École Polytechnique Fédérale de Lausanne

Machine Learning Project

Decoding Future Stock Returns: Revealing the Key Predictors

FIN-407, Spring 2024

Gustave Paul Besacier (376507)
Guillaume Ferrer (311582)
Agustina María Zein (310254)
Eva Perazzi (377433)

June 7, 2024

Abstract

This report investigates 212 firm characteristics proposed by Chen and Zimmermann in their paper on *Open Source Cross-Sectional Asset Pricing*. We aim to find a concise subset of these characteristics that effectively describes stock returns. In order to run machine learning methods, we applied thorough data preprocessing to eliminate highly correlated predictors and used data fitting algorithms to address the high density of missing values. The methods used include Sequential Forward Floating Selection (SFFS), Bootstrapped-Enhanced Lasso, Bootstrapped-Enhanced Elastic Net, and Nonparametric Estimation. For each method, we described the implementation process, results, and limitations. Each model successfully reduces the number of predictors, we assessed the quality of these subsets using a deep neural network, **it ... accuracy/mean error.**
Keywords: Stock return prediction, Machine learning, Penalized least squares, Neural network.

I Introduction

Since the early ages of Finance, prediction of future stocks' returns has been in the center of most debates, despite it going in the opposite direction of the efficient market hypothesis, under which stock returns would be true martingales. Under this assumption, prediction would be pointless, as stocks' price would either go up or down with equal probability at each time. However, market movements exhibit a behavior that does not align with this hypothesis. As a result, analyzing and accurately predicting financial returns is fundamental for effective investment management, to help investors mitigate risks and optimize portfolio performance. However, putting all efforts and ressources in building state of the art models is useless if the data is not handled properly.

In the last decades, with the increasing connectivity between markets, consolidation of regulations and technology progresses, data has become larger and larger. This results in a drastic increase of potential features influencing the stock returns. Therefore, it is not too difficult to find links between any kind of data and stock returns, however weak they might be. When trying to identify the set of predictors of stock returns, the cardinality of its universe gets quickly huge, and leads to heavy models and high computational costs. On top of that, the marginal benefits of high complexity over lower ones (CAPM, Fama-French, Carhart, ...) might not be worth it.

In that spirit, we propose methodologies for feature selection, in a view of stock return prediction. In part III, we formally define the problem we intend to solve, which is the core objective of our work. In part IV, we present data extraction and explore data structure and characteristics. Then, we provide detailed methods for data handling and the way to extract as much information as possible from sparse data in part V. In part VI, we introduce our algorithms for feature selection: SFFS, BE-Lasso and BE-ENet, and non-parametric adaptative group Lasso. Results and findings are discussed in part VII. Finally, we introduce FLASH, a deep neural network trained for predicting trading signals in part VIII.

II Related Literature

Many papers have tried to harness the power of firms characteristics to predict their expected returns. The most famous and trivial example would be the Capital Asset Pricing Model (CAPM) developed independently by William Sharpe (1964), John Lintner (1965) and Jan Mossin (1966). It introduced the concept of β, which describes the relationship between the expected return of a security and its systematic risk relative to the overall market [1]. More recently, Chen and Zimmermann wrote *Open Source Cross-Sectional Asset Pricing* (2021) [2], where they tried to assess

the statistical significance of 319 firm-level characteristics for long-short portfolio returns. The results were statistically significant, meaning that including extra characteristics could improve returns forecasting.

Using these extra predictors appears practical. However, the vast majority of this data has missing values. Tackling this issue as well as predicting returns presents challenges as shown by Joachim Freyberger, Björn Höppner, Andreas Neuhierl and Michael Weber in *Missing Data in Asset Pricing Panels*(2022) [3]. They used a generalized method of moments (GMM) in a cross-sectional way to simultaneously fit the missing entries and find an efficient estimator for predciting returns. Svetlana Bryzgalova, Sven Lerner, Martin Lettau and Markus Pelger introduced another way of fitting data by harnessing not only the cross-sectional data but also the time series information to infer missing values. The major drawbacks of the methods presented above is the assumption of linearity or even polynomial interaction in the characteristics. In *Dissecting Characteristics Nonparametrically*", Joachim Freyberger, Andreas Neuhierl and Michael Weber (2017) [4] highlighted the effectiveness of nonparametric approaches in capturing stock return variability.

III Problem Definition

In total, we have N_{tot} assets. However, since the assets do not all exist at the same time, we note N_t the number of assets at time t. Furthermore, each asset n has a return at time t of $\mathcal{R}_t^n$ and we define the vector of all stocks return at time t as $\mathcal{R}_t = (\mathcal{R}_t^1, ..., \mathcal{R}_t^{N_t})^\top$. We postulate that this return depends on a maximum of P predictors at time $t-1$. $\mathcal{P}_t^{n,p}$ represents the value of the p-th predictor at time t for asset n. Finally, we write the vector of predictor for asset n at time t as $\mathcal{P}_t^n$.

Definition 1. The predicted return is defined as the expectation of the returned conditioned on last period predictors.

$$\hat{\mathcal{R}}_t^n = \mathbb{E}[\mathcal{R}_t^n | \mathcal{P}_{t-1}^n] \tag{1}$$

Let us denote $\mathcal{S}$ with cardinality $|\mathcal{S}| = p$ the set of all predictors, regardless of time. We want to find the optimal subset $\mathcal{S}^*$ such that $\mathcal{S}^* \subseteq \mathcal{S}$. We aim for $\mathcal{S}^*$ to best approximate $\hat{\mathcal{R}}_t^i$, while selecting the "most important" predictors forecasting future returns. Therefore, we penalize large models.

To find the best estimates $\mathcal{R}_t$, we use the least squares framework. This method keeps the integrity of our predictors while identifying the most appropriate ones, denoted as $\mathcal{S}^*$. This might not be feasible with more sophisticated machine learning algorithms, such as Recurrent Neural Networks.

$$\mathcal{L}_{[\theta,T]} = \sum_{\tau=\theta}^{T} \mathcal{L}_\tau = \sum_{\tau=\theta}^{T} (\mathcal{R}_t - \hat{\mathcal{R}}_t)^2 = \sum_{\tau=\theta}^{T} \sum_{n=1}^{N_\tau} (\mathcal{R}_t^n - \mathbb{E}[\mathcal{R}_t^n | \mathcal{P}_{t-1}^n])^2 \tag{2}$$

where θ is the starting index of our model and T the last one.

Given both intuitions, we turn ourselves towards regularization methods to make the space of predictors sparse.

$$\mathcal{L}_{[\theta,T]} = \sum_{\tau=\theta}^{T} [\mathcal{L}_\tau + \lambda G(\mathcal{P}_{\tau-1})] \tag{3}$$

where G denotes the penalizing function on the predictors.

IV Data Analysis

A Data Acquisition

The predictors are retrieved from a dataset created by Andrew Y. Chen and Tom Zimmermann [5]. The file contains 209 predictors for a total of $N_{tot} = 37'774$ stocks (different `permno` number). Using the same `permno` number and date interval, we retrieved the corresponding monthly returns `STreversal` (stock return over the previous month), `Price` and `Size` from the Center for Research in Security Prices (CRSP, [6]). Chen and Zimmermann made available a piece of code intended to perform this manipulation (available on their GitHub repository, [7]), but it required further development to better fit our needs. The `STreversal` incorporates the delisting return bias[1] according to the paper written by T. Shumway, [8]. The `Price` is the negative of the natural logarithm of the absolute value of the stock price[2] from the CRSP data [6]. Using this transformation of the stock price serves to scale values and favor better interpretation. Similarly, the `Size` represents the negative of the natural logarithm of market equity, calculated as the absolute value of price multiplied by shares outstanding. In the end, our dataset consists of a set of 212 predictors.

B Data Analysis

The raw data needs to be analyzed to get an overview on the way to process it. In Table 1, we summarize the raw description of the data.

Years	Stocks	Predictors	Shape	Entries	% `NaNs`
100	37'774	212	5'249'456 × 215	1'128'633'040	56.05%

Table 1: Raw description of the data

As we can see, the proportion of `NaNs` is quite high. To get a better understanding of their distribution, we plot the `NaN` proportion as a function of time for three randomly selected predictors in the dataset. The results are presented in Figure 1.

[1] According to Shumway, CRSP would adjust the return of stocks from companies being delisted or that cease trading on a particular exchange. These can be *surprise*, in the sense that markets might not anticipate it beforehands. To that extent, CRSP would correct returns to reduce the impact of delisting surprises [8].

[2] Taking the absolute value of the price, even if it can seem to be counterintuitive as people would expect it to be positive at any point in time, is due to the fact that if no closing price is available for a stock at the end of the trading period, CRSP averages the bid and ask. To distinguish from the *real* value, it is then set to negative.

Figure 1: Proportion of `NaN`s as a function of time

We observe that the proportion of `NaN`s is higher when going back further in time. This phenomenon will be studied more in details later on. Moreover, we plot the predictors occurrences for fifteen random predictors. These visualizations provide an overview and a closer examination of the behavior of our data, giving us a precise input on how to handle it. The results are given in Figure 2.

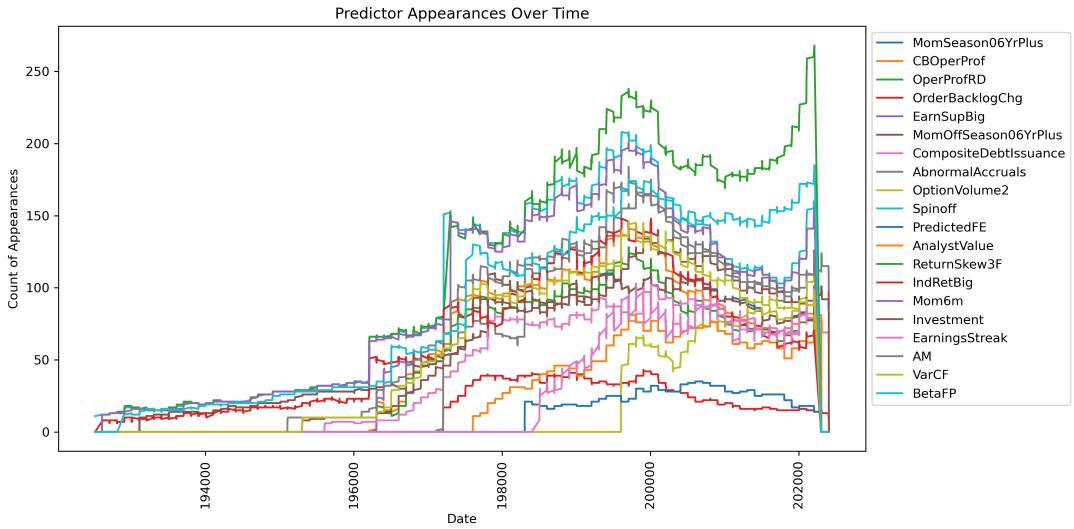

Figure 2: Predictor appearances over time, the spikes represent some yearly observations.

We notice in the figure that, before the 1970s, the data is much sparser and the predictors seem to appear less or not at all. This can be explained by the fact that information 100 years ago was

much harder to fetch, store and archive. Also, regulations and compliance gained in importance, leading to an increasing amount of data.

This strongly suggests that different dates or periods of time carry different weights, and we need to adapt our approach accordingly, which will be detailed later on in this report. Indeed, if we plot the number of observations per date, as we can see in figure 3, this phenomenon is clearly observable.

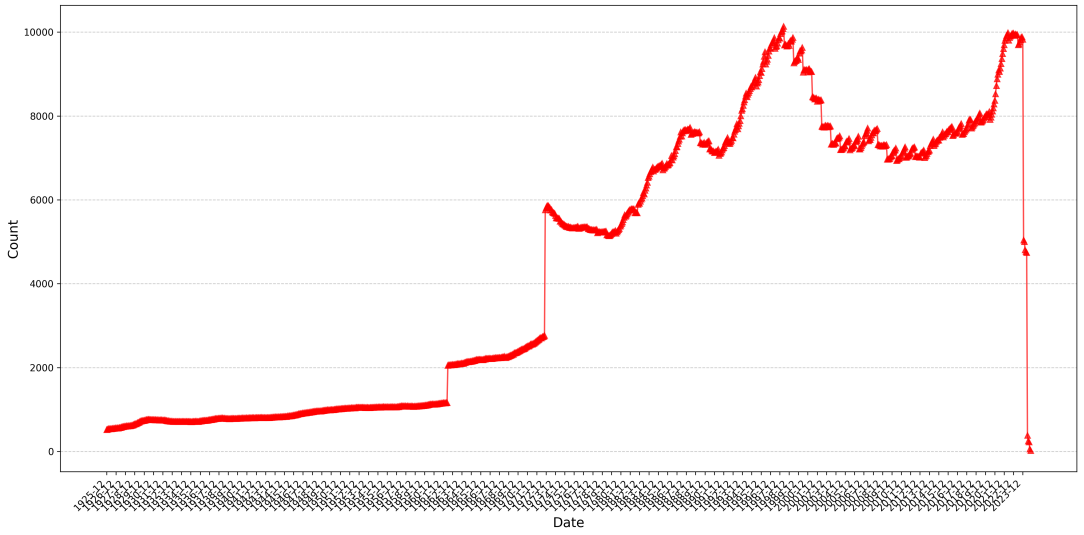

Figure 3: Observations appearances

We also find interesting to analyze the occurrences of the different predictors. As we can see in Figure 4, some predictors appear only in a few stocks. Such predictors will be the first removed from the list of candidates as they do not exhibit enough occurrences throughout the stocks, making them less useful for inference and prediction.

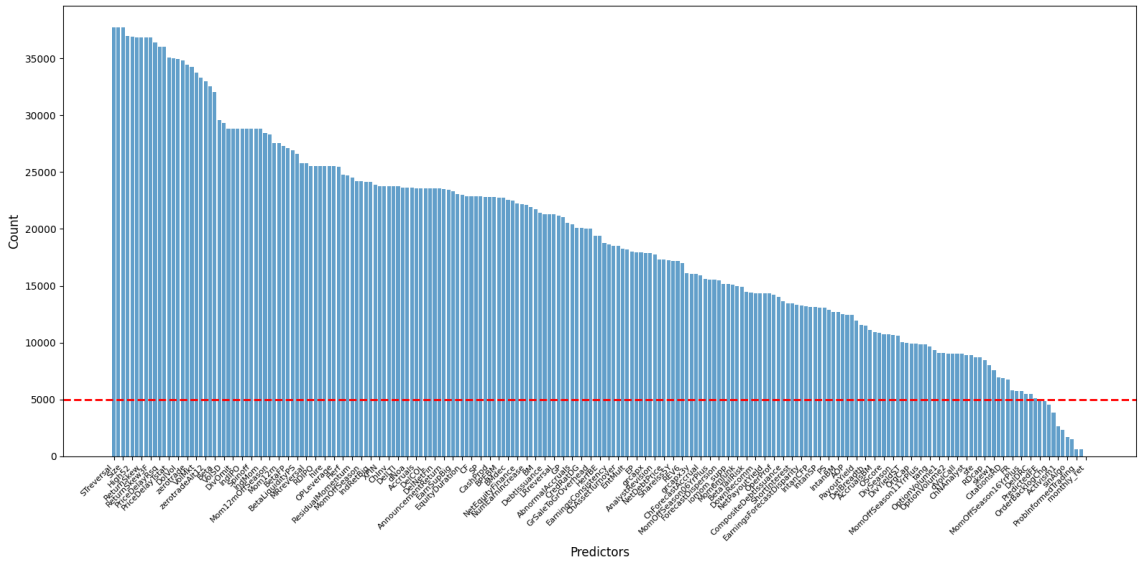

Figure 4: Predictors occurrences among the stocks

We established a threshold of 5'000 companies, meaning that a predictor will be considered as a candidate if it has a non-empty value for at least 5'000 different `permno` numbers.

C Predictor handling

The number of predictors available for our study being high, we proceed to two selection processes to narrow down our predictor set in a meaningful way.

We started by conducting a mechanic selection, that is we dropped from the dataset the predictors that appeared in less than 5'000 stocks, as explained above. Alongside this, we decided to exclude from the analysis predictors that were too empty, meaning predictors that exhibited over 85% `NaN` values.

After this first cleaning process, we decided to split the data. As motivated above, the given time period was too wide and for the data from the 1930s to have the same weight as nowadays' made little to no sense. To avoid such conflicts, we decided to split our dataset into 9 time periods (windosw) of 15 years, with 5 years overlap every time, and we focused our study of the last three periods, that is 1985 - 2000, 1995 - 2010 and 2005 - 2020, leaving the last years of data to do our final testing. Such time intervals were chosen to enhance explainability, precision and relevance of the results. The 5 year overlaps were used to avoid cutting neatly our dataset but do it in a more continous way, while still being consise. The subdatasets that we are considering exhibit the characteristics presentend in table 2.

Time windows	1985 - 2000	1995 - 2010	2005 - 2020
Observations	1'455'173	1'490'101	1'328'018
Predictors	62	64	81
Firms	18'742	17'573	15'004

Table 2: Characteristics of the different time windows

Once the data was split, we conducted on each subdataset a discretionary selection. Using the predictors description made available at here, we gained a better understanding of the predictors we have in our dataset. We went through each description and performed a selection process, removing those that seemed to be highly similar. For example, `Accruals` and `AbnormalAccruals` or `Activism1` and `Activism2` (both measure the accruals of a company using slightly different definitions). We also dropped predictors that were combination of others, such as `AOP` being

$$\texttt{AOP} = \frac{\texttt{AnalystValue} - \texttt{IntrinsicValue}}{|\texttt{IntrinsicValue}|}$$

while still retaining `AnalystValue` and `IntrinsicValue`. In addition, we removed the predictors labeled as *Lack of data*, as they would not allow for robust inference in any case. Hence, from the original 212 predictors, we ended up with 154.

Finally, to filter bad quality data, we decided to conduct a cluster analysis, that is we grouped predictors with a correlation criterion and kept only one representant of each group. We chose a threshold of 0.7 to define highly correlated predictors. Let us note that it is slightly lower that what is classically done in statistics, but this was motivated by the fact that we wanted to filter out as much predictors as possible. This was done to avoid near-multicollinearity and useless expense in computations. This step was implemented as follows:

1. For each window of 15 years, group the observations per date. If there were multiple observations for the same date, we took their mean. The `NaN` values present in the data were replaced by the median of the corresponding predictors, as this influences minimally the correlations and allows us to keep control on how the empty values are manipulated.
2. Then, we created a list with the pairs predictors that exhibited a correlation of 0.7 and above.
3. From the list of highly correlated predictors, we tried to add, to each pair of predictors, another variable from the list, each time checking if it was highly correlated to each member of the initial group. The algorithm stopped as soon as no more variables could be added to the group.
4. Repeat step 3 for each pair of predictors.
5. Remove groups that are repeated.
6. Remove predictors that are repeated in 2 or more groups.

This algorithm lead us to a certain number of groups of variables, for each window of time, that were highly correlated with each other. From there on, we were able to select only one representative from each group, based again of economic criteria. To that new list of predictors, we added those that did not exhibit any high pairwise correlation with the other predictors. This allowed us to keep only the predictors that brought meaningful and new information.

V Data Preprocessing

A Handling missing data

A.1 Simple fitting

As described in the Data Analysis, we have many empty values. Instead of only keeping predictors that occur at every time index and for every stock (`permno`), we will try to fill some predictors to harness some of the explanatory power they have. We chose to handle stocks individually, as the motion of the predictors are firm specific. Hence, running for instance a k-nearest neighbour or random forest to infer missing values would make us loose distinctive features of the firm and lead to a soft homogenization of the sample. For each firm, we will analyze and fit the missing values according to the cross-sectional motion of the 154 other predictors.

Our initial idea was inspired from Freyberger and al. [3]. We tried splitting the set of predictors of stock n denoted by $\mathcal{S}^n$ in two. The first set of predictors $\mathcal{S}^n_f$ is the one where the observation are present for all indices of the timeline and the second set $\mathcal{S}^n_m$ contains the rest of the predictors; the ones that have missing observations.

Let us write $\overline{\mathcal{P}}^{n,p}_t$ the p-th predictor belonging to the missing set $\mathcal{S}^n_m$ and $\mathcal{P}^{n^c}_t$ the predictors belonging to $\mathcal{S}^n_f$. We define a new variable $\tilde{\mathcal{P}}^{n,p}_t$ such that :

$$\tilde{\mathcal{P}}^{n,p}_t = \begin{cases} \overline{\mathcal{P}}^{n,p}_t & \text{if the } p\text{-th predictor is observed at time } t \\ \mathbb{E}[\overline{\mathcal{P}}^{n,p}_t | \mathcal{P}^{n^c}_t] & \text{if the } p\text{-th predictor is missing at time } t \end{cases} \tag{4}$$

For simplicity and because the data is consequent, we simply estimated the expectation $\mathbb{E}[\overline{\mathcal{P}}_t^{n,j}|\mathcal{P}_t^{n^c}]$ using a linear setup.

$$\mathbb{E}[\overline{\mathcal{P}}_t^{n,j}|\mathcal{P}_t^{n^c}] = \mathbb{E}[\gamma_0^{n,j} + \gamma^{n,j} \cdot \mathcal{P}_t^{n^c} + \epsilon_t^{n,j}] = \gamma_0^{n,j} + \gamma^{n,j} \cdot \mathcal{P}_t^{n^c} \tag{5}$$

where ϵ is a white noise, $\gamma_0^{n,j} \in \mathbb{R}$ and $\gamma^{n,j} \in \mathbb{R}^{|\mathcal{S}_f^n|}$, $|\mathcal{S}_f^n|$ the cardinality of $\mathcal{S}_f^n$. We find the estimator $\hat{\boldsymbol{\gamma}}^{n,j} = (\hat{\gamma}_0^{n,j}, \hat{\gamma}^{n,j})$ using the least squares setup and use it to find the corresponding missing values.

We found that this was a poor estimator as there was a lot a missing data and the patterns were very diverse. We notice that data could be missing by blocks especially in the extremities, sometimes it was fully random or even periodic. To solve these issues, we decided to turn ourselves towards a more robust and consistent data fitting technique.

A.2 Advanced data fitting

For this part, we inspire ourselves from the paper *Missing Financial Data* written by Bryzgalova and al. [9]. They propose to infer the missing values using the available cross-sectional data at time t. This time however, we use the relation between predictors from all firms to infer the missing observations. Let us recall that we write $\mathcal{P}_t^n$ the vector of predictors from stock n at time t. We introduce a new matrix $\boldsymbol{P}_t \in \mathbb{R}^{N_t \times P}$, which represents all the predictors observations at time t. We assume that the predictors matrix can be explained by a cross-sectional K factor models.

$$\boldsymbol{P}_t = \boldsymbol{F}_t \cdot \boldsymbol{\Lambda}_t^T + \boldsymbol{e}_t \tag{6}$$

where $\boldsymbol{F}_t \in \mathbb{R}^{N_t \times F}$ denotes the matrix of factors, $\boldsymbol{\Lambda}_t \in \mathbb{R}^{P \times K}$ the weights and $\boldsymbol{e}_t \in \mathbb{R}^{N_t \times P}$ the error matrix. The weights $\boldsymbol{\Lambda}_t$ can be estimated using the empirical covariance of the predictors, which describes how similar the values are to each other.

Definition 2 (***Empirical Covariance***)**.** We define the empirical covariance with missing values as

$$\hat{\boldsymbol{\Sigma}}_t \quad \text{st. } \hat{\Sigma}_t^{l,s} = \frac{1}{|O_t^{l,s}|} \sum_{n \in O_t^{l,s}} \mathcal{P}_t^{n,l} \mathcal{P}_t^{n,s} \quad \forall l,\, s \tag{7}$$

where $O_t^{l,s}$ denotes the set of firms that observe both l and s at the same time such that $|O_t^{l,s}| \leq N_t$.

The matrix $\boldsymbol{\Lambda}_t$ is defined by the scaled K-biggest eigenvectors of $\hat{\boldsymbol{\Sigma}}_t$.

$$\hat{\boldsymbol{\Lambda}}_t = \hat{\boldsymbol{V}}_t \cdot (\hat{\boldsymbol{D}}_t)^{\frac{1}{2}} \tag{8}$$

where $\hat{\boldsymbol{D}}_t \in \mathbb{R}^{K \times K}$ is the diagonal matrix of eigenvalues and $\hat{\boldsymbol{V}}_t \in \mathbb{R}^{P \times K}$ the corresponding eigenvectors from $\hat{\boldsymbol{\Sigma}}_t$.

The factors matrix can be characterized by performing a linear regression of the predictors $\boldsymbol{P}_t$ onto the weights $\hat{\boldsymbol{\Lambda}}_t$. We estimate the factor matrix using a regularized regression based only on the observed value of the predictors. Let F_t^n represents the vector of factors for the n-th stock:

$$\hat{\boldsymbol{F}}_{t,\gamma} = \begin{pmatrix} \hat{F}_{t,\gamma}^{1\,T} \\ \vdots \\ \hat{F}_{t,\gamma}^{N_t\,T} \end{pmatrix}$$

$$\hat{F}^n_{t,\gamma} = \left(\frac{1}{P} \sum_{p=1}^{P} W_t^{n,p} \hat{\Lambda}_t^p \otimes \hat{\Lambda}_t^p + \gamma I_K \right)^{-1} \left(\frac{1}{P} \sum_{p=1}^{P} W_t^{n,p} \mathcal{P}_t^{n,p} \hat{\Lambda}_t^p \right) \tag{9}$$

where $\otimes$ denotes the outer product, $\hat{\Lambda}_t^p$, the p-th row of $\hat{\mathbf{\Lambda}}_t$, (weight vector for the p-th predictor) and I_K the identity matrix $\in \mathbb{R}^{K \times K}$. The matrix coefficients $W \in \mathbb{R}^{N_t \times P}$ act as observers: the entries are 1 if the predictor p is observed for firm n at time t, otherwise it is equal to 0.

According to [10], this estimator is consistent. We also chose to add a regularisation term γ as Bryzgalova and al. [9] did in their paper. Since we are working with empirical data with many missing patterns, adding this constraint helps avoiding over-fitting or giving too much power to predictors with large weights. It also acts as a bias-variance trade-off and can help reducing the asymptotic mean squared error. The regularized estimator maintains the same consistency and convergence rate as the unregularized one when choosing the optimal γ. Specifically, it asymptotically shrinks such that $\gamma \leq \mathcal{O}(\frac{1}{\sqrt{min(P,N_t)}})$.

Definition 3 (***Cross-sectional Fitting***)**.** We introduce a new matrix $\tilde{\boldsymbol{\mathcal{P}}}_t$, which is composed of the original values with the missing entries imputed using the estimated weights and factors:

$$\hat{\mathcal{P}}_t^{n,p} = \langle \hat{F}^n_{t,\gamma}, \hat{\Lambda}_t^p \rangle \quad \overset{matrix\ form}{\Longrightarrow} \quad \hat{\boldsymbol{\mathcal{P}}}_t = \hat{\boldsymbol{F}}_{t,\gamma} \cdot \hat{\mathbf{\Lambda}}_t^T$$

$$\tilde{\boldsymbol{\mathcal{P}}}_t = W_t \odot \boldsymbol{\mathcal{P}}_t + (\mathbb{I}_{N_t \times P} - W_t) \odot \hat{\boldsymbol{\mathcal{P}}}_t \tag{10}$$

where $\mathbb{I}_{N_t \times P}$ is a matrix full of ones and $\odot$ denotes the Hadamard product.

This approach helps us infer the missing entries in the predictors matrix $\boldsymbol{\mathcal{P}}_t$, but it only considers the cross-sectional information known at time t. We could use the global model using the aggregated empirical covariance matrix. However, that would mean that we would consider future information to fill our data and hence, it would introduces a look-ahead bias. To further push our model, we will incorporate time series information from the previous cross-section. As Bryzgalova and al.[9], we assume AR(1) processes in the factors matrix $\boldsymbol{F}_t$ and in the errors $\boldsymbol{e}_t$ such that:

$$\mathcal{P}_t^{n,p} = \langle F_t^n, \Lambda_t^p \rangle + e_t^{n,p}, \quad F_t^{n,p} = \rho_{\boldsymbol{F}}^p F_{t-1}^{n,p} + \epsilon_{t,\boldsymbol{F}}^n, \quad e_t^{n,p} = \rho_{\boldsymbol{e}}^p e_{t-1}^{n,p} + \epsilon_{t,\boldsymbol{e}}^{n,p} \tag{11}$$

Using this new model, we aim to infer the missing values based on both the weighted information from time series and the cross-sectional point of view. We start by computing the residuals between the values predicted by the cross-sectional model and the past observations:

$$\hat{e}_{t-1}^{n,p} = \mathcal{P}_{t-1}^{n,p} - \langle \hat{F}^n_{t,\gamma}, \hat{\Lambda}_t^p \rangle \quad \overset{matrix\ form}{\Longrightarrow} \quad \hat{\boldsymbol{e}}_{t-1} = \tilde{W}_t \odot (\boldsymbol{P}_{t-1} - \hat{\boldsymbol{F}}_{t,\gamma} \cdot \hat{\mathbf{\Lambda}}_t^T) \tag{12}$$

where $\tilde{W}_t \in \mathbb{R}^{N_t \times P}$ is a weight matrix with entries 1 or 0 if the predictor p is observed for firm n at time $t-1$ or not respectively. The matrix $\boldsymbol{P}_{t-1}$ represents the observed or unobserved predictors for the same firms as in time t. We infer the missing values based on the weighted information between the cross-sectional and time-series data.

Definition 4 (***Cross-sectional & Time-series weight estimator***)**.** We begin by introducing a new tri-dimensional tensor, which incorporates the explanatory covariates:

$$X_t^{n,p} = \begin{pmatrix} \langle \hat{F}^n_{t,\gamma}, \hat{\Lambda}_t^p \rangle \\ \mathcal{P}_{t-1}^{n,p} \\ \hat{e}_{t-1}^{n,p} \end{pmatrix} \in \mathbb{R}^3 \quad \overset{matrix\ form}{\Longrightarrow} \quad \boldsymbol{X}_t = \begin{pmatrix} \hat{\boldsymbol{F}}_t \cdot \hat{\mathbf{\Lambda}}_t^T & \boldsymbol{P}_{t-1} & \hat{\boldsymbol{e}}_{t-1} \end{pmatrix} \in \mathbb{R}^{N_t \times P \times 3}$$

We estimate the weight coefficients β using a cross-sectional regression on the partial information available at time t. This is the estimator that Bryzgalova and al.[9] proposed:

$$\hat{\beta}_t^p = \left(\sum_{n=1}^{N_t}(W_t^{n,p} \times \tilde{W}_t)X_t^{n,p} \otimes X_t^{n,p}\right)^{-1}\left(\sum_{i=1}^{N_t}(W_t^{n,p} \times \tilde{W}_t)X_t^{n,p}\mathcal{P}_t^{n,p}\right) \tag{13}$$

Here, $(W_t^{n,p} \times \tilde{W}_t)$ equals 1 if $\mathcal{P}_t^{n,p}$ and $\mathcal{P}_{t-1}^{n,p}$ are observed, else we set it to 0.

Definition 5 (***Cross-sectional & Time-series Fitting***)**.** Finally, the missing values can be fitted using the previous estimator. Let us write $\breve{\boldsymbol{\mathcal{P}}}_t$ a new matrix composed of the original values with the missing entries imputed using the cross-sectional and time series information.

$$\bar{\mathcal{P}}_t^{n,p} = \langle X_t^{n,p}, \beta_p \rangle \quad \overset{matrix\ form}{\Longrightarrow} \quad \bar{\boldsymbol{\mathcal{P}}}_t = \begin{pmatrix} \boldsymbol{X}_t^1 \cdot \beta_1 | & \dots & |\boldsymbol{X}_t^p \cdot \beta_p| & \dots & |\boldsymbol{X}_t^P \cdot \beta_P \end{pmatrix} = \boldsymbol{X}_t \cdot \hat{\boldsymbol{\beta}}^T$$

$$\breve{\boldsymbol{\mathcal{P}}}_t = \underbrace{W_t \odot \boldsymbol{\mathcal{P}}_t}_{\text{Observations}} + \underbrace{(\mathbb{I}_{N_t \times P} - W_t) \odot (\mathbb{I}_{N_t \times P} - \tilde{W}_t) \odot \hat{\boldsymbol{\mathcal{P}}}_t}_{\text{Estimation without prior observation}} + \underbrace{(\mathbb{I}_{N_t \times P} - W_t) \odot (\tilde{W}_t) \odot \bar{\boldsymbol{\mathcal{P}}}_t}_{\text{Estimation with time series information}} \tag{14}$$

Using the factor model to impute the missing observations is similar to creating portfolios of characteristics. Essentially, the common component of a stock is a weighted average of its observed characteristics, with weights derived from the overall correlation patterns across stocks. As described by Bryzgalova and al.[9], "a factor model can be interpreted as a generalization of using an industry average for imputing missing characteristics, but instead of defining similarity arbitrarily, we learn it from the data.".

VI Modelling and Optimization

A Model selection using AIC, BIC and HQIC

In the context of model selection, researchers commonly employ metrics, such as the Akaike (AIC), the Bayesian (BIC) and the Hannan-Quinn (HQIC) Information Criteria for model comparison [11]. These information criteria have the advantage of being easy to implement. The chosen model would be the one that minimizes the information criterion.

Definition 6 (***AIC***)**.** The AIC method provides a trade-off between the logarithm of the maximized likelihood and the penalty term [12]. It can be expressed as:

$$\text{AIC} = n\log(\text{RSS}/n) + 2k \tag{15}$$

where RSS is the sum of squared residuals, k is the number of parameters (including the intercept) used in the model and n is the number of observations.

This measure of fit is used to avoid over-fitting. Indeed, penalizing the likelihood in the model allows for the selection of a simpler model. However, particularly in large samples, one of the criticisms against AIC is its lack of consistency: the method tends to favor over-parameterized models, leading to potential overfitting.

Definition 7 (***BIC***)**.** Several adjustments have been developed to address this inconsistency. BIC, developed by Schwarz (1978), considers models in terms of their posterior probability [12]. The measure is defined as:

$$\text{BIC} = n\log(\text{RSS}/n) + k\log(n) \tag{16}$$

It penalizes model complexity more strongly than AIC, as it includes a term proportional to the logarithm of the sample size. This method favors simpler models, especially as the sample size increases.

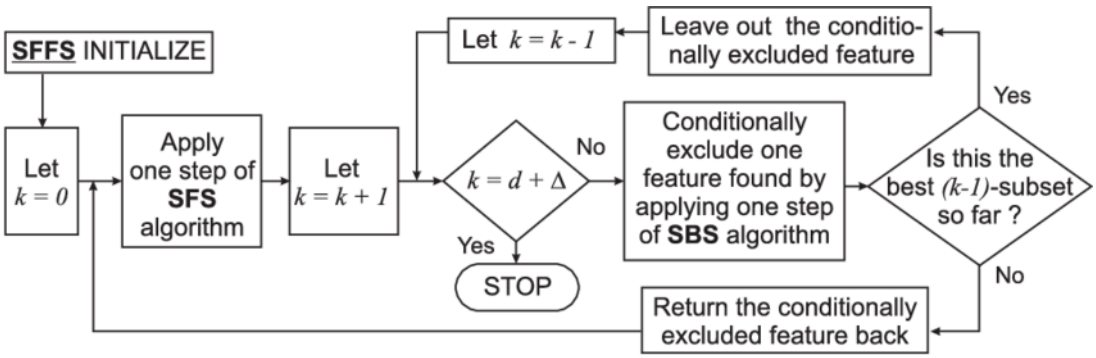

Figure 5: SFFS algorithm explained [15]

Definition 8 (***HQIC***)**.** Another consistent criterion proposed by Hannan-Quinn (1979) and Hannan (1980) is the Hannan-Quinn Information Criterion [13]. The criterion is given by:

$$\text{HQIC} = n \log(\text{RSS}/n) + 2k \log(\log(n)) \tag{17}$$

With increasing sample size, the penalty function for this method, based on the law of iterated logarithm, decreases rapidly.

However, had we tried a classical approach to AIC/BIC/HQIC model selection, using either backward or forward selection, this would have represented fitting over $100^{100} \approx \texttt{gogol}^2$ models, which is unfeasible. To avoid this dimensionality curse, we employed the Sequential Forward Floating Selection (SFFS) method, introduced by Pudil, Novovičová, and Kittler (1994, [14]). This algorithm combines a forward wrapper selection (Sequential Feature Selection — SFS) with a backward wrapper selection (Sequential Backward Selection — SBS).

The SFS method begins with an empty set of selected features and iteratively adds the feature that most improves the model until the subset reaches a user-defined size parameter, denoted as d. At this level, adding more features does not improve the model. Conversely, the SBS method works in similar way, as it evaluates whether any of the selected features can be removed without negatively impacting performance. If removing a feature does not decrease the model's performance, it is excluded from the set. It iterates the whole process until meeting d. In our case, we do not set a fixed number of predictors to be found, but try and find the best subset minimizing our criterion of choice, using `'best'`.

This algorithm is however very costly computationally speaking for a performance that is questionable, as it is purely based on the criterion loss, and does not take into account penalties. We inspect more advanced methods in the following of this report.

B Bootstrapped-enhanced Lasso and Elastic Net

1. Lasso Penalized least squares methods are widely used in machine learning and econometrics, selecting a subset of the best predictors explaining a dependent variable, as they present a good trade-off between accuracy and complexity for feature selection problems [16]. One of them is the Lasso (Least Absolute Shrinkage and Selection Operator). It consists of a linear regression plus a regularization term, that penalizes large coefficients. This means some are shrunk to zero, leading to a selection of explaining factors and reducing dimensions. In our

context of return prediction, Lasso [17] is defined as

$$\boldsymbol{\beta}(\lambda) = \arg\min_{\beta} \left\{ \sum_{\tau=\theta}^{T} \left[\mathcal{R}_t - \sum_{j=1}^{p} \beta_j \mathcal{P}_{t-1}^{j} \right]^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\} \tag{Lasso}$$

where β_j is the coefficient of predictor $\mathcal{P}^j$ and λ is the regularization hyperparameter, which represents the balance between accuracy and complexity. Lasso's penalty has the advantage of leading to a convex optimization problem, making it computationally more efficient relative to other selections techniques such as pure AIC/BIC criteria presented before. In addition, Lasso preserves results' interpretability, unlike Principal Component Analysis (PCA) for example.

We decided to implement BE-Lasso, a Bootstrapped-Enhanced Lasso method presented by Bunea et al. [16]. This algorithm addresses one of the main issues of the Lasso, that is that the model's quality is heavily influenced by the choice of the penalization parameter λ. Although a single cross-validation can be used to determine λ, it may not always result in the best choice, as it depends on specific sub-samples of the dataset. Therefore, BE-Lasso augments the cross-validation selection of tuning hyperparameters with bootstrapping. The main idea is to first use cross-validation to select the optimal penalization parameter from a predefined list. For each value appearing in this list, use Lasso to select features, then regress a linear model with OLS on this set of predictors[3]. No further details are given in our work, but tey can be found in Efron et al. (2004, [18]). Choose the parameter that results in the smallest error. Then, use bootstrapping to simulate a large number of samples and observe the frequency of selection for each parameter. The detailed algorithm can be found in Bunea et al. (pp.1522, [16]).

Alternatively, Zou et al. (2007, [19]) present a different method for selecting λ, relying on the Stein's Unbiased Risk Estimator (SURE) framework. The authors show the number of non-zero parameters from the Lasso regression is "an exact unbiased estimate of the degrees of freedom of the regression", and can be used to determine the optimal value of the penalization parameter. Assuming both the features matrix and the dependent variables are normally distributed, Efron (2004, [18]) shows the Stein's unbiased risk estimator is given by :

$$df(\hat{\mu}_{\mathcal{R}}^{\lambda}) = \sum_{i=1}^{\sum_{\tau=\theta}^{T} N_\tau} \frac{\text{cov}(\mathcal{R}_i, \hat{\mu}_{\mathcal{R},i}^{\lambda})}{\sigma^2} \tag{18}$$

where $\hat{\mu}_{\mathcal{R},i}^{\lambda}$ is the i^{th} entry of $\hat{\mu}_{\mathcal{R}}^{\lambda}$, the vector of predicted value of the lagged stock return of a Lasso with regularization parameter λ, and $\mathcal{R}_i$ is the i^{th} entry of $\mathcal{R}$, the vector of stock returns. Zou et al. show that $\hat{df}(\lambda)$, the estimator of Eq. (18), is unbiased (Theorem 1, [19]) and consistent (Theorem 2, [19]). Using this estimator, they propose an unbiased formula for the prediction error of a model and generalize it to AIC and BIC formulas. In our stock return prediction context, the general formula is given by:

$$C_p(\hat{\mu}_{\mathcal{R}}^{\lambda}) = \frac{\|\mathcal{R} - \hat{\mu}_{\mathcal{R}}^{\lambda}\|^2}{\sum_{\tau=\theta}^{T} N_\tau} + \frac{w}{\sum_{\tau=\theta}^{T} N_\tau} \hat{df}(\hat{\mu}_{\mathcal{R}}^{\lambda}) \sigma^2 \tag{19}$$

Note that both $\hat{\mu}_{\mathcal{R}}^{\lambda}$ and $\mathcal{R}$ are of size $\sum_{\tau=\theta}^{T} N_\tau$: we are concerned with the full number of observations N_τ at each date τ between θ and T. Here, w is a weight that defines $C_p(\hat{\mu}_{\mathcal{R}}^{\lambda})$ as

[3]The regression of the selected predictors using OLS (rather than simply using the error from the Lasso regression) is intended to reduce bias [16].

being AIC when $w = 2$ and BIC when $w = \sum_{\tau=\theta}^{T} N_\tau$. Theorem 3 [19] states the optimal value of the penalization parameter can be obtained as:

$$\lambda^* = \arg\min_{\lambda} C_p(\hat{\mu}^{\lambda}_{\mathcal{R}}) \tag{20}$$

Combining these two different approaches, we present a bootstrapped-enhanced Lasso method relying on a determination of λ, itself relying on the convex optimization from Eq. 20. Our goal is to lever both the power of bootstrapping to provide robust results and the unbiased estimation of the regularization parameter presented by Zou et al., while preserving a relatively low computational cost. The proposed method is as follows:

1. Normalize the set of predictors $\tilde{\mathcal{P}}$ and lagged returns $\tilde{\mathcal{R}}$ (zero mean and unit variance).
2. Select a range of values for the regularization parameter $\Lambda = [\lambda_1, ..., \lambda_D]$
3. For each value λ_i in Λ:
   (a) Fit a Lasso model: $L_i = \text{Lasso}(\mathcal{R}_t, \tilde{\mathcal{P}}_{t-1}; \lambda_i)$.
   (b) Compute the prediction error (Eq. 19) based on the degrees of freedom (Eq. 18).
   (c) Select the Lasso model L_i leading to the smallest prediction error.
   (d) Report non-zero coefficient features.
4. Use bootstrapping technique to generate B random samples of size b from the data. Perform B times step 3, report the frequency of selection of each parameters.
5. Select features with frequency larger than threshold ϑ.

2. Elastic Net An interesting augmentation of the traditional Lasso regression for tackling correlated features is the Elastic Net method. It is also a penalized least square method, with an extra quadratic penalization term, as presented in **??**. In presence of groups of highly pairwise-correlated features, Elastic Net is more robust than Lasso [20].

$$\bar{\boldsymbol{\beta}}(\lambda, \mu) = \arg\min_{\bar{\beta}} \left\{ \sum_{\tau=\theta}^{T} \left[\mathcal{R}_t - \sum_{j=1}^{p} \bar{\beta}_j \mathcal{P}^j_{t-1} \right]^2 + \lambda \sum_{j=1}^{p} |\bar{\beta}_j| + \mu \sum_{j=1}^{p} (\bar{\beta}_j)^2 \right\} \tag{ENET}$$

Based on this new model, we use the same methodology as in the Lasso case, adding an extra dimension to the grid search for the quadratic penalization parameter μ.

4. Limitation The major concern about using this technique is the fact that the method presented by Zou et al. [19] relies on the assumption that both features and dependent variables are normally distributed. To address this, we standardize both set of variables, to make them zero-mean and unit variance. The resulting vector of dependent variables has a distribution relatively close to a standard normal, as per Figure 6. This leads to zero-centered distributions but the normality is more than controversial.

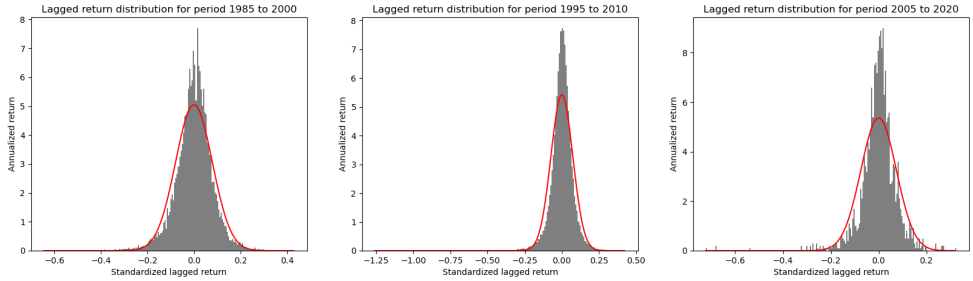

Figure 6: Distribution the scaled lagged return for each of the three periods (black) and standard-normal density (red)

One way to tackle this could be to use copula based transformations. In the context of an EPFL course[4], we presented a method to transform random variables that are correlated into the independent and identically, normally distributed variables, using copulas.

Theorem 1. *Let X and Y be two (dependent) random variables. Then the variables*

$$Z_X = \Phi^{-1}(F_X(X)) \qquad Z_Y = \Phi^{-1}(F_Y(Y))$$

are approximately independent standard normal variables, where Φ in the standard Gaussian cdf. This follows from the fact that the marginals $F_X(X)$ and $F_Y(Y)$ are uniform over $[0,1]$ and that the function Φ^{-1} is continuous and strictly increasing.

In a further development, this method could be implemented to assess the performance of our bootstrapping method.

C Nonparametric Estimation with Adaptive Group Lasso

As Jian Huang, Joel L. Horowitz, Fengrong Wei stated in their paper *Variable selection in nonparametric additive models* (2010), "(...) there is little a priori justification for assuming that the effects of covariates take a linear form or belong to any other known, finite-dimensional parametric family"[21]. Based on their insight, we will try to extend our model to include such cases.

Definition 9 (***Nonparametric Estimation***)**.** We assume that the returns can be estimated by a nonparametric additive model defined as:

$$\mathcal{R}_t^n = \sum_{p=1}^{P} f_t^j(\mathcal{P}_{t-1}^{n,p}) + \gamma_t^n \tag{21}$$

where the functions $f_t^j(\cdot)$ are unknown and γ_t^n are unobserved white noises.

In the paper by Huang and al.[21], the function and the random variables $\mathcal{P}_{t-1}^{n,j}$ have to satisfy some conditions, notably the set on which they exist and especially the partition they have to fulfill. We will transform our data, using a transformation proposed by Joachim Freyberger,

[4]Project *Advanced Portfolio Construction*, May 2024. Financial applications of blockchains and distributed ledgers, FIN-413 [link].

Andreas Neuhierl, Michael Weber in their original paper *Dissecting Characteristics Nonparametrically* [4]:

$$\tilde{\mathcal{P}}_t^n = F_t(\mathcal{P}_t^n) = \begin{pmatrix} F_t^1(\mathcal{P}_t^{n,1}) \\ \vdots \\ F_t^j(\mathcal{P}_t^{n,j}) \\ \vdots \\ F_t^p(\mathcal{P}_t^{n,p}) \end{pmatrix} \tag{22}$$

Definition 10 (***Rank Transformation***)**.** The rank normalization takes care of outliers and accomplishes stationarity throughout the cross-sectional data and over time:

$$\tilde{\mathcal{P}}_t^{n,j} = F_t^j(\mathcal{P}_t^{n,j}) = \frac{rank(\mathcal{P}_t^{n,j})}{N_t + 1} \tag{23}$$

This method ensures that the cross-sectional distribution of characteristics lies within the unit interval $[0, 1]$. This transformation is in agreement with the fact that our data is cross-sectional as some predictors do not have the same influence depending on the timeline.

Moreover, authors showed empirical evidence suggesting that the transformed estimator $\tilde{\mathcal{P}}_t^n$ tends to yield better out-of-sample predictions compared to the original estimator $\mathcal{P}_t^n$. This improvement is attributed to the rank transformation's robustness to outliers. We denote $\tilde{\mathcal{P}}_t^i$ the transformed predictor and $\tilde{f}_t^j(\cdot)$ the new unknown function:

$$\mathcal{R}_t^n = \sum_{j=1}^{p} \tilde{f}_t^j(\tilde{\mathcal{P}}_{t-1}^{n,j}) + \epsilon_t^n \tag{24}$$

According to Schumaker [22], the functions $\tilde{f}_t^j(\cdot)$ can be approximated by normalized B-spline. Furthermore, J. Freyberger, A. Neuhierl and M. Weber "propose to estimate quadratic functions over parts of the normalized characteristic distribution"[4]. We obtain the following approximation:

$$\tilde{f}_t^j(\tilde{\mathcal{P}}_{t-1}^{n,j}) \approx \sum_{k=1}^{L+2} \beta_t^{j,k} b_k(\tilde{\mathcal{P}}_{t-1}^{n,j}) \tag{25}$$

The number of intervals L controls the level of flexibility and smoothness of the estimation. These intervals serve as the basis for approximation using a quadratic spline function over each interval. This method, as described by the authors [4], effectively captures potentially nonlinear relationships between the transformed characteristics and expected returns. The functions b_i, $i = 1, \cdots, L + 2$ are called basis functions and are defined as:

$$b_1(x) = 1, \quad b_2(x) = x, \quad b_3(x) = x^2, \quad b_k(x) = max(x - s_{k-3}) \tag{26}$$

where s denotes the number of nodes in the intervals.

Now that we have built our model, all that is left to do is find the predictors that best predict future returns. We will employ the adaptive group Lasso method proposed by Freyberger and al.[4].

Definition 11 (***Adaptive Group Lasso***)**.** This method is called adaptive as it runs two group Lasso regressions one after the other to further shrink the subset of predictors. We begin this process by estimating the first group Lasso using the following minimization problem:

$$\tilde{\boldsymbol{\beta}}_t = \underset{\beta_t^{j,k}: j=1,\ldots,p; k=1,\ldots,L+2}{\arg\min} \sum_{i=1}^{N} \left(\mathcal{R}_{it} - \sum_{j=1}^{p} \sum_{k=1}^{L+2} \beta_t^{j,k} b_k(\tilde{\mathcal{P}}_{t-1}^{n,j}) \right)^2 + \lambda_1 \sum_{j=1}^{p} \left(\sum_{k=1}^{L+2} \beta_t^{j,k^2} \right)^{\frac{1}{2}} \tag{27}$$

Once the first set of estimators are chosen, we launch the second phase of group Lasso and exclude the predictors that were already excluded using the following weights:

$$w_s = \begin{cases} \left(\sum_{k=1}^{L+2} \tilde{\beta}_t^{j,k^2}\right)^{-\frac{1}{2}} & \text{if } \sum_{k=1}^{L+2} \tilde{\beta}_t^{j,k^2} \neq 0 \\ \infty & \text{if } \sum_{k=1}^{L+2} \tilde{\beta}_t^{j,k^2} \approx 0 \end{cases} \tag{28}$$

The weights close to zero are penalized infinitely to enforce the $\beta = 0$ constraint. The other predictors are scaled inversely proportionally by their coefficient estimation in the first regression. The second objective function follows:

$$\breve{\boldsymbol{\beta}}_t = \underset{\tilde{\beta}_t^{j,k}: j=1,\dots,p; k=1,\dots,L+2}{\arg\min} \sum_{i=1}^{N} \left(R_{it} - \sum_{j=1}^{p} \sum_{k=1}^{L+2} \tilde{\beta}_t^{j,k} b_k(\tilde{\mathcal{P}}_{t-1}^{n,j}) \right)^2 + \lambda_2 \sum_{j=1}^{p} \left(w_s \sum_{k=1}^{L+2} \tilde{\beta}_t^{j,k^2} \right)^{\frac{1}{2}} \tag{29}$$

This final estimator should have selected the predictors that are the most likely to generate the returns based on a nonparametric motion.

This method has many advantages as it can delete more predictors and more specifically, it integrates non-polynomial functions in the regression. It has however a few major drawbacks. First of all, the model is hard to calibrate. Since it runs a double minimization, the hyperparameters are extremely important and they are dependent between the two objective functions. Indeed, if λ_1 is too large then the weighting function attributes huge weights to the group Lasso part and if λ_2 is not small enough the function struggles to converge and simply assigns all weights to 0. Another issue is the initialization when running the minimization: the outcome highly depends on where the algorithm started.

As results vary widely when calibrating hyperparameters, no explicit results are presented in this report, as no clear-cut dimension goal was set. The sensitivity the method achieves is up to the reader.

We offer two methods to solve these issues, the first one being to use the iterative algorithm proposed by Ming Yuan and Yi Lin in their article *Model Selection and Estimation in Regression with Grouped Variables* (2005)[23]. It has however downsides as it scales badly with the number of predictors and the regularization coefficients would still need to be determined.
Another solution would be to use the recent algorithm proposed by Qingyuan Zhang and Hien Duy Nguyen in *Consistent information criteria for regularized regression and loss-based learning problems* (2024)[24]. The algorithm treats the problem as a duality in optimization between the estimator β and the regularization term λ.

VII Results

In this section, we present the results from our study, comparing the optimal predictors identified by five methods over three time periods: 1985 - 2000, 1995 - 2010, and 2005 - 2020.

The Lasso is trained using $\Lambda = [0.0001, 0.0005, 0.001, 0.005, 0.01, 0.1]$ as tuning parameters, with a maximum of 100'000 iterations when performing the optimization. The random seed is set to 42. The bootstrapped enhancement is performed on 50 samples. For the tuning of the Elastic-Net models, we use the same parameters with an initial set of `l1_ratio` $= [0.05, 0.1, 0.5, 0.95]$, which corresponds to the ratio between the two penalization terms.

	AIC	BIC	HQIC	Initial	BE-LASSO	BE-Enet	Initial
1985 - 2000	18	4	6	47	11	12	61
1995 - 2010	28	7	17	60	3	4	63
2005 - 2020	34	13	22	74	10	14	80

Table 3: Comparison of optimal predictor for each method
The results presented for AIC, BIC and HQIC were found with the simple fitting of the data.

In Figure 7, we present the selected features for BE-Lasso and BE-Elastic net in the different periods. The x-axis corresponds to all the different selected predictors regardless of the model, and each line of the y-axis is a method for a time period. For a given period-method pair if the predictor is in dark, then it has been selected. We clearly observe similarities in the patterns presented, but ENet selects more predictors than Lasso.

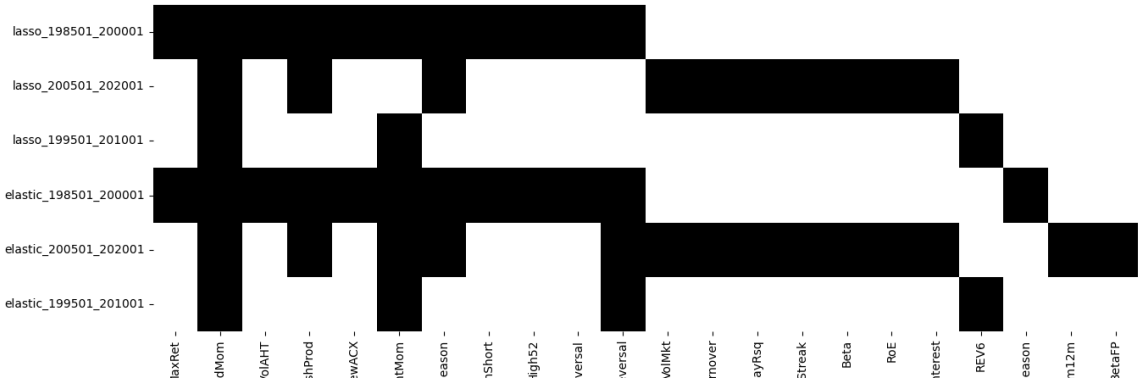

Figure 7: Feature selection with BE-Lasso and BE-ENet

As we can see in Figure 7, depending on the time period the ouputs vary widely. Also, the predictors selected by BE-Lasso and BE-ENet are closely related, which underlines the consistency between the 2 methods. However, during the 1995 - 2010 period, the number of predictors is quite low. It could be interesting to pursue further analyses in that direction, especially in terms of parameters tuning.

Finally, using SFFS, BE-Lasso and BE-ENet, we have successfully retrieved a subset a predictors which best predict the returns based on the loss function from III.

VIII Predicting

To lever the selection of the predictors we found, we introduce our Financial Learning Algorithm for Signal Heuristics (FLASH), a deep neural network designed to predict trading signals, indicating selling or buying strategies. FLASH is 8 layers depth, alternating between `Linear` (that performs a simple linear transformation) and `LeakyReLU` activations defined in 30. We also add some `Dropout` with probability $p = 0.30$ to reduce the risk of over-fitting. A simplified representation of the model's architecture presented in Figure 8.

$$\text{LeakyReLU}(x;\, \text{s}) = \max(0, x) + \text{s} \cdot \min(0, x) \tag{30}$$

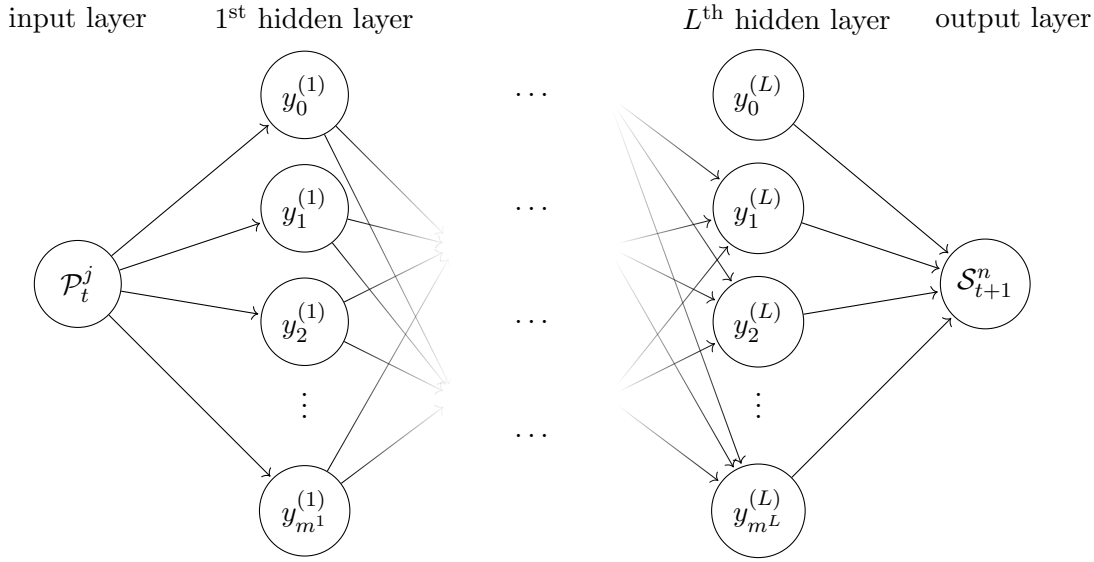

Figure 8: FLASH neural network

The data consists of observations of the values of P predictors of X companies over a period of T months. At each period $t \leq T$, the predictors also contain the current return $\mathcal{R}_t$. The dependent variable is $\mathcal{S}_{t+1}$, defined as:

$$\mathcal{S}_{t+1} = \begin{cases} 1 & \text{if } \mathcal{R}_t \geq 0 \\ 0 & \text{if } \mathcal{R}_t \leq 0 \end{cases} \tag{31}$$

FLASH is trained on the last time period (2005-2020), on scaled features. We use 80% of the data for training the model and the remaining 20% for in sample testing. The model is trained on 70 epochs with cosine learning rate scheduler starting at 5^{-10} and a warm-up of 50 steps. This scheduler is beneficial in order to adapt the speed of learning of the model throughout the process, to avoid getting stuck in a local minimum while boosting learning effectiveness.
We use Binary Cross Entropy Loss with Adam optimizer, as their use is adapted for binary classification. The accuracy and Cohen scores are also reported in Figure 9.

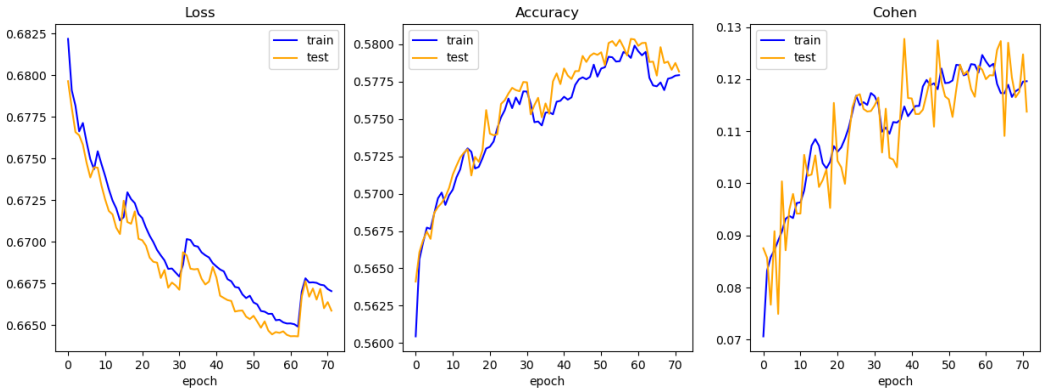

Figure 9: FLASH training on period 2005 - 2020

Using our trained FLASH model, we evaluate them on the validation sample. This consists in

data over a period from January 2020 to June 2022. This is intended to evaluate the prediction of FLASH on data that it never saw and therefore give more faithful results. Out of sample, we get an accuracy of 0.5296 and a Cohen score of 0.00468.

The performance of the model is promising in sample, as the neural network seems to learn well and fast. Despite the performance of the out-sample data being lower, the time period is two years ahead, which is huge. The optimal would be to update the knowledge as soon as the predictors are available, thus proceeding in rolling windows.

IX Conclusion

Identifying the most significant predictors for predicting financial returns was the main goal of our study. By combining traditional and advanced selection methods, alongside meticulous data extraction and cleaning procedures, we aim to achieve this while ensuring dataset reliability and accuracy.

We conduct a deep analysis of the predictors using the Open Asset Pricing dataset by Chen and Zimmermann, ensuring careful handling of the dataset, which contained over 56% `NaN`s values. Advanced techniques featured bootstrap-enhanced Lasso and Elastic Net, which are particularly proficient at handling high-dimensional data and multicollinearity. Furthermore, the implementation of a nonparametric approach, as outlined in *Dissecting Characteristics Nonparametrically* [4] highlights the independent contributions of different characteristics to expected returns.

Our findings reaffirm the importance of combining traditional and advanced statistical techniques to improve the accuracy of return forecasting models. The traditional methods establish a foundation, while the advanced techniques uncovered more complex relationships within the data. These comprehensive approaches allows for a deeper understanding of return variability, facilitating effective risk management and portfolio performance optimization for investors.

Our study presents a robust framework for identifying significant predictors. Future research could build upon our findings by applying these methodologies to different market conditions and to additional financial datasets.

References

[1] E. F. Fama and K. R. French, "The capital asset pricing model: Theory and evidence," *Journal of Economic Perspectives*, pp. 25–46, 2004. [Online]. Available: https://mba.tuck.dartmouth.edu/bespeneckbo/default/AFA611-Eckbo%20web%20site/AFA611-S6B-FamaFrench-CAPM-JEP04.pdf.

[2] A. Y. Chen and T. Zimmermann, "Open source cross-sectional asset pricing," *Critical Finance Review*, vol. 27, no. 2, pp. 207–264, 2022. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3604626.

[3] A. N. Joachim Freyberger Björn Höppner and M. Weber, "Missing data in asset pricing panels," National Bureau of Economic Research, Working Paper 30761, Dec. 2022. DOI: 10.3386/w30761. [Online]. Available: http://www.nber.org/papers/w30761.

[4] A. N. Joachim Freyberger and M. Weber, "Dissecting characteristics nonparametrically," National Bureau of Economic Research, Working Paper 23227, Mar. 2017. DOI: 10.3386/w23227. [Online]. Available: http://www.nber.org/papers/w23227.

[5] A. Y. Chen and T. Zimmermann, "Open source asset pricing," Tech. Rep., Aug. 2023. [Online]. Available: https://www.openassetpricing.com/data/.

[6] "Center for research in security prices." (2024), [Online]. Available: https://wrds-www.wharton.upenn.edu/pages/about/data-vendors/center-for-research-in-security-prices-crsp/.

[7] A. Y. Chen. "Crosssectiondemos." (2023), [Online]. Available: https://github.com/OpenSourceAP/CrossSectionDemos/tree/main.

[8] T. Shumway, "The delisting bias in crsp data," Mar. 1997. [Online]. Available: https://doi.org/10.1111/j.1540-6261.1997.tb03818.x.

[9] M. L. Svetlana Bryzgalova Sven Lerner and M. Pelger, "Missing financial data," May 2022. [Online]. Available: https://ssrn.com/abstract=4106794.

[10] M. P. Junting Duan and R. Xiong, "Target pca: Transfer learning large dimensional panel data," *Journal of Econometrics*, p. 105 521, 2023, ISSN: 0304-4076. DOI: https://doi.org/10.1016/j.jeconom.2023.105521. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304407623002373.

[11] Y. Tao and J. Yu, "Model selection for explosive models," Mar. 2017. [Online]. Available: http://dx.doi.org/10.2139/ssrn.2933621.

[12] A. Kumar, "Aic & bic for selecting regression models: Formula, examples," Nov. 2023. [Online]. Available: https://vitalflux.com/aic-vs-bic-for-regression-models-formula-examples/#AIC_BIC_Concepts_Explained_with_Formula.

[13] H. J. Bierens, "Information criteria and model selection," Aug. 2004. [Online]. Available: https://faculty.wcas.northwestern.edu/lchrist/course/assignment2/INFORMATIONCRIT.pdf.

[14] J. N. P. Pudil and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letters*, vol. 15, no. 11, pp. 1119–1125, 1994, ISSN: 0167-8655. DOI: https://doi.org/10.1016/0167-8655(94)90127-9. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0167865594901279.

[15] J. P. Somol J. Novovicova and P. Pudil, "Sequential forward floating selection algorithm," *Efficient Feature Subset Selection and Subset Size Optimization*, 2010. [Online]. Available: https://www.researchgate.net/figure/Sequential-Forward-Floating-Selection-Algorithm_fig1_221907632.

[16] F. Bunea, Y. She, H. Ombao, A. Gongvatana, K. Devlin, and R. Cohen, "Penalized least squares regression methods and applications to neuroimaging," *NeuroImage*, vol. 55, pp. 1519–27, Dec. 2010. DOI: 10.1016/j.neuroimage.2010.12.028.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996, ISSN: 00359246. [Online]. Available: http://www.jstor.org/stable/2346178 (visited on 05/31/2024).

[18] I. J. Bradley Efron Trevor Hastie and R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004. DOI: 10.1214/009053604000000067. [Online]. Available: https://doi.org/10.1214/009053604000000067.

[19] T. H. Hui Zou and R. Tibshirani, "On the "degrees of freedom" of the lasso," *The Annals of Statistics*, vol. 35, no. 5, pp. 2173–2192, 2007. DOI: 10.1214/009053607000000127. [Online]. Available: https://doi.org/10.1214/009053607000000127.

[20] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 67, no. 2, pp. 301–320, Mar. 2005, ISSN: 1369-7412. DOI: 10.1111/j.1467-9868.2005.00503.x. eprint: https://academic.oup.com/jrsssb/article-pdf/67/2/301/49795094/jrsssb_67_2_301.pdf. [Online]. Available: https://doi.org/10.1111/j.1467-9868.2005.00503.x.

[21] J. L. H. Jian Huang and F. Wei, "Variable selection in nonparametric additive models," *Annals of Statistics*, vol. 38, no. 4, pp. 2282–2313, 2010, Published in the Annals of Statistics by the Institute of Mathematical Statistics, ISSN: 0090-5364. DOI: 10.1214/09-AOS781. arXiv: 1010.4115 [math.ST]. [Online]. Available: https://doi.org/10.48550/arXiv.1010.4115.

[22] L. Schumaker, *Spline Functions: Basic Theory* (Cambridge Mathematical Library), 3rd. Cambridge University Press, 2007. DOI: 10.1017/CBO9780511618994. [Online]. Available: https://doi.org/10.1017/CBO9780511618994.

[23] M. Y. Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 68, no. 1, pp. 49–67, Dec. 2005, ISSN: 1369-7412. DOI: 10.1111/j.1467-9868.2005.00532.x. eprint: https://academic.oup.com/jrsssb/article-pdf/68/1/49/49794691/jrsssb_68_1_49.pdf. [Online]. Available: https://doi.org/10.1111/j.1467-9868.2005.00532.x.

[24] Q. Zhang and H. D. Nguyen, *Consistent information criteria for regularized regression and loss-based learning problems*, 2024. arXiv: 2404.17181 [stat.ME].