Matrix Evolutions: Synthetic Correlations and Explainable Machine Learning for Constructing Robust Investment Portfolios

Jochen Papenbrock, Peter Schwendner, Markus Jaeger, and Stephan Krügel

Jochen Papenbrock
is the CEO of Firamis GmbH in Oberursel, Germany.
jp@firamis.de

Peter Schwendner
is the director of the Institute of Wealth & Asset Management at the Zurich University of Applied Sciences in Winterthur, Switzerland.
scwp@zhaw.ch

Markus Jaeger
is a senior investment expert at Munich Re Markets in Munich, Germany.
majaeger@munichre.com

Stephan Krügel
is a senior investment expert at Munich Re Markets in Munich, Germany.
skruegel@munichre.com

KEY FINDINGS

- The authors introduce the matrix evolutions concept based on an evolutionary algorithm to simulate correlation matrixes useful for financial market applications.
- They apply the resulting synthetic correlation matrixes to benchmark hierarchical risk parity (HRP) and equal risk contribution allocations of a multi-asset futures portfolio and find HRP to show lower portfolio risk.
- The authors evaluate three competing machine learning methods to regress the portfolio risk spread between both allocation methods against statistical features of the synthetic correlation matrixes and then discuss the local and global feature importance using the SHAP framework by Lundberg and Lee (2017).

ABSTRACT

In this article, the authors present a novel and highly flexible concept to simulate correlation matrixes of financial markets. It produces realistic outcomes regarding stylized facts of empirical correlation matrixes and requires no asset return input data. The matrix generation is based on a multiobjective evolutionary algorithm, so the authors call the approach *matrix evolutions*. It is suitable for parallel implementation and can be accelerated by graphics processing units and quantum-inspired algorithms. The approach is useful for backtesting, pricing, and hedging correlation-dependent investment strategies and financial products. Its potential is demonstrated in a machine learning case study for robust portfolio construction in a multi-asset universe: An explainable machine learning program links the synthetic matrixes to the portfolio volatility spread of hierarchical risk parity versus equal risk contribution.

TOPICS

Statistical methods, big data/machine learning, portfolio construction, performance measurement*

Markets in crisis mode are an example of how assets correlate or diversify in times of stress. It is essential to see how markets, asset classes, and factors change their correlation and diversification properties in different market regimes. The most recent crisis now adds to the list of scenarios available

*All articles are now categorized by topics and subtopics. **View at PM-Research.com**.

to risk managers, but it is still unknown what future crises will look like. Every crisis looks different because history does not repeat itself. That is why simple backtests are problematic, being a view in the rear mirror only.

Therefore, it is desirable not only to consider real manifestations of market scenarios from history but to simulate new, realistic scenarios systematically. To model the real world, quants turn to synthetic data, building artificially generated data based on so-called market generators. Market simulation of correlation matrixes unveils a new and flexible way of modeling financial time series, which has recently inspired a surge of research activity in the quantitative finance community.

Investment strategies could be systematically tested based on these simulations like a new car is systematically tested in a wind tunnel before hitting the road. In such environments, key system parameters are configurable. An investment strategy is said to be robust if it is not fragile in some situations. We could make specific assumptions regarding future scenarios, but if any of these assumptions do not hold, it will adversely affect the strategy.

There are many practical use cases and applications for simulating correlation matrixes, especially in correlation-sensitive products such as multi-asset derivatives. Other applications include creating reference datasets without licensing restrictions (e.g., for education and academia) and conducting standardized testing of model performance or recent approaches to crowd sourced model development.

Matrix simulation can also be used in risk management, such as in creating scenarios or new ways of (stress) testing portfolios, giving rise to a shift in the asset management industry toward a more mature, industrialized, digitized, systematic, and scientific way of constructing investment portfolios using the support of artificial intelligence (AI). Future regulatory requirements could be met, and sound risk management practices could be implemented. Examples are packaged retail investment and insurance products performance scenarios and European Securities and Markets Authority stress testing.

Quant funds use Monte Carlo methods with parameters for asset processes and regime shifts as well as AI to master correlations. Lopez de Prado (2019) has described use of nonparametric, AI-based Monte Carlo methods when data-generating processes are too complex to model explicitly. Examples are neural network–based approaches such as variational autoencoders (VAEs) and generative adversarial networks (GANs).

Along these lines, Marti (2019) recently proposed a promising approach called CorrGAN, which uses GANs to sample plausible financial correlation matrixes. The authors list a battery of matrix evaluations of financial correlation matrixes that is encompassed by their approach. An online test system at http://www.corrgan.io/ has also been set up in which a user can declare whether certain correlation matrixes are fake or real. At the time of writing this article, the CorrGAN test had a chance of 50%, meaning people could not decide whether a matrix was fake or real.

Approaches like CorrGAN depend on the input data and the parameterization/training of the machine learning (ML) approach. Thus, it needs to be assumed to approximate the underlying data-generating process (Lopez de Prado 2019). A novel method is to use evolutionary algorithms to generate realistic matrixes. Doing so requires almost no assumptions and no training data. We outline the matrix evolutions approach in the following section before we continue with a practical example of investment portfolio construction.

MATRIX EVALUATIONS OF FINANCIAL CORRELATION MATRIXES

At this point, a question arises: What is a realistic matrix in this context? Matrixes obtained from financial datasets tend to exhibit a very specific structure.

A straightforward approach is to consider the matrix evaluations of empirical correlation matrixes. Marti (2019) described them as follows and referred to classical literature on network complexity and hierarchy in financial markets:

- A distribution of pairwise correlations that is significantly shifted to the positive
- Eigenvalues that follow the Marchenko-Pastur distribution but for
  - a very large first eigenvalue (the market)
  - a couple of other large eigenvalues (industries)
- The Perron–Frobenius property (first eigenvector has positive entries)
- A hierarchical structure of correlations
- The scale-free property of the corresponding minimum spanning tree (MST)

The CorrGAN approach seems to meet all of these matrix evaluation criteria, thus producing very realistic matrixes. Such an approach can be used to sample many correlated asset paths, providing a large number of realistic scenarios that have never been observed before. The output matrixes need to be positive definite, of course (e.g., for applications involving Cholesky factorization).

The literature on matrix evaluations of empirical asset returns is much larger than that on asset correlations. Even smaller is the number of approaches to simulate such realistic correlation matrixes, maybe because finding such correlation matrixes is highly complex. For example, Huettner, Mai, and Mineo (2018) pointed out that "to the best of our knowledge, to date there exists no simulation algorithm that can reproduce all of them [the matrix evaluations], or even more than just one." The same authors later came up with a solution (Huettner and Mai 2019). They generated correlation matrixes with the Perron–Frobenius property based on a given eigenvalue structure. The authors chose eigenvalues distributed according to a power law, and a significant percentage of the simulated correlation matrixes then exhibited a realistic distribution of pairwise correlations in addition to realistically distributed eigenvalues. Furthermore, when additionally fixing the largest eigenvalue at a realistic value of 40% of the total variance, large correlation matrixes simulated from their algorithm tended to exhibit a power-law-like degree distribution in their corresponding MST.

Other metrics to consider in empirical correlations are the ranges; the mean shifted to the positive, smooth, and unimodal (one-peak) distributions of correlations; and largest eigenvalues not larger than a certain amount. The approach of Huettner and Mai (2019) seems to meet most requirements. However, the hierarchical properties are discussed less intensively than the MST properties.

Hierarchies in Financial Markets

Hierarchy is an important concept in financial markets, as was highlighted by Mantegna (1999) and Marti et al. (2017). In 1962, Nobel Laureate Simon wrote: "The central theme that runs through my remarks is that complexity frequently takes the form of hierarchy, and that hierarchic systems have some common properties that are independent of their specific content. Hierarchy, I shall argue, is one of the central structural schemes that the architect of complexity uses."

Financial markets are indeed complex systems with hierarchies owing to their emergent, self-organizing properties. For this reason, in recent years several approaches for portfolio construction have been developed that explicitly take into account hierarchies in financial markets (e.g., Onnela et al. 2003; Tola et al. 2008; Papenbrock 2011; Baitinger and Papenbrock 2015, 2017; Papenbrock and Schwendner 2015).

Why is taking these hierarchies into account so important? Lopez de Prado (2016b) gave two reasons for the instability of traditional inversion-based portfolio construction such as equal risk contribution (ERC): instability caused by noise and instability caused by signals. Inversion-based algorithms often replace very different assets when small changes in the estimation parameters occur. A natural way to replace assets, however, would be in the same correlation cluster. An optimization-free, heuristic algorithm called hierarchical risk parity (HRP) has been proposed by Lopez de Prado (2016a). It has been shown that such portfolios can outperform traditional approaches out of sample.

Explainable Machine Learning

The most common task of ML is to train a model that can predict an unknown outcome (response variable) based on a set of known input variables/features. When using such models for real-life applications, it is often crucial to understand why a particular set of features leads to precisely that prediction. Many ML models exhibit a so-called *explainability gap*: The more accurate they are, the more they are like a black box whose decision making cannot be explained. However, we want to trust the model and need simple, interpretable explanations, or at least we need to know the essential features involved in a market phenomenon and how the features interrelate.

There has recently been increased activity in developing explainable AI (XAI) or interpretable ML approaches in many industries, especially in financial services where supervisors have started to focus on AI governance and risk management (see an example of ML [XML] in credit risk management from Bussmann et al. 2021). Many use the concept of Shapley (1953) values, the only prediction explanation framework with a solid theoretical foundation (Lundberg and Lee 2017), being rooted in cooperative game theory.

Unless the actual distribution of the features is known and there are fewer than, say, 10–15 features, these Shapley values need to be estimated/approximated. A unified approach to interpreting model predictions is the SHAP (Lundberg and Lee 2017). There is also tree SHAP (Lundberg et al. 2020), which is an algorithm to compute exact SHAP values for decision tree–based models such as XGBoost (Chen and Guestrin 2016). These are among the most popular and successful ML algorithms in practice. New tools allow us to explain the predictions and gain insight into the global behavior of these models.

Jaeger et al. (2020) employed block-bootstrap multi-asset market return series to analyze whether HRP or ERC outperforms. The authors concluded that HRP is more stable than ERC, expressed in a closer matching of risk control parameters and in lower drawdowns. The authors also used a novel XML setup to attribute important variables to the success of different investment strategies. Their approach opens the possibility to challenge heuristic strategies and to study their relationship with the properties of their asset universe that otherwise would be hidden under very nonlinear relationships or complex statistical dependencies. They used Shapley values to determine the variables that were significant in explaining the difference in performance. This approach can be helpful in comparisons and facilitate attribution analysis, factor interaction, and assigning importance.

As mentioned before, the study by Jaeger et al. (2020) is based on bootstrapped data and thus is dependent on the underlying empirical data and the bootstrap procedure. It would be desirable to change this analysis to simulate data to be able to generate more potential scenarios and to control better the input space for the XML. In this article, we do precisely that: We augment the sample space by using synthetic correlation matrixes with real properties, which can improve the robustness of the

ML approach. However, choices of simulation methodology and matrix evaluations addressed will be decisive for the performance of the ML.

In this article, for each correlation matrix, we carry out a Monte Carlo simulation and measure the average spread between the risk of ERC and HRP; risk is measured as volatility. We then set up an ML regression in which we have the matrix evaluations of each correlation matrix as input and regress on the spread.

After finishing the ML, we extract the variable importance for each data point with an XML approach based on cooperative game theory (Shapley values; see Jaeger et al. 2020). In this way, we are able to see which matrix evaluations (and combinations thereof) drive the risk spread between ERC and HRP. To summarize, this analytical workflow is quite similar to the one by Jaeger et al. (2020), but in this article, we use synthetic matrixes generated by matrix evolutions and not block-bootstrapped data.

Multiobjective Evolutionary Algorithm

For our XML approach, we need a flexible and controlled environment to generate realistic correlation matrixes in which all matrix evaluations are addressed at the same time. We therefore introduce a novel methodology for generating realistic correlation matrixes using a multiobjective evolutionary algorithm based on decomposition (MOEA/D). We call this approach *matrix evolutions*. It is model-free and does not require training data. It can be formulated to produce matrixes with desired properties (e.g., those typical for a stress scenario or crisis).

Matrix evolutions decomposes a multiobjective optimization problem (MOP) into several scalar optimization subproblems and optimizes them simultaneously. MOEA/D were initially proposed by Zhang and Li (2007) and represent a widely used class of population-based metaheuristics for solving MOPs, as further discussed by Trivedi et al. (2017). MOEA/Ds can generate a set of very evenly distributed solutions.

MOPs are problems in which multiple objective functions must be optimized simultaneously. These problems are characterized by a set of objective functions, which results in the existence of a set of optimal compromise (Pareto-optimal) solutions instead of a single globally optimal one. This set is called the *Pareto frontier*, well known from the efficient line in modern portfolio theory.

We set up our MOP such that each of the matrix evaluations is a single objective. The objectives can even be contrary and conflicting. We search for a set of best compromise solutions. The new method can iteratively optimize synthetic correlation data based on a set of utility parameters until its difference from the original data achieves the desired level.

This controlled and flexible way of generating synthetic but realistic correlation matrixes systematically augments the sample space for ML. Usually, learner performance is evaluated in terms of accuracy, interpretability, and efficiency, among other factors. The limitation of this approach is that we never know whether the maximum achievable accuracy is reached or if there is still potential for performance improvements. This can be addressed by synthetic data to generate sufficient prior knowledge of the solution space.

Using synthetic data in this way is not new, but to the best of our knowledge, we are the first to use it for generating financial correlation matrixes. Our matrix evolutions is used in an XML approach to understand the circumstances under which specific portfolio construction algorithms work.

To summarize, this article follows an approach that could be called *triple ML*:

1. We use ML to generate synthetic correlations (evolutionary algorithms are sometimes deemed AI).

2. We test where HRP outperforms (HRP uses unsupervised representation learning like hierarchical clustering).
3. We use XML to explain the decision making of the ML.

Potential for Acceleration

The entire workflow can be accelerated considerably because there are many parallel[1] steps involved. First, the multiobjective evolutionary algorithm can be accelerated by graphics processing units (GPUs), as described by Souza and Pozo (2014) and Oliveira, Davendra, and Guimarães (2013), some of them based on CUDA technology. There also exists quantum-inspired MOEA/D, as, for example, outlined by Wang, Li, and Jiao (2016).

The Monte Carlo simulations can, of course, be parallelized and processed with multiple-core central processing units (CPUs) or one or more GPUs. Last, SHAP can be computed in an accelerated way following the approach of Mitchell, Frank, and Holmes (2020) for tree-based ML models such as XGBoost, which in turn can be accelerated by multiple CPUs and GPUs, as we do in this work.

Case Study for a Multi-Asset Portfolio

The purpose of this case study is to determine whether we should pick HRP or ERC as a portfolio construction approach for a special dataset. We would like to develop an ML program with synthetic learning data (matrix evaluations of simulated correlation matrixes) that can predict the risk spread. This prediction should also be explainable in terms of variable contributions.

Our backtest simulation study is very close to the one by Lopez de Prado (2016a), in which correlated returns are sampled, and several portfolio construction approaches are compared in a walk-forward test with rebalancing to see which strategy exhibits the lowest portfolio risk. The comparison is made with respect to the realized volatility of the strategy, and we focus on HRP and ERC.

Exhibit 1 shows our investment universe of futures covering commodities, equities, and fixed income.

The choice of the universe is a classical one for constructing diversified multi-asset portfolios. We use daily index returns and concentrate on the daily data from May 2, 2000, to October 7, 2019. We construct walk-forward tests with monthly rebalancing and one-year rolling windows for parameter estimation. We construct the two strategies, HRP and ERC. The strategies are unleveraged and long only, and transaction costs are not considered.

In our simulation, the HRP portfolio allocation strategy shows an annualized volatility of 0.0400, whereas the ERC strategy exhibits a higher risk of 0.0424. The risk spread is defined as Portfolio volatility (ERC)—Portfolio volatility (HRP), and it is positive in this case.

We also compute the simple Pearson empirical correlation matrix across the entire period and measure the following four properties (Exhibit 2):

- avg_corr: The average correlation coefficient of the matrix.
- eigen_gini: The Gini coefficient of the eigenvalues (Gini ranges from 0 to 1, where 0 means complete equality).
- coph_corr_single: This measures how close the single linkage hierarchical clustering is compared to the original correlation distance matrix. It is a proxy for how hierarchical the data are. Correlation distance is measured as $D = \sqrt{2(1 - CORR)}$.

[1] See Herlihy and Shavit (2012): "Some computational problems are 'embarrassingly parallel': they can easily be divided into components that can be executed concurrently."

EXHIBIT 1

Investment Universe

Ticker	Asset Class	Currency	Long Name
CLA Comdty	Commodities	USD	NYMEX WTI Light Sweet Crude Oil
GCA Comdty	Commodities	USD	COMEX Gold
SIA Comdty	Commodities	USD	COMEX Silver
BZA Index	Equities	BRL	BM&F IBOVERSPA
ESA Index	Equities	USD	CME E-mini S&P 500
HIA Index	Equities	HKD	HKFE Hang Seng
NKA Index	Equities	JPY	OSE Nikkei 225
NQA Index	Equities	USD	CME E-mini NASDAQ-100
SMA Index	Equities	CHF	Eurex SMI
VGA Index	Equities	EUR	Eurex EURO STOXX 50
XPA Index	Equities	AUD	ASX SPI 200
Z A Index	Equities	GBP	ICE FTSE 100
CNA Comdty	Fixed Income	CAD	10Y Canadian GB
G A Comdty	Fixed Income	GBP	ICE Long Gilt
RXA Comdty	Fixed Income	EUR	Eurex 10Y Euro-Bund
TYA Comdty	Fixed Income	USD	CBOT 10Y US T-Note
XMA Comdty	Fixed Income	AUD	ASX 10Y Australian T-Bonds

- perron_frob_sum_neg: This measures the sum of negative entries of the first eigenvector.

EXHIBIT 2

Multivariate Measures

	empcorr_orig
avg_corr	0.110
eigen_gini	0.578
coph_corr_single	0.948
perron_frob_sum_neg	1.264

Exhibit 2 shows that the average correlation in the universe (avg_corr) is slightly positive, the Gini of eigenvalues is 0.578, the cophenetic correlation between the original correlation distance matrix and the cophenetic matrix of the hierarchical clustering (single linkage) is 0.948, and the sum of negative entries of the first eigenvector is 1.264.

The matrix is visualized in Exhibit 3. The block structure of asset classes is clearly observable.

Matrix Evolutions

In the matrix evolution step, we create realistic correlation matrixes with the desired properties as in the intended matrix evaluations. We sample synthetic correlation matrixes that are in the neighborhood of the empirical matrix. In this way, the ML gets information about more or less similar scenarios around the empirical matrix and can thus learn the link between matrix evaluations and risk spread. This technique is used to analyze the circumstances (matrix constellations) under which one method would be preferred over the other. If there are opinions about future matrix evaluations, the ML could give answers about which method to pick. It would also show how robust the decision for a specific strategy is and whether a small change in the estimations would change the decision, thus reflecting the robustness of the decision making.

We define upper and lower barriers for the matrix evaluations inside which the matrixes should be sampled. This neighborhood could be defined in a static way or by expert knowledge. We pick another approach by choosing several alternative measurement approaches to correlation matrixes, thus creating neighborhoods. We take

EXHIBIT 3

Heatmap of the Empirical Correlation Matrix

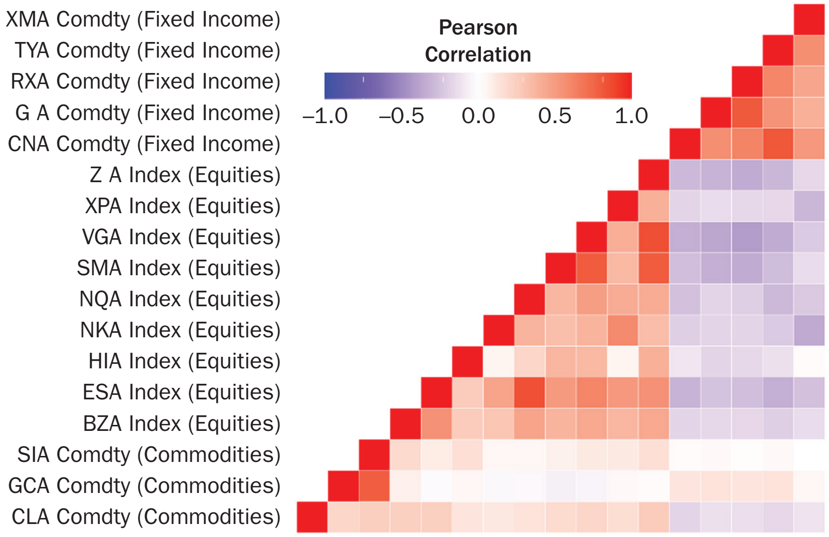


EXHIBIT 4

Multivariate Measures

	Lower	empcorr_orig	Upper
avg_corr	0.106	0.110	0.160
eigen_gini	0.536	0.578	0.711
coph_corr_single	0.928	0.948	0.948
perron_frob_sum_neg	1.217	1.264	1.411

the most extreme values of the matrix evaluations as barriers. These alternative estimators for correlation matrixes are random matrix denoising, shrinkage, exponential weighting, and some related ones. CorrGAN could also be useful for such neighborhood sampling/exploration around a given point in case the GANs or VAEs have a smooth latent space.

We sample 10,480 correlation matrixes by our simpler procedure. We then run the MOP with the following four objective functions:

1. Minimize the deviation of avg_corr outside the boundaries.
2. Minimize the deviation of eigen_gini outside the boundaries.
3. Minimize the deviation of coph_corr_single outside the boundaries.
4. Minimize the deviation of perron_frob_sum_neg outside the boundaries.

An important constraint to the resulting matrix is to be positive definite (i.e., to have only positive eigenvalues). This constraint is a necessary condition for Cholesky factorization. In practical applications of Monte Carlo simulation for value-at-risk or pricing models, issues with gaps in time-series data often lead to nonpositive eigenvalues for empirical correlation matrixes and require involved regularization procedures.

Exhibit 4 shows the minima and maxima of the quantities resulting from the sampled correlation matrixes. It can be seen that the matrix evaluations of the empirical matrix (middle column) are almost always covered by sample data.

We run the MOP to produce 10,480 sample matrixes, involving several initial seed populations to get a range of different results. In some situations, the empirical quantities are the most extreme variables, but in most cases, they are well surrounded by samples. These situations could be improved by running another MOP in which boundaries and constraints are set in a different way.

In the next step, we create the training dataset based on the generated matrixes. First, we evaluate the matrix with respect to the measures that had already been used in the MOP. We enrich these measures by two more quantities:

EXHIBIT 5

Range of Multivariate Measures

	Min	empcorr	Max
avg_corr	0.105	0.110	0.145
eigen_gini	0.536	0.578	0.614
coph_corr_ward	0.615	0.862	0.904
coph_corr_single	0.927	0.948	0.948
perron_from_sum_neg	1.217	1.264	1.412
power_eigen_values	1.284	2.071	5.540

- The power law exponent of the eigenvalue distribution
- Ward hierarchical clustering

The first was approximated to a certain extent in the MOP by the eigen_gini. Ward hierarchical clustering delivers a different view on the data aside from the single-linkage hierarchical clustering that was already part of the MOP because the HRP also uses this clustering.

Exhibit 5 shows the range of realized matrix properties.

Having generated the matrixes, we order the rows and columns such that they are as close as possible to the original empirical correlation matrix. In this way of permutation, we can use the empirical vector of risk and return of the assets and apply them to the simulated correlation matrixes. Therefore, we really exchange only the correlation matrixes and keep the rest unchanged, so matrix order and the quantities empmean and empsds are fixed.

We need to compute the correlation matrix distance (CMD; see Herdin et al. 2005), which compares two correlation matrixes. The CMD becomes zero for equal correlation matrixes and unity if they differ to a maximum extent. The optimal reordering to find the lowest CMD is found by a genetic permutation algorithm, as by Scrucca (2013). Exhibit 3 shows the empirical correlation matrix. Exhibit 6 shows one of the simulated matrixes with optimal permutation to be as close as possible to the original matrix.

Exhibit 7 shows the difference between the two previous matrixes.

Exhibit 8 shows the difference between the two matrixes expressed in terms of features.

We have now produced the input for the regression. The response data for the regression (labels) are created in the following way: For each of the 10,480 synthetic correlation matrixes, we sample 100 times from a multivariate normal distribution.

EXHIBIT 6

Heatmap of a Synthetic Correlation Matrix with Ordering of Columns and Rows as Close as Possible to the Empirical Matrix

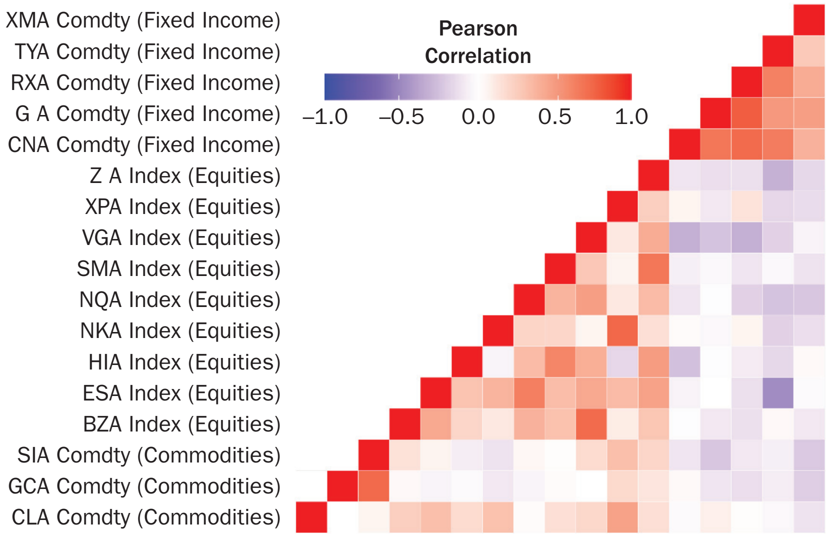


EXHIBIT 7

Heatmap of the Difference between the Two Previous Matrixes

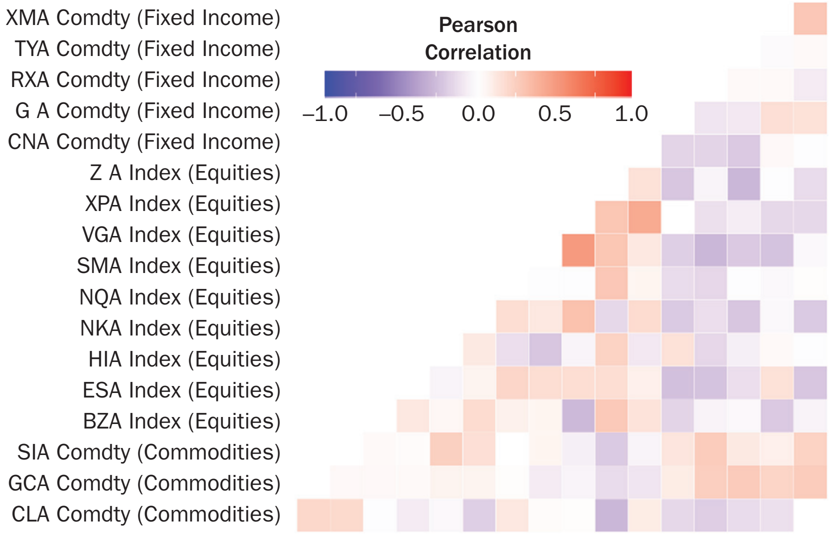


EXHIBIT 8

Multivariate Measures for Correlation Matrixes

	orig_matrix	synthetic_matrix
avg_corr	0.110	0.108
eigen_gini	0.578	0.578
coph_corr_ward	0.862	0.862
coph_corr_single	0.948	0.927
eigenvalues_sum_neg	0.000	0.000
perron_frob_sum_neg	1.264	1.282
power_eigen_values	2.071	1.753

We use the procedure by Ripley (1987) based on the matrix eigendecomposition. This creates multivariate normal deviates, given means, standard deviations, and correlations among the variables.

Based on the sampled returns, we parametrize a Monte Carlo simulation with empirical asset means and standard deviations to run the same walk-forward test (constructing the portfolios based on HRP and ERC), as reported for the original empirical data in daily time steps. We then average the realized annualized portfolio volatilities of HRP and ERC allocations across the simulations for each specific sampled correlation matrix. Thus, for each matrix, we get the annualized volatilities of HRP and ERC. We compute the difference between the annualized standard deviations of portfolio returns from the ERC allocation and that of the HRP allocation to get the annualized portfolio volatility spread in Exhibit 9.

On average, the risk spread between HRP and ERC is 0.007, which means the average risk of ERC is higher (Exhibit 10). Please note that the parametric Monte Carlo simulations lead to different portfolio volatilities compared to the nonparametric backtest using the empirical time series. This is not surprising, because the Monte Carlo simulation averages across many paths, whereas the nonparametric backtest considers only the single historical price paths.

Machine Learning Training Results

We run several ML regression algorithms on a training set with cross-validation, splitting the set 50/50. We have a simple linear model (LM), a classification and regression tree (CART), and XGBoost. Exhibit 11 presents the R^2 statistics.

The synthetically generated data are able to produce valuable information for an ML program because *R*_squared in-sample and out-of-sample can be high. The learned relationships are rather complex, such that only the XGBoost is able to make a reasonably accurate prediction.

EXHIBIT 9

Statistics of Annualized Portfolio Volatility Spread ERC versus HRP

Min.	–0.005
1st Qu.	0.006
Median	0.007
Mean	0.007
3rd Qu.	0.008
Max.	0.011

EXHIBIT 10

Boxplot of the Volatilities across Samples of HRP and ERC

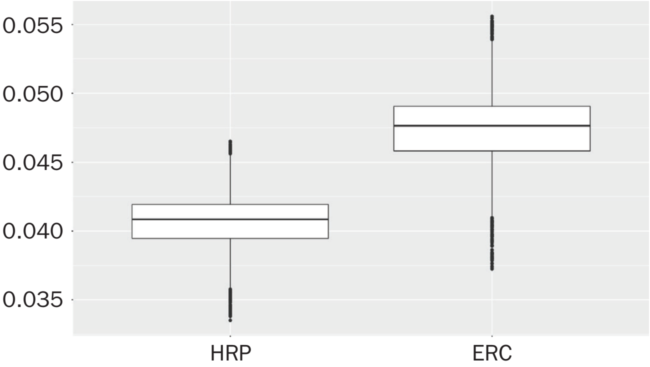


EXHIBIT 11

In-Sample R^2

LM		CART	XGBoost
*R*_squared_train	0.393	0.670	0.955
*R*_squared_test	0.401	0.654	0.901

Exhibit 12 shows the trained decision tree of the CART model. At each branching of the tree, the decision rule together with the cutoff parameter is displayed. The marginal and cumulative probabilities to reach each decision are also visible.

The most important variables are eigen_gini, coph_corr_ward, and avg_corr because those have the highest discriminating impact. Exhibit 13 is the accuracy plot for the train set for XGBoost. The squared correlation coefficient of the point cloud reflects the *R*_squared_train of XGBoost in Exhibit 11.

Exhibit 14 shows the accuracy plot for the test set for XGBoost. The squared correlation coefficient of the point cloud reflects the *R*_squared_test of XGBoost in Exhibit 11.

The predicted portfolio volatility spread between ERC and HRP for the empirical correlation matrix is 0.0077, whereas the true value of the volatility spread in the Monte Carlo simulations using the empirical correlation matrix is 0.0037, so the difference between prediction and true value is 0.0040. This means the prediction error for the risk spread of the empirical matrix is relatively high. This is surprising because the model looks quite accurate and stable out of sample.

There are several approaches to potentially mitigate this issue:

- The label of risk spread of the empirical matrix is not precise enough. Running more Monte Carlo trials for each of the generated matrixes could improve this because it could lead to a more accurate average risk estimate.
- Another approach is to generate more matrixes that are more diverse in the close neighborhood of the empirical matrix. The area seems to be underrepresented by the synthetic data.
- A third approach would be to enhance the features describing the matrixes in the regression.

The advantage of matrix evolutions is that large quantities of data with desired properties can be sampled. The advantage of XML is that it can also be used for model debugging. The next section shows the results of XML.

XML

Next, we explain the ML output locally by means of feature contributions. We extract the SHAP values from the XGBoost model using the TreeSHAP procedure. Based on the trained model, we create an explainer and explain the test data, including the empirical data. First, we rank the features by mean |SHAP| and show the global feature importance of the model in Exhibit 15.

Most important in the model is eigen_gini, followed by coph_corr_ward, and avg_corr.

EXHIBIT 12

Trained CART Model Results

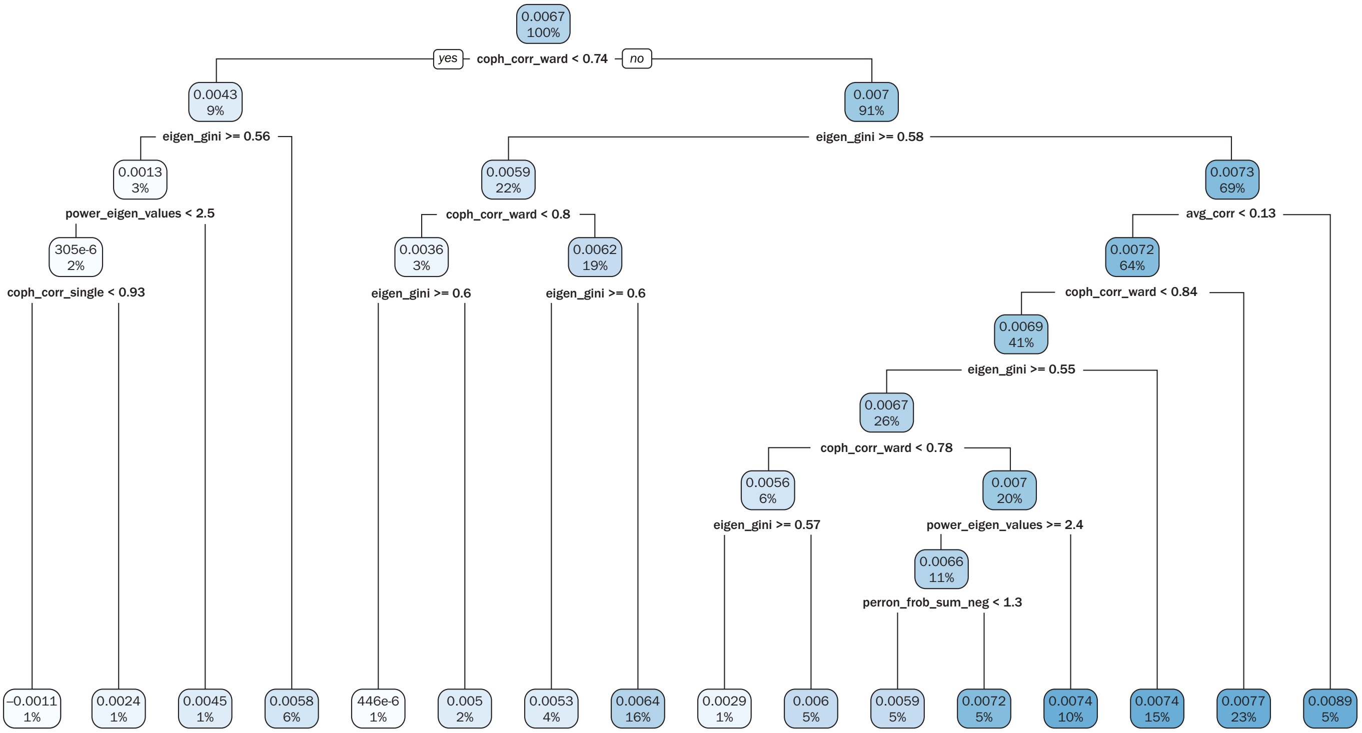


EXHIBIT 13

Accuracy Plot for the Train Set for XGBoost

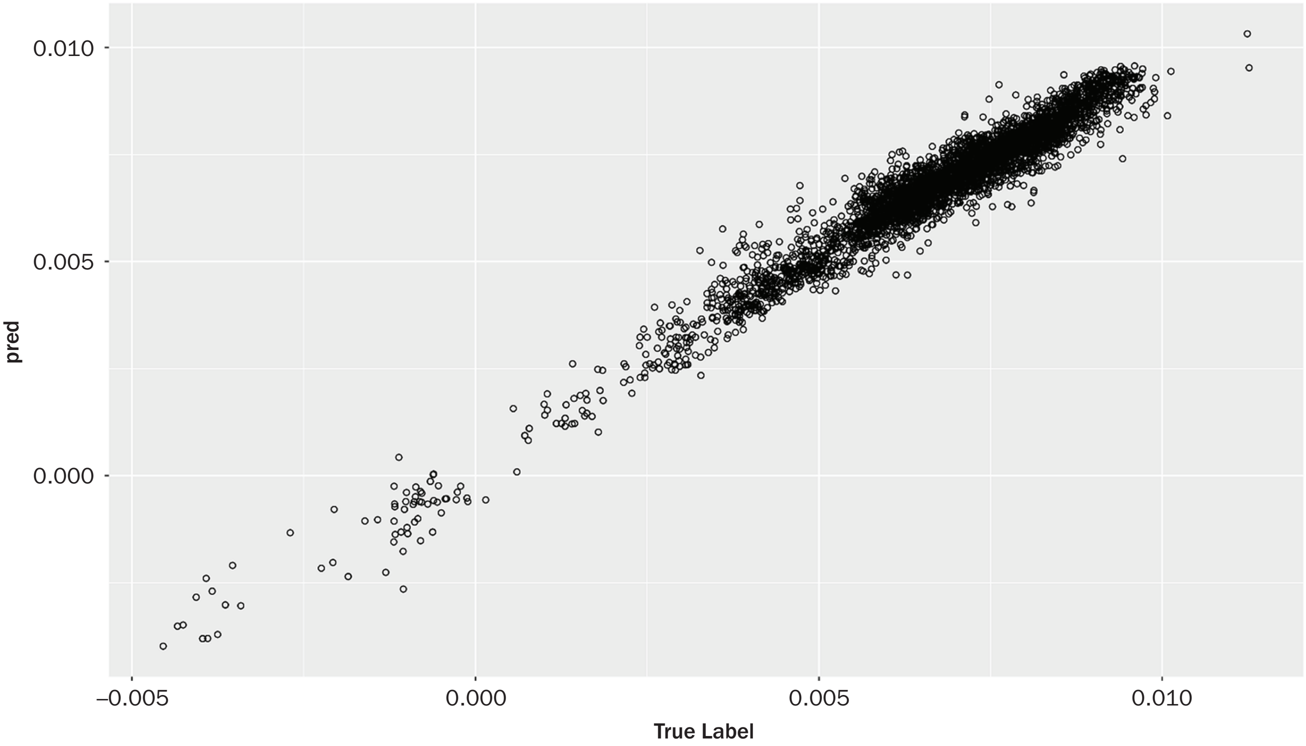
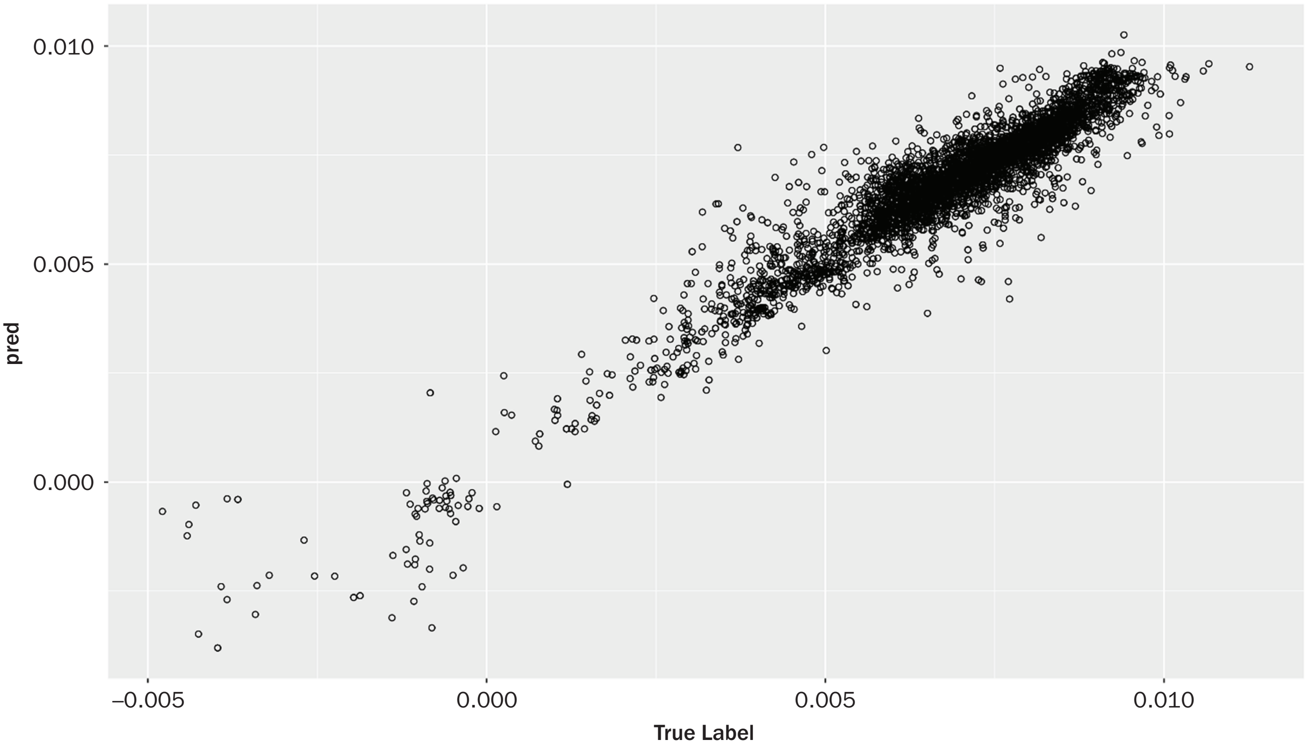


EXHIBIT 14

Accuracy Plot for the Test Set for XGBoost



Next, we use the Liu and Just (2019) package to display the SHAP value of the test set for each feature. Exhibit 16 shows the results. The feature value quantities are color coded.

High-feature values for eigen_gini are connected to negative feature contribution. This means that for concentrated eigenvalues (high-feature eigen_gini), the risk spread prediction is pushed lower, meaning ERC is almost as successful as HRP.

High values for the feature coph_corr_ward (i.e., large Ward-like hierarchical cluster structure) lead to more successful HRP (high-risk spread). This makes sense because several more-pronounced blocks along the diagonal can potentially be handled by the HRP algorithm in a diversifying way. The plot also shows a large average correlation better handled by HRP, still finding ways to diversify.

Exhibit 17 shows similar information in another way as we scatterplot the features versus their SHAP contributions.

The red smoothed lines indicate monotone relationships, but there are no single straight lines or steps, so more complex relationships are involved.

Exhibit 18 groups the data points into clusters. The clusters are found based on the Euclidean distance between the data items with respect to SHAP contributions and the best cut in a Ward hierarchical clustering according to a relatively high level of the cluster quality criterion silhouette width. The outcome is six clusters.

Each cluster stands for another set of SHAP contributions in the trained model, so each cluster represents a specific type of decision making.

Exhibit 19 arranges the data points using the SHAP contributions in *t*-SNE, whereas the previously generated clusters are color coded.

Exhibit 20 shows the breakdown of feature contributions for the empirical set. The intercept reflects the mean portfolio volatility spread across the training set.

EXHIBIT 15

Global Importance of Features Aggregated by Local Feature Contribution

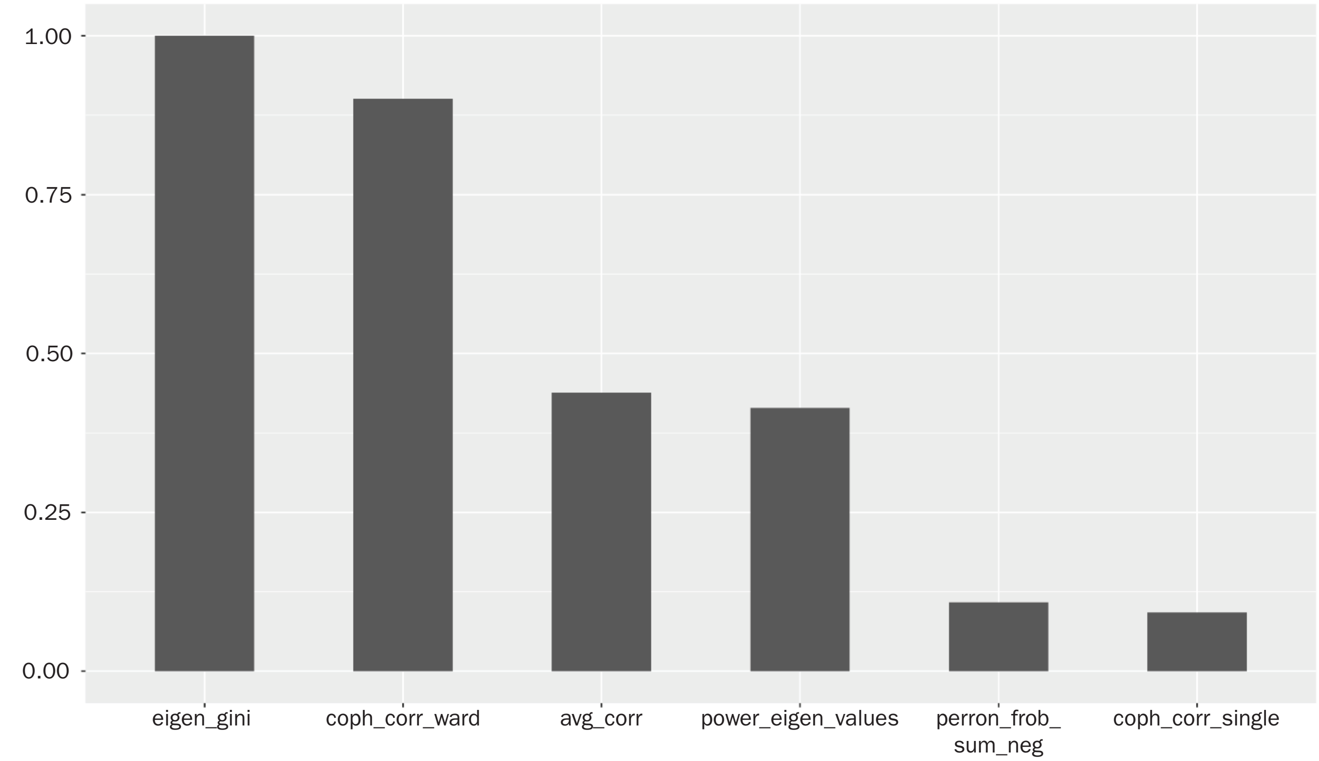


EXHIBIT 16

SHAP Value of the Test Set for Each Feature, Color Coding for Feature Value

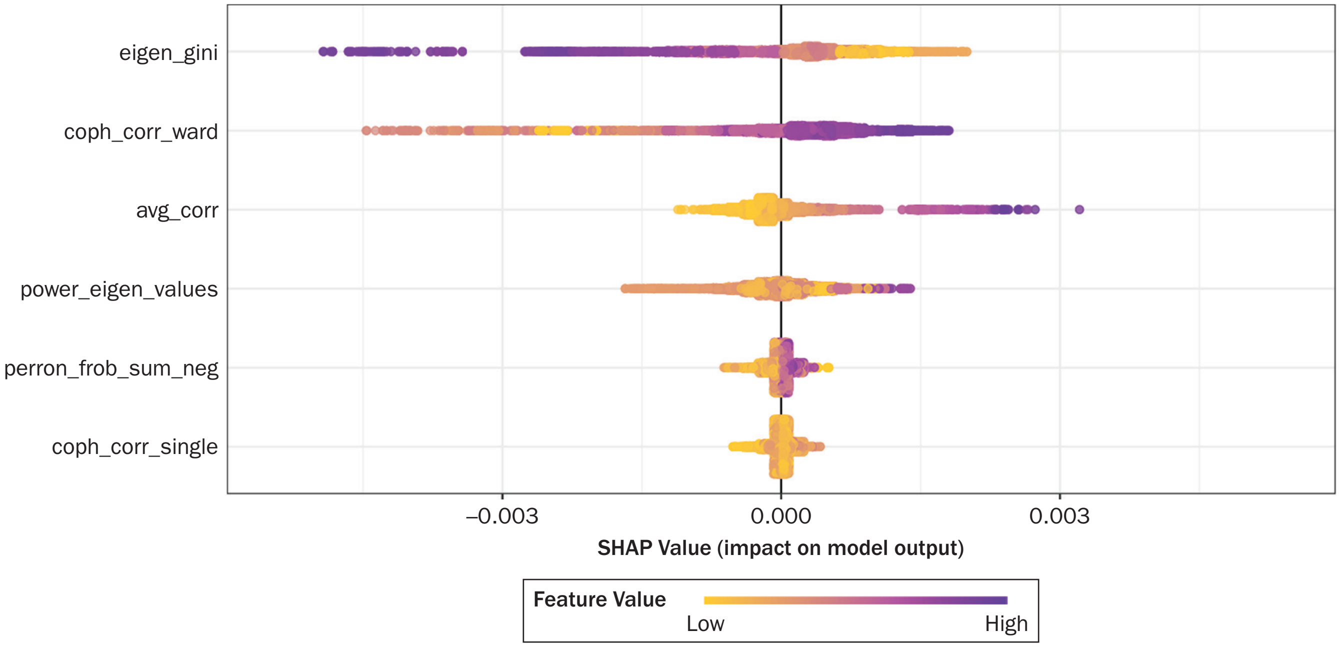


EXHIBIT 17

Scatterplot of the Features versus their SHAP Contributions

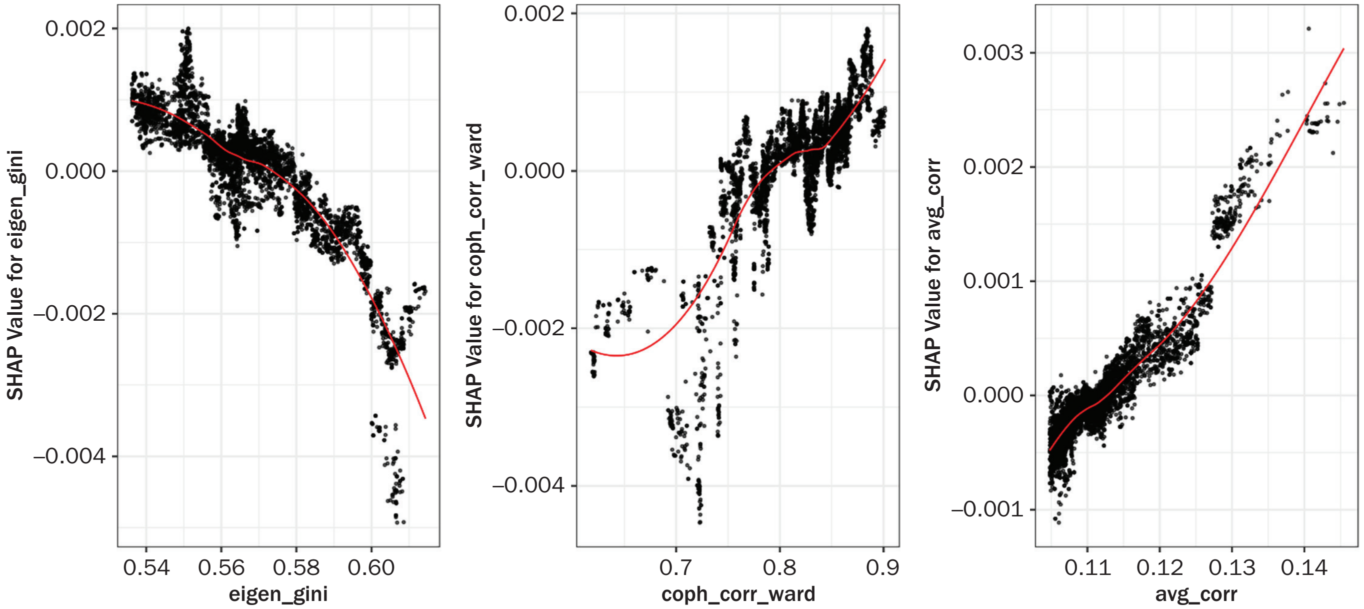


EXHIBIT 18

Force Plot of the Feature Contributions, Separated by Groups

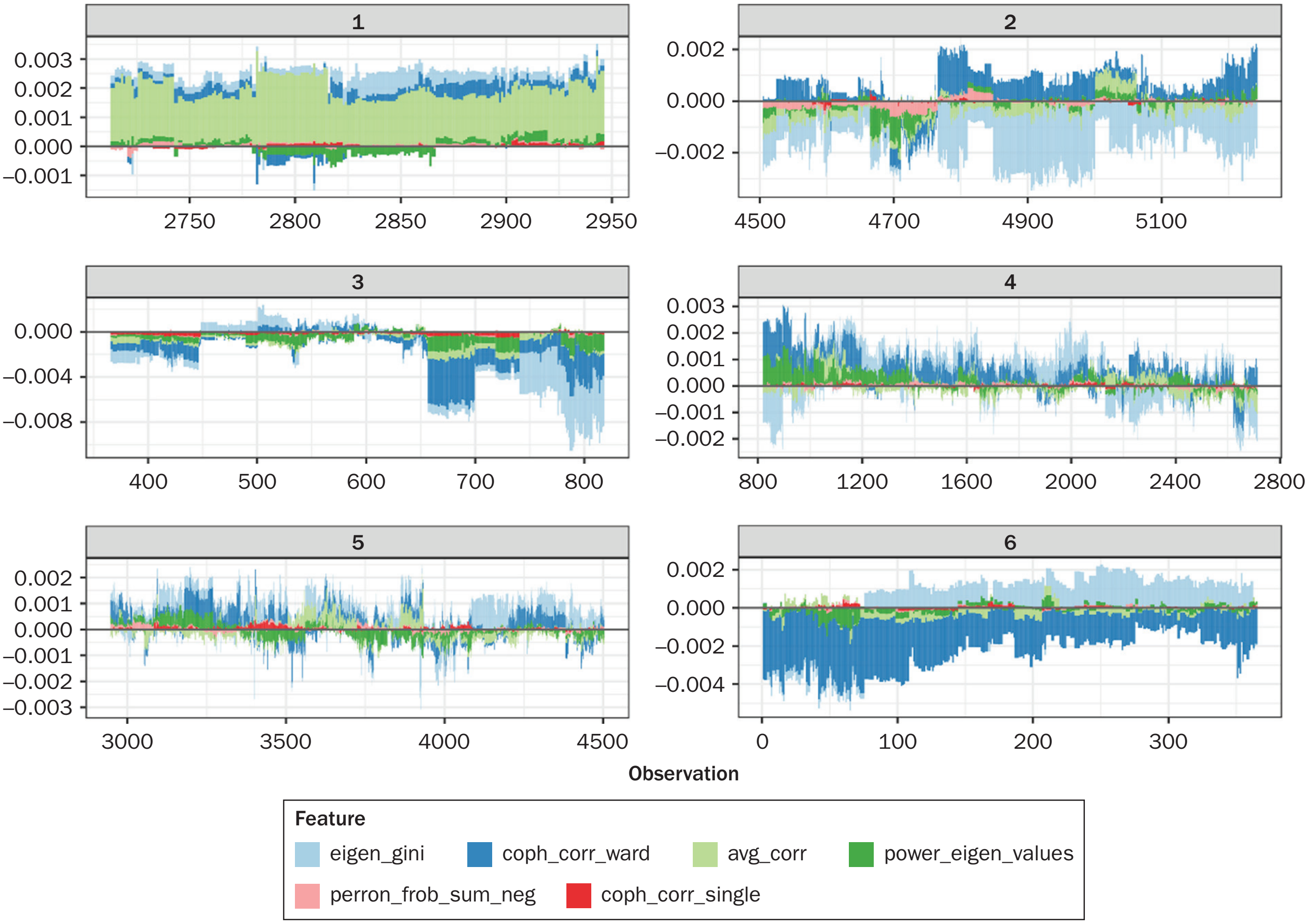


EXHIBIT 19

Arranging the Data Points Using the SHAP Contributions

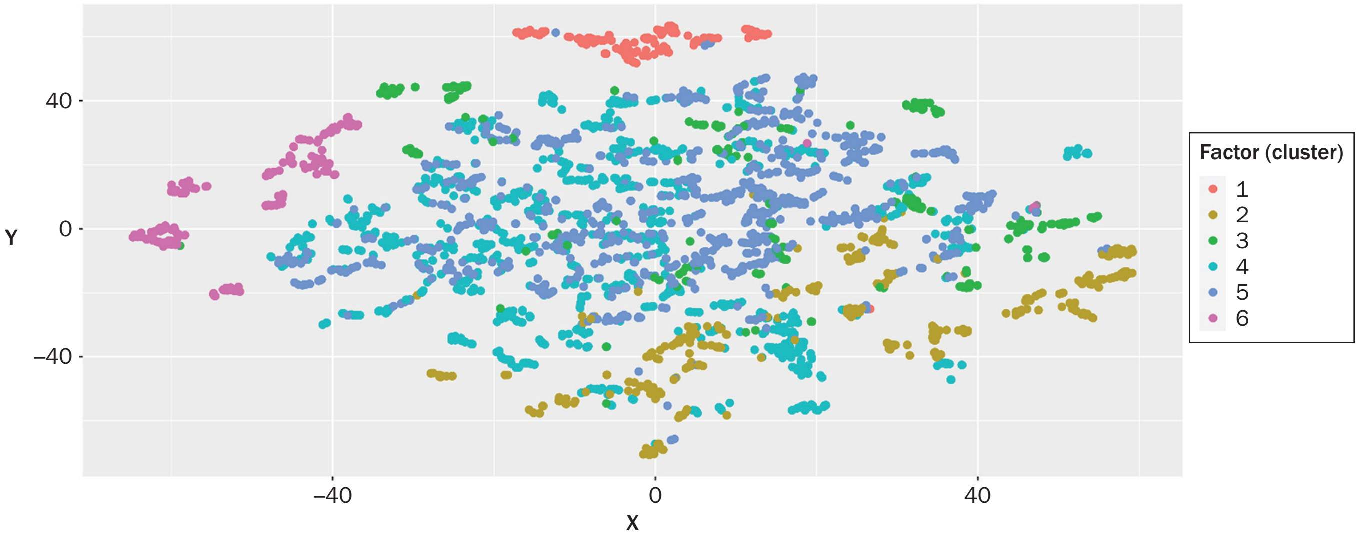
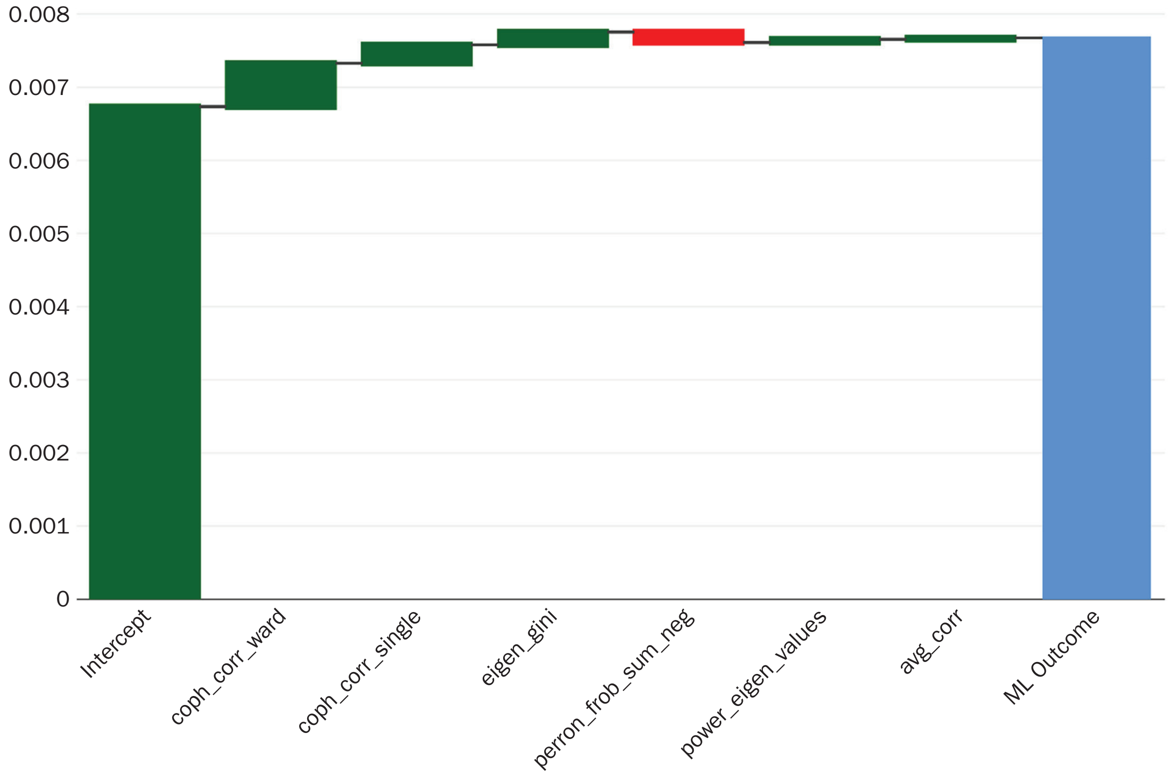


EXHIBIT 20

Breakdown of Feature Contributions for the Empirical Set



We also considered explaining individual predictions when features are dependent on getting more accurate Shapley values, following the procedure of Aas, Jullum, and Løland (2019). However, this methodology does not necessarily lead to better results, as stated by Janzing, Minorics, and Bloebaum (2020).

CONCLUSION

We use evolutionary algorithms to generate realistic correlation matrixes in a novel approach called matrix evolutions. The approach augments the training data space for an explainable ML program to identify the most critical properties in matrixes that lead to the relative performance of competing approaches to portfolio construction. We show that HRP is very robust and that our method can identify the driving variables behind it. Matrix evolutions can be used for many different applications, such as generating risk scenarios for portfolios and pricing of multi-asset derivatives. The entire workflow involving matrix evolutions scales well with technologies of acceleration such as GPUs and quantum-inspired algorithms. In this way, millions of realistic samples can be run to simulate correlated markets.

ACKNOWLEDGMENTS

This research has been sponsored by Munich Re Markets. We appreciate the infrastructure by Open Telekom Cloud and the NVIDIA GPU resources provided for this research. This research was also supported by the European Union's Horizon 2020 research and innovation program FINTECH: A financial supervision and technology compliance training programme, under the grant agreement no. 825215 (topic: ICT-35-2018, type of action: CSA). We would like to thank Gautier Marti for his valuable input and an anonymous referee for helpful comments.

REFERENCES

Aas, K., M. Jullum, and A. Løland. 2019. "Explaining Individual Predictions When Features Are Dependent: More Accurate Approximations to Shapley Values." *arXiv*. http://arxiv.org/abs/1903.10464.

Baitinger, E., and J. Papenbrock. 2015. "Interconnectedness Risk and Active Portfolio Management: The Information-Theoretic Perspective." *Journal of Network Theory in Finance* 1 (4): 1–25.

——. 2017. "Interconnectedness Risk and Active Portfolio Management." *Journal of Investment Strategies* 6 (2): 63–90.

Bussmann, N., P. Giudici, D. Marinelli, and J. Papenbrock. 2021. "Explainable AI in Credit Risk Management." *Computational Economics* 57: 203–216. https://doi.org/10.1007/s10614-020-10042-0.

Chen, T., and C. Guestrin. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. ACM, 2016.

Herdin, M., N. Czink, H. Ozcelik, and E. Bonek. "Correlation Matrix Distance, a Meaningful Measure for Evaluation of Non-Stationary Mimo Channels." In *2005 IEEE 61*st *Vehicular Technology Conference*, Vol. 1, 136–140. IEEE, 2005.

Herlihy, M., and N. Shavit. 2012. *The Art of Multiprocessor Programming, Revised Reprint*, 1st ed. San Francisco: Morgan Kaufmann.

Huettner, A., and J.-F. Mai. 2019. "Simulating Realistic Correlation Matrices for Financial Applications: Correlation Matrices with the Perron–Frobenius Property." *Journal of Statistical Computation and Simulation* 89 (2): 315–336.

Huettner, A., J.-F. Mai, and S. Mineo. 2018. "Portfolio Selection Based on Graphs: Does It Align with Markowitz-Optimal Portfolios?" *Dependence Modeling* 6: 63–87.

Jaeger, M., S. Krügel, D. Marinelli, J. Papenbrock, and P. Schwendner. "Understanding Machine Learning for Diversified Portfolio Construction by Explainable AI." SSRN, 2020. https://ssrn.com/abstract=3528616.

Janzing, D., L. Minorics, and P. Bloebaum. "Feature Relevance Quantification in Explainable AI: A Causal Problem." In *Proceedings of the Twenty-Third International Conference on Artificial Intelligence and Statistics*, edited by S. Chiappa and R. Calandra, 2907–2916. PMLR, 2020. http://proceedings.mlr.press/v108/janzing20a.html.

Liu, Y., and A. Just. "SHAPforxgboost: SHAP Plots for 'XGBoost'." 2019. https://github.com/liuyanguu/SHAPforxgboost.

Lopez de Prado, M. 2016a. "Building Diversified Portfolios That Outperform Out of Sample." *The Journal of Portfolio Management* 42 (4): 59–69.

——. "A Robust Estimator of the Efficient Frontier." SSRN, 2016b, https://ssrn.com/abstract=346996.

——. "Tactical Investment Algorithms." SSRN, 2019. https://ssrn.com/abstract=3459866.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee. 2020. "From Local Explanations to Global Understanding with Explainable AI for Trees." *Nature Machine Intelligence* 2 (1): 56–67.

Lundberg, S., and S.-I. Lee. "A Unified Approach to Interpreting Model Predictions." In *Advances in Neural Information Processing Systems 30*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–4774. Curran Associates, 2017. http://arxiv.org/abs/1705.07874.

Mantegna, R. N. 1999. "Hierarchical Structure in Financial Markets." *The European Physical Journal B-Condensed Matter and Complex Systems* 11: 193–179.

Marti, G. "CorrGAN: Sampling Realistic Financial Correlation Matrices Using Generative Adversarial Networks." *arXiv*, 2019. http://arxiv.org/abs/1910.09504.

Marti, G., F. Nielsen, M. Binkowski, and P. Donnat. 2017. "A Review of Two Decades of Correlations, Hierarchies, Networks and Clustering in Financial Markets." *arXiv*, 2017. http://arxiv.org/abs/1703.00485.

Mitchell, R., E. Frank, and G. Holmes. 2020. "GPUTreeShap: Fast Parallel Tree Interpretability." *arXiv*, 2020. http://arxiv.org/abs/2010.13972.

Oliveira, F. B., D. Davendra, and F. G. Guimarães."Multi-Objective Differential Evolution on the GPU with c-CUDA." In *Advances in Intelligent Systems and Computing*, 123–132. Berlin-Heidelberg: Springer, 2013.

Onnela, J.-P., A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto. 2003. "Dynamics of Market Correlations: Taxonomy and Portfolio Analysis." *Physical Review E* 68 (5): 056110.

Papenbrock, J. "Asset Clusters and Asset Networks in Financial Risk Management and Portfolio Optimization." PhD thesis, Karlsruhe, 2011. https://doi.org/10.5445/IR/1000025469.

Papenbrock, J., and P. Schwendner. 2015. "Handling Risk On/Risk Off Dynamics with Correlation Regimes and Correlation Networks." *Financial Markets and Portfolio Management* 29: 125–147.

Ripley, B. D., ed. *Stochastic Simulation*. Hoboken, NJ: John Wiley & Sons, 1987.

Scrucca, L. 2013. "GA: A Package for Genetic Algorithms in R." *Journal of Statistical Software* 53 (4): 1–37.

Shapley, L. S. "A Value for n-Person Games." In *Contributions to the Theory of Games II*, edited by H. Kuhn and A. W. Tucker, 307–317. Princeton, NJ: Princeton University Press, 1953.

Souza, M. Z., and A. T. R. Pozo. "Parallel MOEA/d-ACO on GPU." In *Advances in Artificial Intelligence—IBERAMIA 2014*, 405–417. Springer International Publishing, 2014.

Tola, V., F. Lillo, M. Gallegati, and R. N. Mantegna. 2008. "Cluster Analysis for Portfolio Optimization." *Journal of Economic Dynamics and Control* 32 (1): 235–258.

Trivedi, A., D. Srinivasan, K. Sanyal, and A. Ghosh. 2017. "A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition." *IEEE Transactions on Evolutionary Computation* 21 (3): 440–462.

Wang, Y., Y. Li, and L. Jiao. 2016. "Quantum-Inspired Multi-Objective Optimization Evolutionary Algorithm Based on Decomposition." *Soft Computing* 20 (8): 3257–3272.

Zhang, Q., and H. Li. 2007. "MOEA/d: A Multi-Objective Evolutionary Algorithm Based on Decomposition." *IEEE Transactions on Evolutionary Computation* 11 (6): 712–731.