calibrations were run on a data set between $January\ 1^{st}$ 2015 and $January\ 1^{st}$ 2023.

The MAE values were calculated for 30 and 100 long time series.

TABLE I
MEAN ABSOLUTE ERRORS MAE FOR 30 AND 100 LONG TIME SERIES DATA.

VAR(p)	MAE (30 long)	MAE (100 long)
VAR(1)	0.6686	155.37
VAR(2)	0.0926	0.4053
VAR(3)	0.0805	0.3035
VAR(4)	0.0584	0.1605
VAR(5)	0.0609	0.1780
VAR(6)	0.0586	0.1603
VAR(10)	0.0560	0.1325

From Table I it can be seen that the measure MAE significantly decrease when changing from 1 to 2 and does not change significantly from parameter $p \geq 4$. So the value of the parameter used for portfolio optimization has been set to 4 in the tests.

An example for estimating stock future prices with $VAR(p)$ models is on figure 2.

Figure 3 shows an example for the long term prediction results that is used to determine the trading range.

The network was trained on the same time range as used for $VAR(p)$ calibration. The portfolio was created with the regression matrix and used as an input for the training. The number of $LSTM$ layers is 4, batch size is 1, epochs is 100. We used 3 consecutive data points to predict.

Fig. 2. $VAR(p)$ regressions for $p = 1$ to 5 on $Perrigo\ Company\ PLC$.

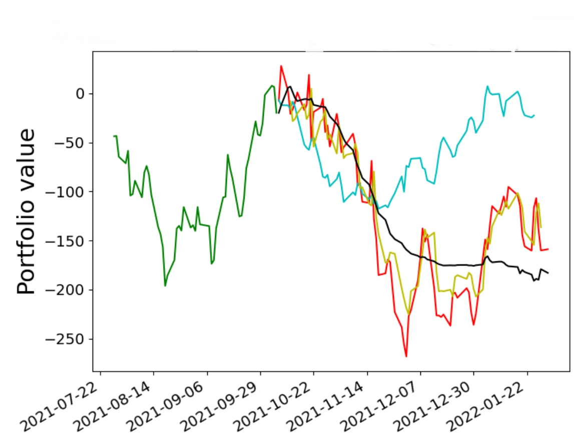

Fig. 3. An example of predicting long term portfolio value. Green line is the input data for the network, red line is the real data to to be estimated and black is the predicted with $Seq2Seq\ LSTM$ network, the yellow line is the single $LSTM$ prediction, the blue is the $VAR(p)$ estimated. The y-axis is in arbitrary units.

The performance testing was started with $\$10k$ investment and the trading interval was 100 days. The number of tests was 500 and the calibration lengths were 180, 230, 280, 330 and 380 days long with different starting points. The tests were run on portfolios selecting constituents from $S\&P500$ pool from a time period from 1^{st} of January 2015 to 31^{st} of December 2022 and a random point was selected between these two dates. The parameter of the $VAR(p)$ was 4, the sparsity was set to 11. The profit after trading time range was calculated either the difference between buying and selling price in case of cash in hand or by subtracting the buying price from actual price when portfolio in hand.

On Figure 4 the result of performance test can be seen.

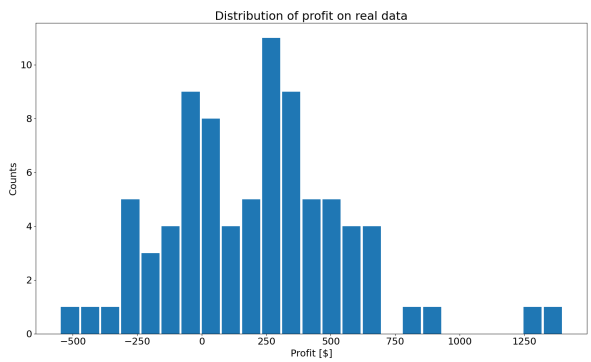

Fig. 4. Profit histogram on portfolio with sparsity of 11.

The mean and the standard deviation of the profit and the mean expected loss is highlighted in Table II.

TABLE II
MEAN, STANDARD DEVIATION AND MEAN EXPECTED ERROR OF THE RETURNS.

Mean	Std Dev	Mean expected loss	Sharpe ratio
$207.82	$351.49	$-181.11	0.59

IV. CONCLUSION AND FUTURE WORKS

From the previous section it can be seen that the overall performance test verified the effectiveness of the trading strategy, however there are some losses associated with this trading strategy. To overcome this, other neural networks such as $Transformer$, GAN, $N-BEATS$, $N-HiTS$, $PatchTST$ or $TimesNet$ would be possible ways and are topics of future researches. As the training and verifying input of the neural network is noisy, this can make learning very difficult. Filtering of the portfolio time series is a possible possible approach to tackle this. Also the logic of trading strategy can be further enhanced in order to reduce losses.

REFERENCES

[1] S. Zaheer, N. Anjum, S. Hussain, A. D. Algarni, J. Iqbal, S. Bourouis and S. S. Ullah A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. *Mathematics*, 64(3):590, 2023.

[2] Zhang, G. Pairs trading with general state space models. *Quantitative Finance*. 21(9):, 1567-1587, 2021

[3] Wu, L., Zang, X. and Zhao, H. Analytic value function for a pairs trading strategy with a Lévy-driven Ornstein-Uhlenbeck process. *Quantitative Finance*. 20(8):, 1285-1306, 2020

[4] Yang, X., Li, H., Zhang, Y. and He, J. Reversion strategy for online portfolio selection with transaction costs. *International Journal of Applied Decision Strategies*. 11(1):, 79-99, 2017

[5] Tah, K. Random walk and structural break in exchange rates. *International Journal of Monetary Economics and Finance*. 11(4):, 384-393, 2018

[6] Narula, I. Stock price randomness of BRICS nations. *International Journal of Public Sector Performance Management*. 4(2):, 231-250, 2018

[7] Banerjee, O., El Ghaoui, L., and d'Aspremont, A. Model selection through sparse maximum likelihood estimation. *J. Mach. Learn. Res.*, 9:, 485–516, 2008.

[8] Jurgen Schmidhube Sepp Hochreiter. Long short-term memorys. *Neural Computation*, 9(8):1735–1780, 1997.

[9] Z. Hu,Y. Zhao and M. Khushi. A Survey of Forex and Stock Price Prediction Using Deep Learning. *Applied System Innovation*, 4(1):9, 2021.

[10] Chuan-Ming Liu, Van-Dai Ta and Direselign Addis Tadesse. Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, 10(8):437, 2020.

[11] Y. Chen, R. Fang, T. Liang, Z. Sha, S. Li, Y. Yi, W. Zhou and H. Song. Stock Price Forecast Based on CNN-BiLSTM-ECA Model. *Scientific Programming*, 2021(5):1–20, 2021.

[12] Aditi Sharan Anita Yadav, C K Jha. Optimizing lstm for time series prediction in indian stock market. *Procedia Computer Science*, 167(3): 2091–2100, 2020.

[13] I. Sutskever, O. Vinyals, Q. V. Le. Sequence to Sequence Learning with Neural Networks. *https://arxiv.org/abs/1409.3215*, 2014.

[14] F. Khalil and G. Pipa Is Deep-Learning and Natural Language Processing Transcending the Financial Forecasting? Investigation Through Lens of News Analytic Process. *Computational Economics*, 60(3):147–171, 2022.

[15] Fogarasi, N. Levendovszky J. Sparse, mean reverting portfolio selection using simulated annealing. *Quantitative Finance*, 11(3):351–364, 2011.

[16] Racz, A., Fogarasi, N. Trading sparse, mean reverting portfolios using VAR(1) and LSTM prediction *Acta Universitatis Sapientiae Informatica* 13(2):, 288–302, 2021

[17] G. E. Box and G. C. Tiao. A canonical analysis of multiple time series. *Biometrika*, 64(2):355, 1977.

[18] A. d'Aspremont. Identifying small mean reverting portfolios. *Quant. Finance*, 11(3):351–364, 2011.

[19] H. Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Springer, 1993.

[20] D. Kingma., J. Ba. Adam: A Method for Stochastic Optimization *https://arxiv.org/abs/1412.6980.*, 2017