

Equity Machine Factor Models

by

Dr Miquel Noguer i Alonso *

Vincent Zoonekynd[†]

December, 2022

*ARTIFICIAL INTELLIGENCE FINANCE INSTITUTE. NYU COURANT. NEW YORK CITY, USA.

E-mail address: `miquel.noguer@aifinanceinstitute.com`

[†]ABU DHABI INVESTMENT AUTHORITY, ABU DHABI, UAE

E-mail address: `vincent.zoonekynd@adia.ae`

Abstract

We examine in this paper the training and test set performance of several equity factor models with a dataset of 20 years of data, 1,200 stocks and 100 factors.

First, we examine several models to forecast expected returns, which can be used as baselines for more complex models: linear regression, linear regression with an L^1 penalty (lasso), constrained linear regression, xgboost and artificial neural networks.

Second, we present a unified framework for portfolio construction, leveraging machine learning for the whole pipeline, from the factor data to the portfolio weights, which scales to a large number of assets and predictors. The results we obtain are interesting and non trivial to interpret; non linear models offer a more balanced outcome considering test set Sharpe ratio and turnover but linear unconstrained models show a good performance in the test set. We introduce a model-free reinforcement learning model, which uses factors to find the portfolio weights maximizing the information ratio.

1 Introduction

We will explore in this paper two of the most fundamental challenges of financial modeling, namely predicting asset returns and covariance matrix of these and finding the optimal weights in a portfolio – Asset Allocation. The traditional approach to quantitative portfolio management is a 3-step process:

- First, forecast the expected returns of the assets under consideration, often, with a linear regression and the covariance matrix of these returns;
- Second, use those forecasts to build a portfolio, through portfolio optimization.
- Third, consider market impact and transaction costs.

Those three steps can be at odds with one another: the first step often minimizes a sum of squared residuals, which has no financial interpretation – the model will make compromises to have good forecasts on average, including for assets that will not be used in the second step.

Machine learning and deep learning have started to take over some of those tasks, in particular the forecasting step using supervised learning methods.

Our contribution is twofold:

- First, we examine a few models to forecast future returns, which can be used as baselines for more complex models:
 - Linear regression;

- Linear regression with an L^1 penalty (lasso);
- Constrained linear regression;
- Neural network;
- Second, we present a unified framework for portfolio construction, leveraging machine learning for the whole pipeline, from the data to the portfolio weights, which scales to a large number of assets and predictors.
 - We start with dozens of “investment factors”: quantities with proven (or believed) predictive power on future returns;
 - We combine them with a neural network, to capture nonlinearities and interactions, to produce a “score”, rather than a return forecast;
 - We turn those scores into portfolio weights, by normalizing them;
 - We optimize the information ratio of the strategy – but the same approach would work with any differentiable objective (drawdown, etc.)

In a follow-up paper, we will combine those approaches by

- Imposing monotonicity constraints on that neural network [Liu et al., 2020, You et al., 2017], to reflect the expected direction in which the investment factors work – that prior knowledge is important;
- Replacing the normalization of the portfolio weights with an actual portfolio optimization – the optimization will then be a layer in the deep learning pipeline [Agrawal et al., 2019].

2 Literature Review

Factor models have been one of the major fields of research in finance from the seminal papers [Fama and French, 2015], [Fama and French, 1993] and [Ross, 1976]. There have been some prior attempts to go directly from data to portfolio weights, with “formulaic portfolios”, using weights proportional to some measure of value [Arnott et al., 2005], some power of capitalization [Fernholz, 2002, Vervuurt, 2015, Fernholz, 2005, Fernholz et al., 1998], or a learned combination of both [Samo and Vervuurt, 2016]. Those attempts are limited to a small number of assets and features [Zhang et al., 2020].

The book [Coqueret and Guida, 2020] offers a detailed coverage of machine learning methods. This paper is a follow-up to [Zoonekynd et al., 2017], with a different dataset and nonlinear models. [Cong et al., 2020] develop a similar idea. Other excellent references are [Gu et al., 2018]. In a very recent paper [de Prado, 2022] explores some of the flaws of these models and suggests new lines of research for factor models.

3 Data and models

We use monthly stock-level data from [Coqueret and Guida, 2020] (20 years, 1200 stocks, 100 features):

- One-month forward returns;

- Investment signals, e.g., earnings yield, momentum, etc.; for each investment signal, we know in which direction it is supposed to be used: for instance, “earnings yield” has a positive impact on future performance, while “volatility” has a negative impact;

We split the data into three periods:

- Training data: until 2016;
- Validation data: 2017–2018;
- Test data: 2019–present.

The data was pre-processed as follows:

- Transform each variable, separately for each date, to make it uniform on $[-1, 1]$;
- Replace the missing values by 0.

We compare the following models:

- Linear regression, to forecast future returns;
- Linear regression with a lasso penalty, to forecast future returns; we have not used the validation period to select a model on the regularization path, but arbitrarily picked the most complex model using at most 10 predictors;
- Linear regression with a constraint on the sign of the coefficients;
- Non-linear model, implemented as a multi-layer perceptron (MLP), to predict future returns, minimizing the sum of squared residuals;
- Linear model, outputting, not return forecasts, but (logarithms of unnormalized) portfolio weights, and optimizing, not a sum of squared residuals, but the information ratio;
- Nonlinear model, outputting portfolio weights, and maximizing the information ratio.

More precisely, let i denote the stocks, t the dates, k the investment signals, x_{ikt} the value of investment signal k for stock i at time t , $y_{i,t+1}$ the log-returns of stock i on $[t, t + 1]$. The lasso finds the coefficients β_k minimizing

$$\frac{1}{N} \sum_{i,t} \left(\sum_k \beta_k x_{ikt} - y_{i,t+1} \right)^2 + \lambda \sum_k |\beta_k|.$$

where λ is a parameter.

The sign-constrained linear regression (or nonnegative OLS) is a similar optimization problem, looking for the coefficients β_k minimizing the sum of squared residuals

$$\frac{1}{N} \sum_{i,t} \left(\sum_k \beta_k x_{ikt} - y_{i,t+1} \right)^2$$

with the penalty replaced by constraints $\forall k \beta_k \geq 0$ if all the signals have a positive predictive power on future returns.

The nonlinear model looks for a non-linear map ϕ , parametrized as

$$\begin{array}{ccccccc} \mathbf{R}^n & \longrightarrow & \mathbf{R}^m & \longrightarrow & \mathbf{R}^m & \longrightarrow & \mathbf{R} \\ x & \longmapsto & Ax & \longmapsto & \sigma Ax & \longmapsto & B\sigma Ax \end{array}$$

where $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{1 \times m}$ are unknown matrices and $\sigma(u) = (1 + e^{-u})^{-1}$ is the elementwise sigmoid function. A deeper neural network would be defined as

$$\phi(x) = B\sigma A_l \sigma A_{k-1} \cdots \sigma A_1 x.$$

Note that there is no sigmoid transform for the output: we want unbounded, both positive and negative return forecasts.

The end-to-end models, computing weights optimizing the information ratio, are a little more complex, but they can be described as a sequence of simple steps:

$\hat{y}_{i,t+1} = \sum_k \beta_k x_{ikt}$	score (it could be a nonlinear function)
$\tilde{w}_{i,t} = \exp \hat{y}_{i,t+1}$	unnormalized weights
$w_{i,t} = \frac{\tilde{w}_{i,t}}{\sum_j \tilde{w}_{j,t}}$	portfolio weights
$\tilde{y}_{i,t+1} = e^{y_{i,t+1}} - 1$	stock ratio-returns on $[t, t+1]$
$\tilde{r}_{t+1} = \sum_i w_{i,t} \tilde{y}_{i,t+1}$	portfolio ratio-returns on $[t, t+1]$
$r_{t+1} = \log(1 + \tilde{r}_{t+1})$	portfolio log-returns on $[t, t+1]$
$\mu = \frac{1}{T} \sum r_t$	average portfolio log-returns
$\sigma = \sqrt{\frac{1}{T-1} \sum_t (r_t - \mu)^2}$	portfolio volatility
$\text{IR} = \frac{\mu}{\sigma}$	information ratio

Those end-to-end models can also be interpreted as a form of reinforcement learning:

- The state is given by the current values of the predictors, the current portfolio holdings, the past portfolio returns;
- The actions are the possible portfolio holdings;
- The reward is zero except for the last time step, when it is the information (or Sharpe) ratio.

Since the model is differentiable (in particular, the actions are continuous and, more importantly, the reward is not given exogenously, but computed, in a known way), we can use a policy gradient algorithm. The last two models also use “mini-batches”: at each iteration, we do not estimate the objective function on the whole training data, but on a random subset of stocks and a random interval of dates. Since the objective is not additive, this is closer to bagging than deep learning mini-batches.

For neural networks, we use the Adam optimization algorithm, with the default parameters.

To compare the models, we built quintile portfolios from their outputs and looked at the performance (returns, volatility, information ratio, turnover, etc.) of the corresponding long-short portfolios.

We provide reproducible code for all the computations.¹

4 Results

Figures 1, 2 and 3 show the performance of the strategies tested:

- The (unconstrained, unpenalized) linear model performs best (this is surprising), but its turnover is very high;
- The constrained linear model does not perform well (this is surprising as well);
- The performance of the lasso is decent, and its turnover lower;
- The nonlinear model to forecast returns presents convergence problems: quite often, the optimization remains stuck around a poor-quality solution;
- The nonlinear model to optimize the information ratio has a good performance and the lowest turnover.

Figure 4 shows the variables used by the lasso.

Figure 5 shows the relation between some of the predictors and the output of the nonlinear model: the relations are always simple, monotonic or V- (or Λ -)shaped, with an occasional glitch around 0 corresponding to missing values.

Model	Portfolio	CAGR (%)	AnnVol (%)	IR	Turnover (%)
unconstrained	LS	21.9	12.4	1.77	52
lasso_26	LS	15.7	13.7	1.14	25
constrained	LS	9.9	14.7	0.67	21
nonlinear	LS	16.6	12.4	1.33	44
IR	LS	15.5	14.6	1.07	17
IR nonlinear	LS	20.9	16.0	1.31	10

Figure 1: Performance of the long-short strategies tested

¹https://github.com/zoonek/2020-12_ML_in_Finance

Model	Portfolio	CAGR (%)	AnnVol (%)	IR	Turnover (%)
unconstrained	5	31.4	19.5	1.61	26
constrained	5	21.1	22.0	0.96	11
nonlinear	5	25.1	21.0	1.20	22
xgboost	L	18.4	14.2	1.29	50
IR	5	23.1	19.2	1.20	9
IR nonlinear	5	27.5	20.3	1.36	6

Figure 2: Performance of the long-only strategies tested, for comparison with [Coqueret and Guida, 2020] (“xgboost”)

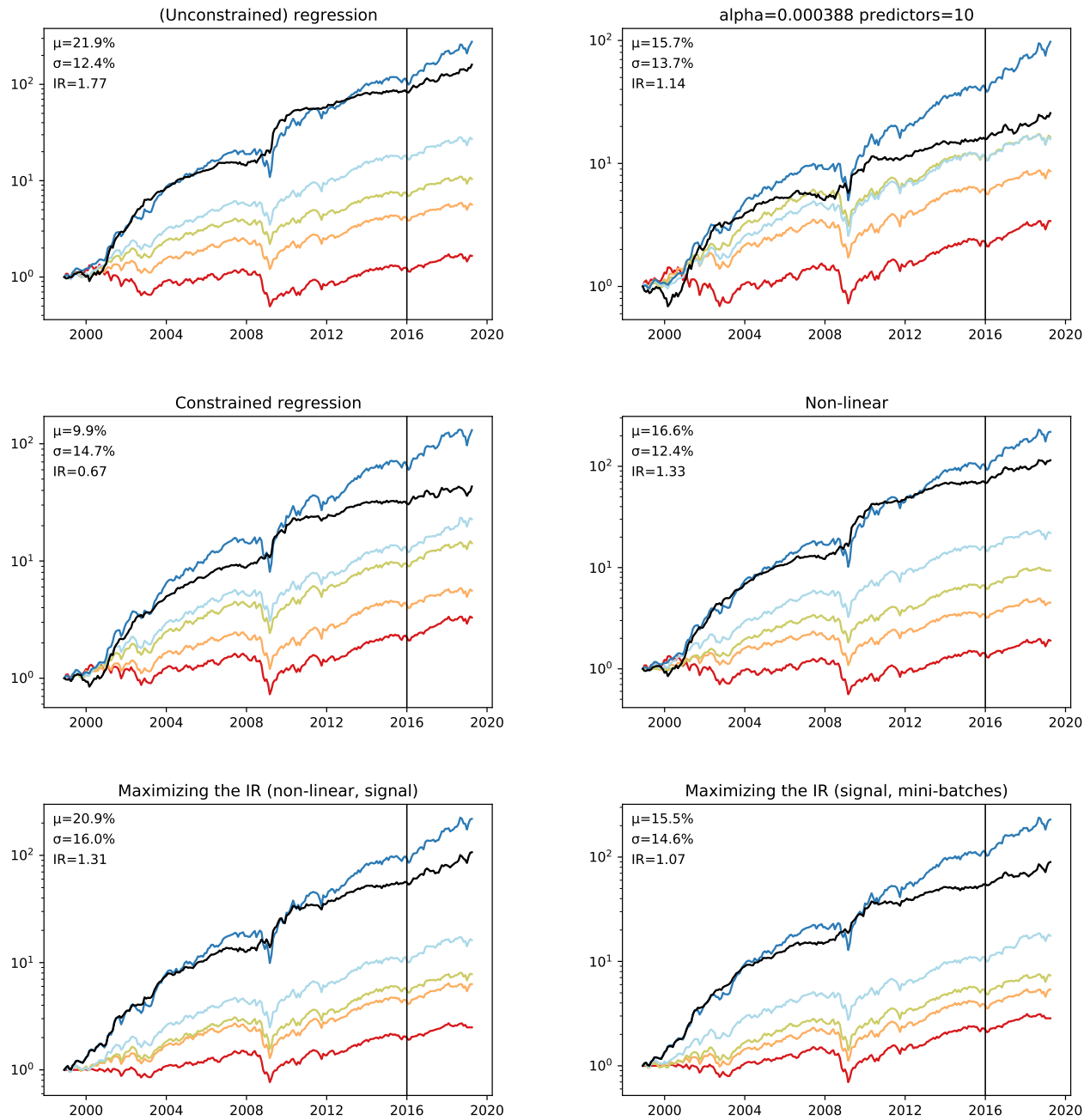


Figure 3: Wealth curves of the quintile portfolios

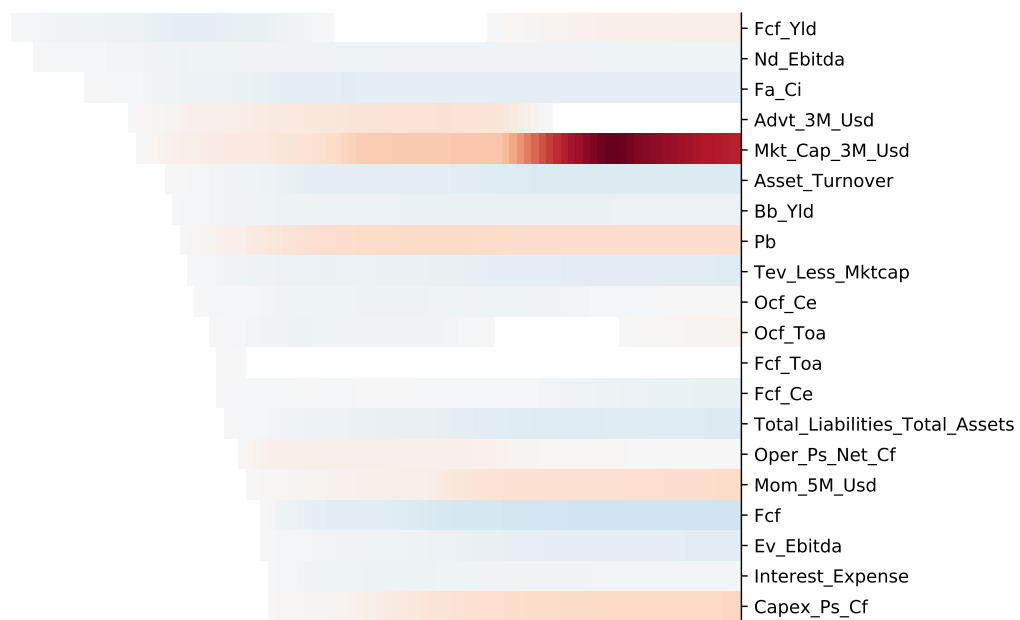


Figure 4: Predictors used in the lasso model

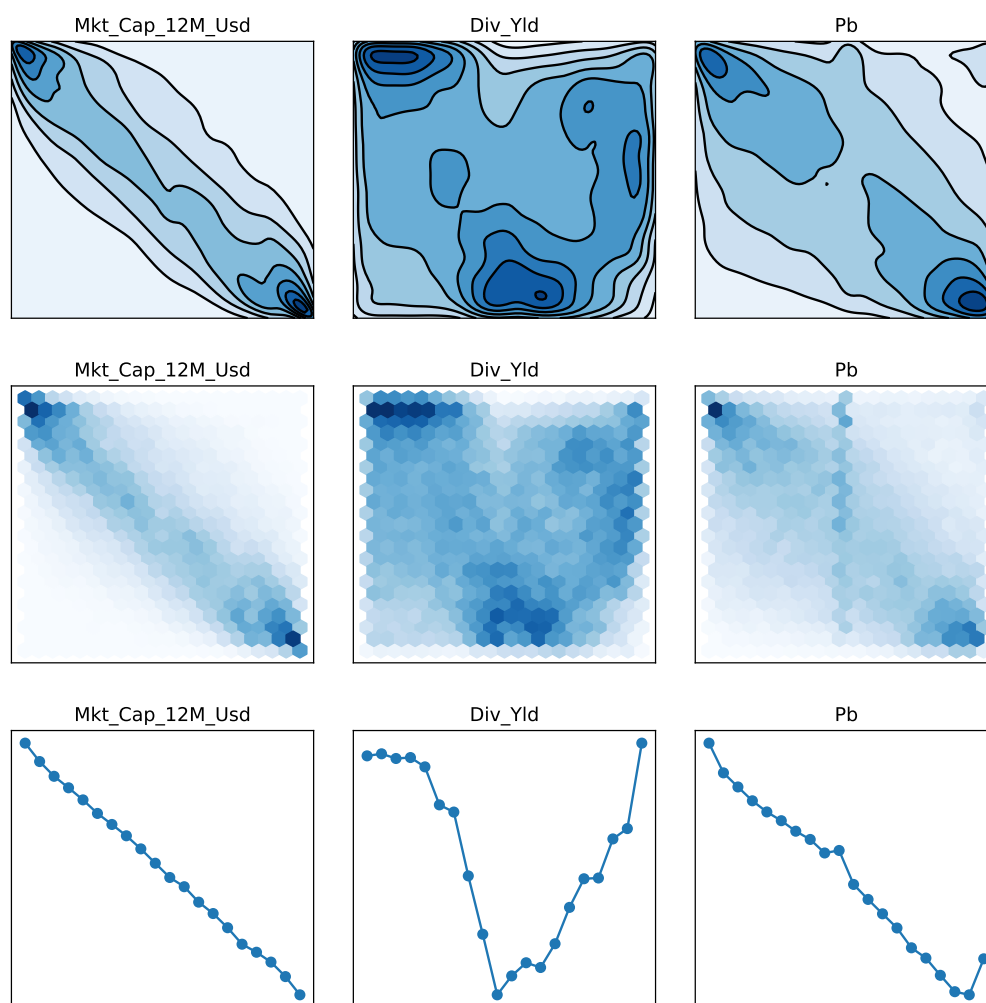


Figure 5: Different ways of examining the relation between each predictor and the output of the nonlinear model maximizing the information ratio: copula density, hexagonal bins, and median-per-quantile. In each plot, the predictor is on the horizontal axis, the variable to predict (future returns) on the vertical axis).

5 Conclusion

We examine in this paper the training and test set performance of several factor models. First, we examine a few models to forecast expected returns, which can be used as baselines for more complex models: linear regression, linear regression with an L^1 penalty (lasso), constrained linear regression, xgboost, artificial neural networks. Second, we present a unified framework for portfolio construction, leveraging machine learning for the whole pipeline, from the data to the portfolio weights, which scales to a large number of assets and predictors.

The results we obtain are interesting and non trivial to interpret: non linear models offer a more balanced outcome considering test set Sharpe ratio and turnover, but linear unconstrained models show a good performance in the test set. We introduce a model-free reinforcement learning model, which uses factors to find the portfolio weights maximizing the information ratio.

References

- [Agrawal et al., 2019] Agrawal, A., Amos, B., Barratt, S., Boyd, S., Diamond, S., and Kolter, Z. (2019). Differentiable Convex Optimization Layers. *arXiv:1910.12430 [cs, math, stat]*.
- [Alonso et al., 2018] Alonso, M. N., Batres-Estrada, G., and Moulin, A. (2018). *Deep Learning in Finance: Prediction of Stock Returns with Long Short-Term Memory Networks*, chapter 13, pages 251–277. John Wiley & Sons, Ltd.
- [Arnott et al., 2005] Arnott, R. D., Hsu, J., and Moore, P. (2005). Fundamental Indexation. *Financial Analysts Journal*, 61(2):83–99.
- [Cong et al., 2020] Cong, L., Tang, K., and Wang, J. (2020). AlphaPortfolio: direct construction through deep reinforcement learning and interpretable AI. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3554486.
- [Coqueret and Guida, 2020] Coqueret, G. and Guida, T. (2020). *Machine Learning for Factor Investing: R Version*. Chapman and Hall/CRC, 1 edition.
- [de Prado, 2022] de Prado, M. L. (2022). Causal factor investing: Can factor investing become scientific? https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4205613.
- [Dixon and Polson, 2020] Dixon, M. and Polson, N. (2020). Short Communication: Deep Fundamental Factor Models. *SIAM Journal on Financial Mathematics*, 11(3):SC–26–SC–37.
- [Fama and French, 2015] Fama, E. F. . and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.
- [Fama and French, 1993] Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56.

- [Fernholz, 2002] Fernholz, E. R. (2002). *Stochastic Portfolio Theory*. Springer New York, New York, NY.
- [Fernholz, 2005] Fernholz, R. (2005). Stock market diversity. Technical report, Intech.
- [Fernholz et al., 1998] Fernholz, R., Garvy, R., and Hannon, J. (1998). Diversity-Weighted Indexing. *The Journal of Portfolio Management*, 24(2):74–82.
- [Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- [Gu et al., 2018] Gu, S., Kelly, B., and Xiu, D. (2018). Empirical asset pricing via machine learning. NBER Working Papers 25398, National Bureau of Economic Research, Inc.
- [Liu et al., 2020] Liu, X., Han, X., Zhang, N., and Liu, Q. (2020). Certified Monotonic Neural Networks. *arXiv:2011.10219 [cs]*.
- [Nakagawa et al., 2019] Nakagawa, K., Ito, T., Abe, M., and Izumi, K. (2019). Deep Recurrent Factor Model: Interpretable Non-Linear and Time-Varying Multi-Factor Model. *arXiv:1901.11493 [cs, q-fin]*. arXiv: 1901.11493.
- [Nakagawa et al., 2018] Nakagawa, K., Uchida, T., and Aoshima, T. (2018). Deep Factor Model. *arXiv:1810.01278 [q-fin]*. arXiv: 1810.01278.
- [Noguer and Srivastava, 2020] Noguer, M. and Srivastava, S. (2020). Deep reinforcement learning for asset allocation in us equities.
- [Ross, 1976] Ross, S. A. (1976). The arbitrage theory of capital asset pricing. *Journal of Economic Theory*, 13(3):341 – 360.
- [Sak et al., 2021] Sak, H., Huang, T., and Chng, M. (2021). Exploring the Factor Zoo With a Machine-Learning Portfolio. SSRN Scholarly Paper ID 3202277, Social Science Research Network, Rochester, NY.
- [Samo and Vervuurt, 2016] Samo, Y.-L. K. and Vervuurt, A. (2016). Stochastic Portfolio Theory: A Machine Learning Perspective. *arXiv:1605.02654 [q-fin, stat]*.
- [Vervuurt, 2015] Vervuurt, A. (2015). Topics in Stochastic Portfolio Theory. *arXiv:1504.02988 [math, q-fin]*.
- [You et al., 2017] You, S., Ding, D., Canini, K., Pfeifer, J., and Gupta, M. (2017). Deep Lattice Networks and Partial Monotonic Functions. *arXiv:1709.06680 [cs, stat]*.
- [Zhang et al., 2020] Zhang, Z., Zohren, S., and Roberts, S. (2020). Deep Learning for Portfolio Optimization. *The Journal of Financial Data Science*.
- [Zoonekynd et al., 2017] Zoonekynd, V., LeBinh, K., Tang, J., Sambatur, H., and Lai, E. (2017). End-to-end portfolio construction. Technical report, Deutsche Bank.