Investment Portfolio Optimization Based on Modern Portfolio Theory and Deep Learning Models

Maciej Wysocki and Paweł Sakowski
QFRG and DSLab Monthly Seminar

Quantitative Finance Research Group
Faculty of Economic Sciences, University of Warsaw

10th October 2022

Agenda

- Motivation, hypotheses and research questions
- Literature review
- Methodology
    - Classical Variance-Covariance Estimators
    - Long Short-Term Memory Neural Networks
    - Probabilistic Autoregressive Recurrent Neural Networks
    - Forecasting Variance-Covariance Matrix with Deep Learning Methods
- Portfolio Construction
- Data and Parameters
- Empirical results
- Conclusions and research extensions

Motivation

The main aims:

- setting up a framework for variance-covariance matrix estimation with deep learning models

Motivation

The main aims:

- setting up a framework for variance-covariance matrix estimation with deep learning models
- exploration of deep learning models' capabilities in variance-covariance matrix estimation

Motivation

The main aims:

- setting up a framework for variance-covariance matrix estimation with deep learning models
- exploration of deep learning models' capabilities in variance-covariance matrix estimation
- comparison of classical and deep learning-based variance-covariance matrix estimation techniques

Motivation

The main aims:

- setting up a framework for variance-covariance matrix estimation with deep learning models
- exploration of deep learning models' capabilities in variance-covariance matrix estimation
- comparison of classical and deep learning-based variance-covariance matrix estimation techniques

Reasoning:

- **Probabilistic deep learning models are promising candidates for solution of the large variance-covariance matrix estimation problem**

Motivation

The main aims:

- setting up a framework for variance-covariance matrix estimation with deep learning models
- exploration of deep learning models' capabilities in variance-covariance matrix estimation
- comparison of classical and deep learning-based variance-covariance matrix estimation techniques

Reasoning:

- **Probabilistic deep learning models are promising candidates for solution of the large variance-covariance matrix estimation problem**
- **A comparison of classical and DL approaches to variance-covariance matrix estimation for MPT** was not yet covered for the portfolios of stocks and cryptocurrencies.

Hypotheses and research questions

First Hypothesis:

The strategies utilizing the variance-covariance matrix estimations from the deep learning methods outperform the strategies based on the classical variance-covariance matrix estimation methods.

Second Hypothesis:

The strategies based on the variance-covariance matrix estimations from the probabilistic deep learning models perform better than the strategies based on the simple LSTM-RNN models.

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator
- Xidonas et al. (2020), Henriques and Ortega (2014) - exponentially weighted moving average

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator
- Xidonas et al. (2020), Henriques and Ortega (2014) - exponentially weighted moving average
- Naccarato and Pierini (2014), Zakamulin (2015) - BEKK-GARCH model

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator
- Xidonas et al. (2020), Henriques and Ortega (2014) - exponentially weighted moving average
- Naccarato and Pierini (2014), Zakamulin (2015) - BEKK-GARCH model
- Moura, Santos and Ruiz (2020), Henriques and Ortega (2014), Zakamulin (2015) - dynamic conditional correlation (DCC) by Engle (2002)

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator
- Xidonas et al. (2020), Henriques and Ortega (2014) - exponentially weighted moving average
- Naccarato and Pierini (2014), Zakamulin (2015) - BEKK-GARCH model
- Moura, Santos and Ruiz (2020), Henriques and Ortega (2014), Zakamulin (2015) - dynamic conditional correlation (DCC) by Engle (2002)
- Lam (2020), Frahm and Memmel (2010) - linear and non-linear shrinkage estimators

Literature review

- **H. Markowitz (1952) - Introducrion of Modern Portfolio Theory**
- Georgiev (2014), Xidonas et al. (2020) - frequentist estimator
- Xidonas et al. (2020), Henriques and Ortega (2014) - exponentially weighted moving average
- Naccarato and Pierini (2014), Zakamulin (2015) - BEKK-GARCH model
- Moura, Santos and Ruiz (2020), Henriques and Ortega (2014), Zakamulin (2015) - dynamic conditional correlation (DCC) by Engle (2002)
- Lam (2020), Frahm and Memmel (2010) - linear and non-linear shrinkage estimators
- Fiszeder and Orzeszko (2021) - machine learning approach based on support vector regression

Mean-Variance Optimization

Methodology. Mean-Variance Optimization

- Shorting assets is not allowed and long-only portfolios are considered.

$$
\begin{gathered}
min \sum_{i=1}^{n} \sum_{j=1}^{n} \omega_i R_i \omega_j R_j \sigma_{ij} \\
max \sum_{i=1}^{n} \omega_i \mu_i \\
s.t. \begin{cases} \sum_{i=1}^{n} \omega_i \\ \forall_{i \in \{1,\dots,n\}} \omega_i > 0 \end{cases}
\end{gathered}
\tag{1}
$$

- R is a vector of multivariate returns
- ω_i is share of asset i in the portfolio
- μ is a vector of expected returns
- σ_{ij} is a covariance between assets i and j

Classical Variance-Covariance Matrix Estimators

Methodology. Classical Variance-Covariance Matrix Estimators

Frequentist Estimator

$$\hat{\Sigma}_t = \frac{1}{k-1} \sum_{m=0}^{k} (R_i^{t-m} - \bar{R}_i)(R_j^{t-m} - \bar{R}_j) \tag{2}$$

- R_i^{t-m} denotes the returns of asset i in the period $[t-k, t]$
- $\bar{R}_i$ is the average of returns
- k is the window parameter controlling how many past observations are considered in the calculations

Semi-Covariance Estimator

$$\hat{\Sigma}_t = \frac{1}{k} \sum_{m=0}^{k} min(R_i^{t-m} - B, 0) * min(R_j^{t-m} - B, 0) \tag{3}$$

- B denotes the returns threshold (2% in this study)

Methodology. Classical Variance-Covariance Matrix Estimators

Exponentially Weighted Variance-Covariance Matrix

$$\hat{\Sigma}_t = \lambda\hat{\Sigma}_{t-1} + (1-\lambda)(R_t-\mu)(R_t-\mu)' \quad (4)$$

- λ is a decay rate (set to 0.94)
- μ is a vector of the expected returns

Shrinkage Estimators

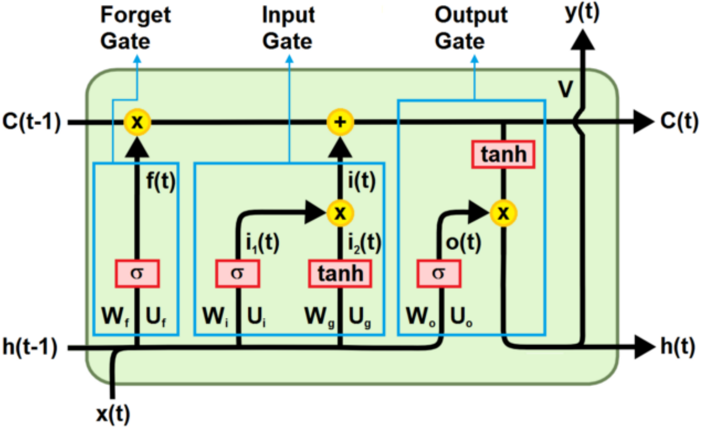
$$\hat{\Sigma}_t = \delta F + (1-\delta)S, 0 \leq \delta \leq 1 \quad (5)$$

- δ is a shrinkage coefficient
- F is a highly structured estimator called the shrinkage target
- S is an unstructured sample variance-covariance estimator

Used variants: Constant Variance Shrinkage, Single Factor Shrinkage, Constant Correlation Shrinkage, Oracle Approximating Estimator.

Long Short-Term Memory Neural Networks

Figure 1. Architecture of a LSTM unit



Source: Image by Marco Del Pra downloaded from: https://towardsdatascience.com/time-series-forecasting-with-deep-learning-and-attention-mechanism-2d001fc871fc

Methodology. Long Short-Term Memory Neural Network II

The LSTM cell architecture is given by the following equations:

$$
\begin{aligned}
f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f) \\
i_{1,t} &= \sigma(W_i h_{t-1} + U_i x_t + b_i) \\
i_{2,t} &= \sigma(W_g h_{t-1} + U_g x_t + b_g) \\
i_t &= i_{1,t} \odot i_{2,t} \\
c_t &= \sigma(f_t c_{t-1} + i_t) \\
o_t &= \sigma(x_t U_o + h_{t-1} W_o + b_o) \\
h_t &= \tanh(c_t) \odot o_t \\
\hat{y}_t &= V h_t
\end{aligned}
\tag{6}
$$

- $W_f, W_i, W_g, W_o, U_f, U_i, U_g, U_o, V$ are appropriate weight matrices of adequately forget gate (f), input gate (i), cell state (g), output gate (o) and output vector
- b_f, b_i, b_g, b_o are biases of each gate
- h_t is the hidden state, x_t is the input to the LSTM unit, c_t is the cell state
- σ is the sigmoid activation function
- symbol $\odot$ denotes the Hadamard product

Probabilistic Autoregressive Recurrent Neural Networks

Methodology. Probabilistic Autoregressive RNNs. DeepVAR

- DeepVAR (Salinas et. al, 2020) is a multivariate probabilistic DL model, which estimates the conditional distribution of time series given their preceding values:

$$P(z_{t:T}|z_{1:t-1}, x_{1:T})$$

Methodology. Probabilistic Autoregressive RNNs. DeepVAR

- DeepVAR (Salinas et. al, 2020) is a multivariate probabilistic DL model, which estimates the conditional distribution of time series given their preceding values:

$$P(z_{t:T}|z_{1:t-1}, x_{1:T})$$

- z is a matrix of time series, $[1, t-1]$ and $[t, T]$ are respectively conditioning and prediction ranges, and x is a matrix of covariates.

- DeepVAR (Salinas et. al, 2020) is a multivariate probabilistic DL model, which estimates the conditional distribution of time series given their preceding values:

$$P(z_{t:T}|z_{1:t-1}, x_{1:T})$$

- z is a matrix of time series, $[1, t-1]$ and $[t, T]$ are respectively conditioning and prediction ranges, and x is a matrix of covariates.
- The model distribution is expressed as a product of likelihood functions $l(z|\theta)$, where θ is vector of parameters, which depends on the outputs of the autoregressive RNN.

Methodology. Probabilistic Autoregressive RNNs. DeepVAR

- DeepVAR (Salinas et. al, 2020) is a multivariate probabilistic DL model, which estimates the conditional distribution of time series given their preceding values:

$$P(z_{t:T} | z_{1:t-1}, x_{1:T})$$

- z is a matrix of time series, $[1, t-1]$ and $[t, T]$ are respectively conditioning and prediction ranges, and x is a matrix of covariates.
- The model distribution is expressed as a product of likelihood functions $l(z|\theta)$, where θ is vector of parameters, which depends on the outputs of the autoregressive RNN.
- DeepVAR is both autoregressive and recurrent, as during the training process each time stamp is estimated using lagged observations and the previous output of the NN as the inputs.

Methodology. Probabilistic Autoregressive RNNs. DeepVAR

- DeepVAR (Salinas et. al, 2020) is a multivariate probabilistic DL model, which estimates the conditional distribution of time series given their preceding values:

$$P(z_{t:T}|z_{1:t-1}, x_{1:T})$$

- z is a matrix of time series, $[1, t-1]$ and $[t, T]$ are respectively conditioning and prediction ranges, and x is a matrix of covariates.
- The model distribution is expressed as a product of likelihood functions $l(z|\theta)$, where θ is vector of parameters, which depends on the outputs of the autoregressive RNN.
- DeepVAR is both autoregressive and recurrent, as during the training process each time stamp is estimated using lagged observations and the previous output of the NN as the inputs.
- We used the Gaussian likelihood function:

$$l(z|\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{z-\mu}{2\sigma^2}} \tag{7}$$

Methodology. Probabilistic Autoregressive RNNs. GPVAR

- GPVAR is a multivariate probabilistic DL model estimating the joint conditional distribution.

Methodology. Probabilistic Autoregressive RNNs. GPVAR

- GPVAR is a multivariate probabilistic DL model estimating the joint conditional distribution.
- The joint distribution is parametrized using a Gaussian copula process, which parameters depend on the model state:

$$\begin{aligned} h_{i,t} &= \phi(h_{i,t-1}, z_{i,t-1}) \\ P(z_t|h_t) &= P\left([f_1(z_{1,t}), f_2(z_{2,t}), ..., f_N(z_{N,t})]^T \,|\mu(h_t), \Sigma(h_t) \right) \end{aligned} \tag{8}$$

- h_t is state of the model with transition dynamic ϕ
- μ and Σ are parameters of the Gaussian distribution
- f_i are functions of form: $\Phi^{-1} \circ \hat{F}_i$ combining the inverse of the standard normal distribution CDF and empirical marginal distribution of i-th input series

Forecasting Variance-Covariance Matrix with Deep Learning Methods

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

In general, any variance-covariance matrix is symmetric and positive - semidefinite:

$$\begin{gathered} \Sigma = \Sigma^T \\ \forall_{x \in \mathbb{R}^n} : x^T \Sigma x \geq 0 \end{gathered} \tag{9}$$

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

In general, any variance-covariance matrix is symmetric and positive - semidefinite:

$$\begin{gathered} \Sigma = \Sigma^T \\ \forall_{x \in \mathbb{R}^n} : x^T \Sigma x \geq 0 \end{gathered} \tag{9}$$

Unfortunately, a variance-covariance matrix created as a combination of forecasts of its entrances is not guaranteed to meet these conditions.

How to assure that the resulting matrix is **symmetric** and **positive - semidefinite**?

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

In general, any variance-covariance matrix is symmetric and positive - semidefinite:

$$\begin{gathered} \Sigma = \Sigma^T \\ \forall_{x \in \mathbb{R}^n} : x^T \Sigma x \geq 0 \end{gathered} \tag{9}$$

Unfortunately, a variance-covariance matrix created as a combination of forecasts of its entrances is not guaranteed to meet these conditions.

How to assure that the resulting matrix is **symmetric** and **positive - semidefinite**?

We used **Cholesky decomposition**.

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

Deep learning-based variance-covariance matrix forecasting methodology

1. For each available timestamp calculate the historical $N \times N$ variance-covariance matrix Σ_t over the selected window w (where N is the number of assets).

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

Deep learning-based variance-covariance matrix forecasting methodology

1. For each available timestamp calculate the historical $N \times N$ variance-covariance matrix Σ_t over the selected window w (where N is the number of assets).
2. Apply Cholesky decomposition to each of the obtained variance-covariance matrices $\Sigma_t = X_t X_t^*$.

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

Deep learning-based variance-covariance matrix forecasting methodology

1. For each available timestamp calculate the historical $N \times N$ variance-covariance matrix Σ_t over the selected window w (where N is the number of assets).
2. Apply Cholesky decomposition to each of the obtained variance-covariance matrices $\Sigma_t = X_t X_t^*$.
3. Construct separate time series $x_t^{i,j}$ for each of the entrances of the decomposed matrices resulting in $\frac{N(N+1)}{2}$ series.

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

Deep learning-based variance-covariance matrix forecasting methodology

1. For each available timestamp calculate the historical $N \times N$ variance-covariance matrix Σ_t over the selected window w (where N is the number of assets).
2. Apply Cholesky decomposition to each of the obtained variance-covariance matrices $\Sigma_t = X_t X_t^*$.
3. Construct separate time series $x_t^{i,j}$ for each of the entrances of the decomposed matrices resulting in $\frac{N(N+1)}{2}$ series.
4. Forecast the obtained series of Cholesky factors' entrances using a selected deep learning method trained on the available observations.

Methodology. Forecasting Variance-Covariance Matrix with Deep Learning Methods

Deep learning-based variance-covariance matrix forecasting methodology

1. For each available timestamp calculate the historical $N \times N$ variance-covariance matrix Σ_t over the selected window w (where N is the number of assets).
2. Apply Cholesky decomposition to each of the obtained variance-covariance matrices $\Sigma_t = X_t X_t^*$.
3. Construct separate time series $x_t^{i,j}$ for each of the entrances of the decomposed matrices resulting in $\frac{N(N+1)}{2}$ series.
4. Forecast the obtained series of Cholesky factors' entrances using a selected deep learning method trained on the available observations.
5. Construct the Cholesky factors X_{T+n} from the forecasted series and then reconstruct the variance-covariance matrix $\Sigma_{T+n} = X_{T+n} X_{T+n}^*$.

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.
- The portfolio optimization steps were done on every rebalancing day (every 30, 60, 90 and 120 days).

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.
- The portfolio optimization steps were done on every rebalancing day (every 30, 60, 90 and 120 days).
- Transactional costs were considered during the optimization process.

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.
- The portfolio optimization steps were done on every rebalancing day (every 30, 60, 90 and 120 days).
- Transactional costs were considered during the optimization process.

A single portfolio optimization process:

1. Gather the available prices of assets and calculate the expected returns estimated as the mean historical returns.

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.
- The portfolio optimization steps were done on every rebalancing day (every 30, 60, 90 and 120 days).
- Transactional costs were considered during the optimization process.

A single portfolio optimization process:

1. Gather the available prices of assets and calculate the expected returns estimated as the mean historical returns.
2. Estimate the variance-covariance matrix using the selected method and optimize weights.

Portfolio Construction

- All portfolios were optimized using the minimum variance criterion.
- The portfolio optimization steps were done on every rebalancing day (every 30, 60, 90 and 120 days).
- Transactional costs were considered during the optimization process.

A single portfolio optimization process:

1. Gather the available prices of assets and calculate the expected returns estimated as the mean historical returns.
2. Estimate the variance-covariance matrix using the selected method and optimize weights.
3. Calculate discrete portfolio allocation from the optimal weights, allocate the available capital to update the portfolio structure by buying and selling appropriate assets.

Data

- Daily close prices of cryptocurrencies and US stocks.

Data

- Daily close prices of cryptocurrencies and US stocks.
- Missing (weekends) stock quotes for these days were filled with the last available price.

Data

- Daily close prices of cryptocurrencies and US stocks.
- Missing (weekends) stock quotes for these days were filled with the last available price.
- Around 1000 stocks listed on the New York Stock Exchange and around 500 cryptocurrencies with the highest market capitalization.

Data

- Daily close prices of cryptocurrencies and US stocks.
- Missing (weekends) stock quotes for these days were filled with the last available price.
- Around 1000 stocks listed on the New York Stock Exchange and around 500 cryptocurrencies with the highest market capitalization.
- Filtered data to select 10 stocks and 10 cryptocurrencies with the highest market capitalization at a given date were selected and passed to the portfolio optimization.

Data

- Daily close prices of cryptocurrencies and US stocks.
- Missing (weekends) stock quotes for these days were filled with the last available price.
- Around 1000 stocks listed on the New York Stock Exchange and around 500 cryptocurrencies with the highest market capitalization.
- Filtered data to select 10 stocks and 10 cryptocurrencies with the highest market capitalization at a given date were selected and passed to the portfolio optimization.
- Over the whole study horizon, this method selected 82 unique assets, including 64 cryptocurrencies and 18 stocks.

Parameters

Parameters Shared by All Strategies

Parameter	Options or Range
Window	30, 60, 90, 120
Rebalancing Period	30, 60, 90, 120
Optimization Criterion	minimal variance

Strategies Utilizing LSTM Forecasts

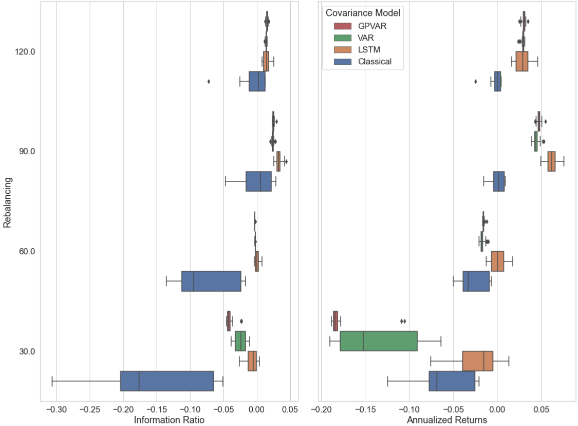
Parameter	Options or Range
Units	5, 10, 15, 20, [5, 5], [5, 10], [10, 5], [10, 10], [15, 15], [20, 20]
Batch Size	8, 16
Sequences Length	15, 20

Strategies Utilizing Probailistic Deep Learning Forecasts

Parameter	Options or Range
Units	5, 10, 15, 20
Scaling	True, False
Low-Rank	True, False
Copula	True, False

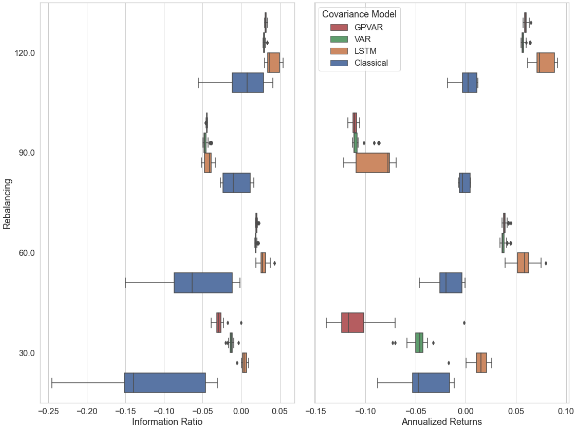
Results. 30-days window

Figure 2. Information ratio and annualized returns of the investment strategies based on a 30-days window.



Note: Performance statistics were aggregated across all parameters for each variance-covariance estimation method and rebalancing period. Number of strategies in each rebalancing periods respectively was: GPVAR - 23, 32, 32, 31; VAR - 25, 32, 32, 30; LSTM - 25, 40, 39, 40; classical - 8, 8, 8, 8.

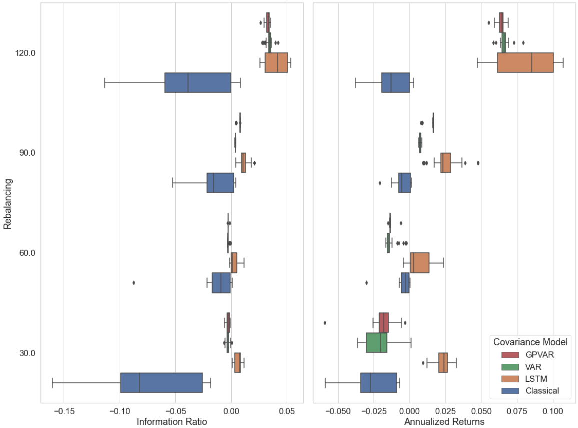
Figure 3. Information ratio and annualized returns of the investment strategies based on a 60-days window



Note: Performance statistics were aggregated across all parameters for each variance-covariance estimation method and rebalancing period. Number of strategies in each rebalancing periods respectively was: GPVAR - 32, 32, 32, 32; VAR - 32, 32, 32, 32; LSTM - 38, 40, 40, 40; classical - 8, 8, 8, 8.

Results. 90-days window

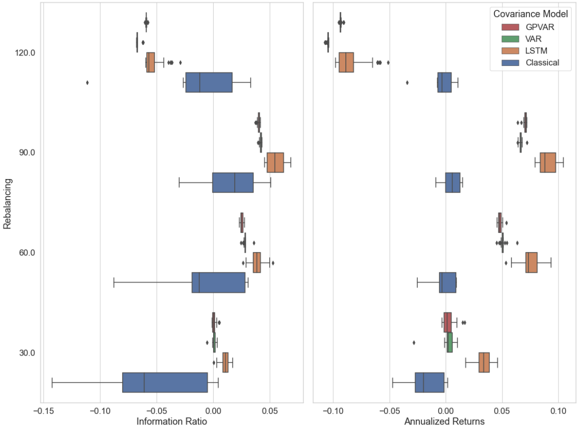
Figure 4. Information ratio and annualized returns of the investment strategies based on a 90-days window.



Note: Performance statistics were aggregated across all parameters for each variance-covariance estimation method and rebalancing period. Number of strategies in each rebalancing periods respectively was: GPVAR - 32, 32, 32, 32; VAR - 32, 32, 32, 32; LSTM - 30, 40, 40, 24; classical - 8, 8, 8, 8.

Results. 120-days window

Figure 5. Information ratio and annualized returns of the investment strategies based on a 120-days window.



Note: Performance statistics were aggregated across all parameters for each variance-covariance estimation method and rebalancing period. Number of strategies in each rebalancing periods respectively was: GPVAR - 32, 32, 32, 32; VAR - 29, 32, 32, 32; LSTM - 40, 40, 40, 34; classical - 8, 8, 8, 8.

Research Hypotheses Verification

- The first research hypothesis concerning the performance of the deep learning-based strategies was partially rejected, as the performance of such strategies strongly depended on window and rebalancing parameters.

Research Hypotheses Verification

- The first research hypothesis concerning the performance of the deep learning-based strategies was partially rejected, as the performance of such strategies strongly depended on window and rebalancing parameters.
- Nevertheless, in most cases the strategies utilizing the variance - covariance matrices from the deep learning models were significantly better than the strategies exploiting the classical variance - covariance estimation methods.

Research Hypotheses Verification

- The first research hypothesis concerning the performance of the deep learning-based strategies was partially rejected, as the performance of such strategies strongly depended on window and rebalancing parameters.
- Nevertheless, in most cases the strategies utilizing the variance - covariance matrices from the deep learning models were significantly better than the strategies exploiting the classical variance - covariance estimation methods.

- The second research hypothesis of this study was rejected, as the strategies employing the probabilistic deep learning models did not perform any better than the strategies with variance-covariance matrix estimation from the LSTM-RNN models.

Conclusions

- Based on our framework we were produced strategies that provided positive returns and were profitable over the backtests.
- **Performance of the strategies strongly dependeds on length of observation window and frequency of rebalancing**.
- **A higher number of observations used in the variance-covariance matrix estimation translated into better results**, especially in case of deep learning-based strategies.
- **A less frequent portfolio re-optimizations** generally **performed better**, hence this framework could be utilized for a long-term portfolio management.
- In most of the considered combinations of parameters, strategies based on matrices forecasted with LSTM-RNN outperformed the others in terms of the examined performance metrics.
- **Although DeepVAR and GPVAR typically achieved slightly worse results, both models were very stable across their hyperparameters**, especially for longer observation windows.
- **The probabilistic models tend to be more robust to hyperparameters changes** and they could provide good results without a lengthy optimization process.

Research extensions

- Use larger portfolios with **longer history**.
- Compare the deep learning-based approach with the **dynamic financial econometrics models** such as the multivariate GARCH.
- Try other optimization criterion for the Markowitz framework such as the Sharpe ratio optimization or maximization of returns for a given risk threshold.

Thank you for your attention!

Maciej Wysocki, m.wysocki9@uw.edu.pl

Faculty of Economic Sciences,

University of Warsaw,

Department of Quantitative Finance

Quantitative Finance Research Group

and

Paweł Sakowski, sakowski@wne.uw.edu.pl,

Faculty of Economic Sciences,

University of Warsaw,

Department of Quantitative Finance

Quantitative Finance Research Group

References I

- Fiszeder, P., & Orzeszko, W. Covariance matrix forecasting using support vector regression. Applied Intelligence, 2021, pp. 1-14
- Frahm, G., Memmel, C. Dominating estimators for minimum-variance portfolios. Journal of Econometrics, 2010, 159(2), pp. 289-302
- Georgiev, B. Constrained mean-variance portfolio optimization with alternative return estimation. Atlantic Economic Journal, 2014, 42(1), pp. 91-107
- Henriques, J., Ortega, J. P. Construction, management, and performance of sparse Markowitz portfolios. Studies in Nonlinear Dynamics & Econometrics, 2014, 18(4), pp. 383-402
- Markowitz H. Portfolio Selection. Journal of Finance (Wiley-Blackwell), 1952, 7(1), pp. 77–91
- Moura, G. V., Santos, A. A., Ruiz, E. Comparing high-dimensional conditional covariance matrices: Implications for portfolio selection. Journal of Banking & Finance, 2020, 118
- Naccarato, A., Pierini, A. BEKK element-by-element estimation of a volatility matrix. A portfolio simulation. In Mathematical and statistical methods for actuarial sciences and finance, 2014, pp. 145-148
- Xidonas, P., Tsionas, M., Zopounidis, C. On mutual funds-of-ETFs asset allocation with rebalancing: sample covariance versus EWMA and GARCH. Annals of Operations Research, 2020, 284(1), pp. 469-482
- Zakamulin, V. A test of covariance-matrix forecasting methods. The Journal of Portfolio Management, 2015, 41(3), pp. 97-108