RISK ESTIMATION UNDER SIGNATURE GENETIC ALGORITHM

A PREPRINT

Ramy Sukarieh *
Economics Department
Paris 1 Panthéon-Sorbonne University
FR, Paris 12 Pl. du Panthéon
ecoramy@gmail.com

Raphael Douady
Economics Department
Paris 1 Panthéon-Sorbonne University
FR, Paris 12 Pl. du Panthéon
rdouady@gmail.com

Eric Meltzer
Development Department
EyeLand Capital Management
USA, Brooklyn 188 Berkeley Pl.
eric lowerdash meltzer@hotmail.com

February 7, 2025

ABSTRACT

Given a large set of explanatory variables, we apply a signature-genetic algorithm ('SigGA') approach to reduce the dimension of the supervised learning algorithm to a handful of variables. Then, we rank stocks and construct investment portfolios using polymodels [1], a collection of nonlinear equations on selected risk factors that have persistently driven much of stock returns' dependencies. From a practical standpoint, the SigGA helps capture volatility clustering in stocks, and polymodels are well-suited for constructing tail-concentrated hedged portfolios, especially when markets are under stress. Irrespective of the underlying models used for dimension reduction and variables mapping and their subsequent ranking, our portfolio construction methodology consists of five building blocks drawn from personal observations and years of investment management experience (fundamental and quantitative). We think qualitative or quantitative portfolio construction and stock investing will eventually necessitate portfolio managers to *order*, *segregate*, *integrate*, *condition*, *exclude*, and *concentrate* (position sizing) their investment ideas, stock selection, and stock dependencies i.e.: factors. Additionally, we use polymodels to calibrate portfolio exposures to different frequencies taking into account nonlinearity. We propose a new definition of the factor risk that takes into account multiple frequencies, lead-lag relationships, and nonlinearity with stocks from a theoretical perspective.

***K*eywords** First keyword · Second keyword · More

1 Introduction

The data sets we use capture two major sources of risks, company-specific unique to each stock, and company non-specific common to all stocks (economic, financial, and market). The former approximately 50 risk factors per stock come from the income statement, balance sheet, and cash flows of the S&P 500 companies' quarterly reporting. The latter around 109 indicators available in monthly or quarterly frequency, cover a spectrum of U.S. economic and financial conditions such as the consumer price index, gross domestic product, and the Chicago Fed National Activity Index. Company-specific fundamental risk factors like liabilities, investments, and inventories are expressed in levels or values ranging from 10^1 up to 10^6. Others are ratio-based, such as price-to-equity and price-to-book. Similarly, stock prices can range from 1 to 10^3, etc.

*PhD candidate information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

2 Dimension Reduction: Signature Genetic Algorithm

Given a large number (> 300) of explanatory variables, we want to model each stock using up to seven or so variables. Over the years, academics have extensively researched dimension reduction techniques and provided a wealth of literature to choose from. The application of signature in time series [2] as feature extraction and selection techniques has recently gained traction among machine learning practitioners and academics. Concurrently, the biologically inspired genetic algorithm has been widely popular in applying artificial intelligence to solve stochastic global optimization problems. However, to this date, we haven't seen publications combining signatures with genetic algorithms, which can potentially solve problems like dimension reduction. To this end, we incorporate signatures within the genetic algorithm framework, which we call SigGA, to achieve dimension reduction. The SigGA is interesting because it is an obvious choice, given the random nature of our selection problem, the random selection property of the genetic algorithm optimizer, and the nonlinear capability of signature transform. For variable selection problems where the input matrix M consists of several hundred series of times, the open question is which members of M we shall select and convert to continuous paths for the signature calculation. As paths increase in dimensional size the complexity of disentangling signature coefficients from the underlying factors increases. Setting up the genetic algorithm requires an appropriate objective (fitness) function to evaluate the random choices generated by the model after their signatures have already been computed. Hence, the SigGA process involves randomly generating thousands of optimized (because of GA) copies of paths X, each containing any number of elements of M between 4 and 15, depending on the configuration desired by the modeler. Then, compute the signatures evaluated through the random forest fitness function. We will describe each model separately before diving into SigGA.

2.1 The Path of the Signature

Signatures provide a method of transforming a sequence of data points called a stream or a path into a series of features that capture the analytical and geometric properties of the path, which can be highly irregular, yet continuous. The signatures do not retain all the specific details of the path such as the exact values at each time step but rather capture a summary or a compressed version of it, retaining essential structure such as the order of variations and the cumulative effects of changes in the path. Since the path can have multiple interacting variables (for example, in a multi-dimensional time series), the signature captures relationships between these variables at different levels of complexity. Signatures are typically computed on datasets that are inherently ordered, or sequential, such as time series or DNA. They can be thought of as creating a unique representation of a path (or paths). Moreover, the signature can handle data of varying lengths, irregular and non-stationary [3]. The definition of the signatures relies on mathematical concepts of paths, tensor algebra, and integration along paths.

2.1.1 Definition of a Path

A path is a smooth line, a continuous curve without sharp angles, such as sequences that connect two or multiple points in a given space. The paths we consider are piecewise differentiable, where the domain is divided into a finite number of intervals, and each interval has a well-defined derivative but may not be differentiable at the points where the intervals meet. The paths are also smooth, which means that derivatives exist for all orders [4]. A path is parameterized by time or another variable and can be seen as a trajectory of points in space. Interestingly, paths may not be differentiable or even continuous within the context of rough path theory, but their signatures can still be defined. Therefore, a path is a continuous function that maps a closed interval of real numbers onto d-dimensional Euclidean space. To compute the signatures paths must be of bounded variation or have finite length [2]. Paths with finite length, hence bounded variation $BV(\mathbb{R}^d)$ capture the concept of an ordered evolution of events [5].

Definition 2.1. (Path) A path is a continuous map $X : [0, 1] \to \mathbb{R}^d$, denoted $X_t = X(t)$ parametrized by time $t \in [0, 1]$ [6] A multi-dimensional path, $\forall\ d \geq 1,\ X : [a, b] \to \mathbb{R}^d,\ X_t = \{X_t^1, X_t^2, X_t^3 \ldots, X_t^d\}$ is the coordinate path residing in d-dimensional real-valued space (number of columns).

Definition 2.2. (Bounded Variation) $X : [0, 1] \to \mathbb{R}^d$ be a continuous function. The total variation of X on an interval $[s, t] \subset [0, 1]$ is the quantity I defined by:

$$I_s^t(X) = \sup_{(t_0,\ldots,t_k)\in D_{s,t}} \Big\{ \sum_{i=1}^{k} |X_{t_i} - X_{t_{i-1}}| \Big\} \tag{1}$$

$D_{s,t}$ denotes the set of all finite partitions of $[s, t]$, $D_{s,t} = \{(t_0, ..., t_k) \,|\, k \geq 0,\ s = t_0 < t_1 < ... < t_{k-1} < t_k = t\}$[2].

Lemma 2.1. *$I_s^t(X) < \infty, X \in BV(\mathbb{R}^d), X$ is continuously differentiable and is Riemann-integrable with derivative $X' : [0, 1] \to \mathbb{R}^d$, its total variation is the vertical component of the arc length of the path of X:*

$$I_s^t(X) = \int_0^1 |X_t'|dt \tag{2}$$

Definition 2.3. (Path Integral) Consider two paths $Y : [s,t] \to \mathbb{R}, X : [s,t] \to \mathbb{R},\ [s,t] \subset [0,1]$. Let $(t_0^{(n)}, \ldots, t_n^{(n)}) \in D_{s,t}$ be a partition $[s,t]$ of length $n\ \forall n \geq 0, (s_1^{(n)}, \ldots, s_n^{(n)})$ be a sequence $\exists\ \forall i \in \{1, \ldots, n\}, s_i^{(n)} \in [t_{i-1}^{(n)}, t_i^{(n)}]$. If the sum on the right converges to the limit I regardless of partition choice, then the Riemann-Stieljes integral of Y (integrand) against X (integrator, measure function) exists:

$$I_s^t(X) \xrightarrow{d} \sum_{i=1}^{n} Y_{s_i^{(n)}}(X_{t_i^{(n)}} - X_{t_{i-1}^{(n)}}) \tag{3}$$

Given two functions Y and X, the Riemann-Stieltjes integral exists if Y is continuous and X has bounded variations on $[s,t]$. As the picture in 1 shows, the integral sums the areas under the curve Y_t with respect to changes in X_t. This means X_t is a step function and the integral sums Y_t values at points where X_t jumps. Take $X_t' = dX_t/dt$, the general form of the path integral of Y_t against X_t is 4. When $Y_t = 1$, the path integral is the increment of X. When $X_t = t$, the path integral is the areas in 1. The path integral of vector-valued X, Y is 7.

$$I_s^t(X_t) = \int_s^t Y_t dX_t = \int_s^t Y_t X_t' dt \tag{4}$$

- $Y_t = 1\ \forall\ t \in [s,t] \implies I_s^t(X_t) = \int_s^t dX_t = \int_s^t X_t' dt = X_t - X_s$ (5)
- $X_t = t\ \forall\ t \in [s,t] \implies I_s^t(X_t) = \int_s^t Y_t dX_t = \int_s^t Y_t dt$ (6)
- $X, Y : [0,1] \to \mathbb{R} \implies I_s^t(X_t) = \int_s^t Y_t dX_t = \begin{pmatrix} \int_s^t Y_t^1 dX_t^1 \\ \vdots \\ \int_s^t Y_t^d dX_t^d \end{pmatrix}$ (7)

Example 1. $d = 1$. Consider the path $X_\tau = \tau \to X_\tau' = 1$ time re-parameterized to $\tau \in [0,7]$ with a slope equal to 1. The path integral of Y_τ against X_τ is given by the areas under (green) and above (red) the curve and zero, obtained by plotting Y_τ against $\tau, \forall\ \tau \in [a,b]$, where $Y_\tau = \pi_\tau$ is the food [2] inflation indicator figure 1:

$$I_\tau = \int_a^b Y_\tau X_\tau' d\tau = \int_a^b Y_\tau d\tau \tag{8}$$

[2]Source: Nasdaq (previously quandl) quarterly food inflation FINF from 12/2004 to 03/2021, 66 observations

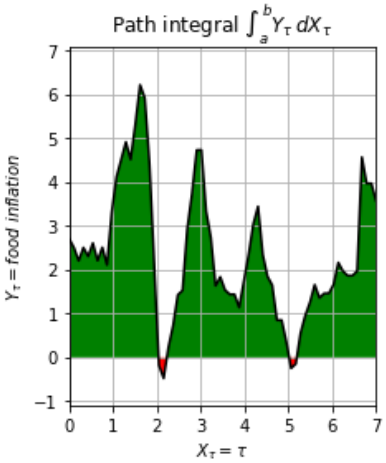


Figure 1: Path Integral

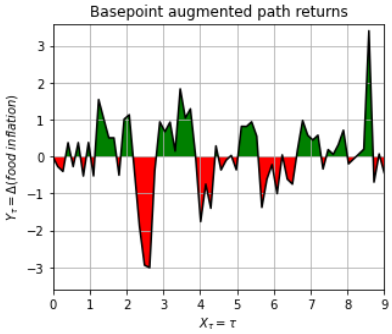


Figure 2: Basepoint

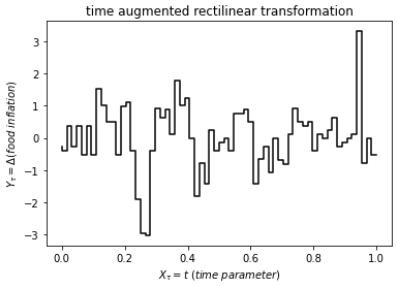


Figure 3: Time Transformation

2.1.2 Embedding, Augmentation

The first step in computing the signatures is to transform the input data from values to paths, a procedure called embedding or augmentation. There are many ways to do that, and they are elaborately discussed in [2], [3]. In this section, we cover augmentation methods relevant to our datasets. T. Lyons and A.D. Mcleod state if the returns of the time series are used (increments of the input values or levels), then the concatenation of the entries can be considered as the underlying path of the time series. Hence, to obtain a path, we scale the input data to a base of 100 and then compute the difference in the natural logarithm between two consecutive points throughout the entire sample. The embedding methods we consider fall into two categories: a) basepoint augmentation, a method with the least transformation, that keeps the number of columns unchanged and only increases the length of the sample size. b) time-stamp parametrization, rectilinear, and lead-lag augmentations are methods that increase the dimension or number of channels. A. Fermanian, 2021, describes a wide range of techniques in [2], we recite a few below, define augmentation for some $e, p \in \mathbb{N}$ as the transformation of the initial sequence $x \in S(\mathbb{R}^d)$ into one or several new sequences:

$$\phi : S(\mathbb{R}^d) \to S(\mathbb{R}^e)^p \tag{9}$$

Basepoint Adds a zero at the beginning of the time series. Also, this zero could be added at the end. This augmentation method makes the signature sensitive to translation, that is a basepoint transformation applied to the time series affects its outcome or "signature" in such a way that shifting the time series data (translation) will change the signature e.g.: adding a constant to each data point in the time series will result in a different signature. This transformation does not preserve the "shape" or pattern of the time series under translation, making it dependent on the original position or value range of the data. Kidger and Lyons, 2020, define basepoint augmentation as per [2]:

$$\phi^b : S(\mathbb{R}^d) \to S(\mathbb{R}^d) \qquad \phi^b(x) = (0, x_1, \ldots, x_n) \tag{10}$$

Time path Adds a monotone time coordinate to the path, that is time parametrization. This transformation guarantees the uniqueness of the signature Hambly and Lyons, 2010, and adds information about the parametrization of the time series. For any vector of increasing timestamps t, Levin et al., 2013, define time augmentation as per [2]:

$$\phi_t : S(\mathbb{R}^d) \to S(\mathbb{R}^{d+1}) \qquad \phi_t(x) = ((t_1, x_1), \ldots, (t_n, x_n)) \tag{11}$$

Rectilinear Maps lines to lines, preserving the structure of straight lines in geometric spaces. It applies to transformations that involve linear scaling, translation, or rotation [7]. Introduced by Chevyrev and Kormilitzin, 2016 [4], it is also known as the "axis path". The first graph to the left in Figure 4 shows the original path (red) after rectilinear transformation (black). The 2-dim path: $\{X_i^1; X_i^2\} = \{(1, 8), (2, 9), (3, 10), (4, 11), (5, 12), (6, 13), (7, 14)\}$ is rectilinearly transformed when every observation on the x-axis is extended once parallel to the y-axis, and every observation on the y-axis is extended once parallel to the x-axis except for the first and last entries, which are kept unchanged.

$$\phi^r : S(\mathbb{R}^d) \to S(\mathbb{R}^d) \tag{12}$$

$$d = 2 \qquad \phi^r(x) = \{(x_1^1, x_1^2), (x_2^1, x_1^2), \ldots, (x_n^1, x_{n-1}^2), (x_n^1, x_n^2)\}$$

$$d = 3 \qquad \begin{aligned}\phi^r(x) = \{&(x_1^1, x_1^2, x_1^3), (x_2^1, x_1^2, x_1^3), (x_2^1, x_2^2, x_1^3),\\ &\ldots, (x_n^1, x_{(n-1)}^2, x_{(n-1)}^3), (x_n^1, x_n^2, x_{(n-1)}^3), (x_n^1, x_n^2, x_n^3)\}\end{aligned}$$

Lead Lag: Creates a lagged sequence of the data. It transforms a one-dimensional time series $X : [0, T] \to \mathbb{R}^d$ into $Y : [0, T] \to \mathbb{R}^{d+\ell}$ two or higher-dimensional paths, depending on the lag parameter ℓ. A lag parameter ℓ equal to 1 maps a one-dimensional path $X_{t_i}, t = t_0, \ldots, t_T$ into a two-dimensional path $Y_{t_i} = (X_{t_i}, X_{t_{i-1}}), \forall\, i \in [0, T]$ by pairing each value (current) with its lagged (previous) value. To account for the data point lost at the start of the sample due to lagging, a zero is inserted, and the last entry is duplicated.

$$\phi^{LL} : S(\mathbb{R}^d) \to S(\mathbb{R}^{d+\ell}) \qquad (Y_{t_i}^{LL})_{i=0}^{T+1} = (X_{t_i}^{lead}, X_{t_i}^{lag})_{i=0}^{T+1}$$

$$Y_t = \begin{cases} (X_{t_i}, 0) & \text{for } t \in [i, i+1) \\ (X_{t_{i+1}}, X_{t_i}) & \text{for } t \in [i+1, T] \\ \quad\vdots & \\ (X_{t_T}, X_{t_T}) & \text{for } t \in (T, T+1] \end{cases} \tag{13}$$

A. Fermanian [2] describes and applies the time-augmented lead-lag transformation, introduced by Chevyrev and Kormilitzin (2016) [4] and Flint et al. [8]

A path generally refers to the parametrization of a multi-dimensional process. In the context of signatures application in general and our datasets in particular, a one-dimensional time series is called a path or a stream with one channel. For example:

1. One-dimensional path $\in \mathbb{R}$: A path represents a one-dimensional signal over time.
2. Multi-dimensional path $\in \mathbb{R}^d$: Multi-dimensional channels, where d represents the number of dimensions, and each d corresponds to a variable or a channel depending on the transformation from input data to path.

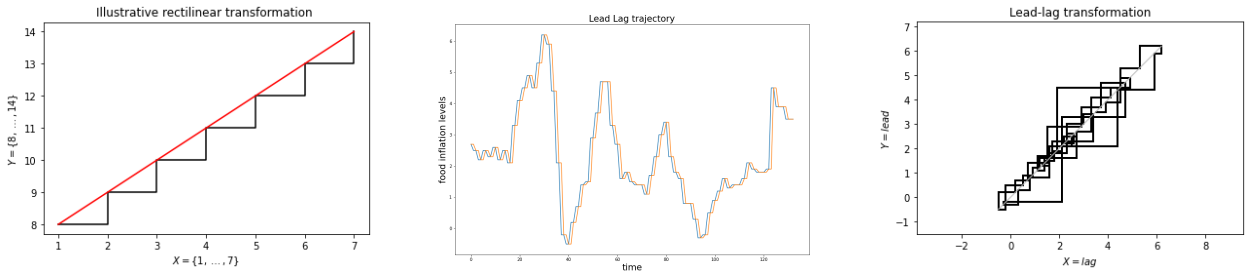


Figure 4: Rectilinear and Lead-Lag Transformation

2.2 Analytical Operators and Properties of the Signature

Before defining the signature, we discuss the analytical tools and properties of the signatures. The outer product of two vectors produces a matrix or a higher-dimensional tensor, capturing the full pairwise interaction of their components. The outer product $a \otimes b$ of two vectors $a \in \mathbb{R}^m$ and $b \in \mathbb{R}^n$ is an $m \times n$ matrix, where each entry is the product of an element from a and an element from b:

$$\mathbf{a} \otimes \mathbf{b} = \begin{pmatrix} a_1b_1 & a_1b_2 & \dots & a_1b_n \\ a_2b_1 & a_2b_2 & \dots & a_2b_n \\ \vdots & \vdots & \ddots & \vdots \\ a_mb_1 & a_mb_2 & \dots & a_mb_n \end{pmatrix}$$

Key properties of the outer product:

- ***Non-commutative***: $a \otimes b \neq b \otimes a$ (if $m \neq n$)
- ***Linearity***: In both the left and right arguments of $a \otimes b$. See 2.4
- ***Rank-1 matrix***: A, $m \times n$ matrix expressed as the outer product of two non-zero vectors $u, m \times 1$ and $v, n \times 1$. All rows (columns) are linearly dependent. A is of rank 1 where $A_{ij} = u_i v_j$ for each element $A_{ij} \in A$:

$$A = u \otimes v = uv^T$$

Definition 2.4. (Tensor product) Let V and W be two vector spaces over the same field $\mathbb{F}$. The tensor product $V \otimes W$ is a vector space over $\mathbb{F}$ with a bilinear map $\otimes : V \times W \to V \otimes W$, $\exists\, v, v' \in V, w, w' \in W$, scalars $\alpha, \beta \in \mathbb{F}$ [9]:

$$\begin{aligned} (\alpha v + \beta v') \otimes w &= \alpha(v \otimes w) + \beta(v' \otimes w) \\ v \otimes (\alpha w + \beta w') &= \alpha(v \otimes w) + \beta(v \otimes w') \end{aligned} \tag{14}$$

For any bases $\nu = \{\nu_i\}_{i=1}^m \subset V, dim(V) = m,\ \omega = \{\omega_j\}_{j=1}^n \subset W, dim(W) = n, V \otimes W$ has a basis consisting of all elements $\{\nu_i \otimes \omega_j\}$ for $i = 1, \dots, m$ and $j = 1, \dots, n$

$$\otimes(\nu \times \omega) = \{\otimes(\nu_i, \omega_j) | \nu_i \in \nu, \omega_j \in \omega\} \qquad dim(V \otimes W) = dim(V).dim(W) = m.n \tag{15}$$

$V^{\otimes n} = \overbrace{V \otimes \cdots \otimes V}^{\text{n}}$ identifies with the space of homogeneous non-commuting polynomials of degree n. Given a basis $(\nu_1, \dots, \nu_d) \subset V$, any elements of $V^{\otimes n}$ can be written as 16 which can be thought of as $\sum a^I X_{i_1}, \dots, X_{i_n}$ [2] :

$$\sum_{I=(i,\dots,i_n) \subset \{1,\dots,d\}^d} a^I \nu_{i_1} \otimes \cdots \otimes \nu_{i_1} \tag{16}$$

Definition 2.5. The extended tensor algebra $T(\mathbb{R}^d)$ [10] or the space of formal series of tensors [2] of $\mathbb{R}^d, d \in \mathbb{N}$, the truncated tensor algebra $T^N(\mathbb{R}^d)$ of order $N \in \mathbb{N}$ are given by:

$$T(\mathbb{R}^d) = \{(a_0, \dots, a_n, \dots) \,|\forall n \geq 0, a_n \in (\mathbb{R}^d)^{\otimes n}\} \tag{17}$$

$$T^N(\mathbb{R}^d) = \{(a_0, \dots, a_N) \,|\forall n \in \{0, \dots, N\}, a_n \in (\mathbb{R}^d)^{\otimes n}\} \tag{18}$$

For $n \in \mathbb{N}_0$ the nth tensor product of the vector space $\mathbb{R}^d$ is given 19. For any $a \in T(\mathbb{R}^d), a = \sum_{n \geq 0} a_n$. For any $a, b \in T(\mathbb{R}^d), \lambda \in \mathbb{R}, T(\mathbb{R}^d)$ is endowed by $(+, \cdot, \otimes)$ operations, and a neutral element $\mathbf{1} := (1, 0, \dots, 0, \dots)$ [2]:

$$(\mathbb{R}^d)^{\otimes 0} = \mathbb{R} \qquad (\mathbb{R}^d)^{\otimes n} = \underbrace{\mathbb{R}^d \otimes \cdots \otimes \mathbb{R}^d}_{\text{n}} \tag{19}$$

$$a + b = (a_0 + b_0, \dots, a_n + b_n, \dots) \qquad \lambda \cdot a = (\lambda \cdot a_0, \dots, \lambda \cdot a_n, \dots) \tag{20}$$

$$a \otimes b = (c_0, \dots, c_n, \dots), \qquad \text{where } c_n = \sum_{k=0}^{n} = a_k \otimes b_{n-k} \tag{21}$$

The signature of a path captures its behavior through a sequence of iterated integrals. The shuffle product is an algebraic structure associated with these iterated integrals, making the signature computation more efficient and allowing deeper insights into the path's structure. It is a specific type of product between two words or sequences and is commutative and associative. Given two "words" or sequences of indices a and b, their shuffle product, denoted $a \sqcup\!\sqcup b$, is the sum of all possible ways of interleaving a and b while preserving the order of their individual elements. For example, $a = (x_1, x_2)$ and $b = (x_3)$, their shuffle product contains $(m+n)!/m!n!$ elements, and:

$$a \sqcup\!\sqcup b = (x_1, x_2, x_3) + (x_1, x_3, x_2) + (x_3, x_1, x_2)$$

Definition 2.6. The $\sigma \in \text{Shuffle}(m, n)$ is a permutation of $(m+n)$ elements that combines two sequences $a = (a_1, \ldots, a_m)$ and $b = (a_1, \ldots, b_n)$ by summing over all permutations that preserve the relative order of each sequence for a finite set of multi-indices $i_1, \ldots, i_m, j_1, \ldots, j_n \in \{1, \ldots, d\}$ [2], where each $c_{\sigma(k)}$ is an element from either a or b and follows the permutation order dictated by the shuffle set $\sigma(i) = j \in \text{Sh}(m, n)$ [3]:

$$(a_1, \ldots, a_m) \sqcup\!\sqcup (b_1, \ldots, b_n) = \sum_{\sigma \in Sh(m,n)} (c_{\sigma(1)}, \ldots, c_{\sigma(m+n)}) \tag{22}$$

where $(c_1, \ldots, c_m, c_{m+1}, \ldots, c_{m+n}) = (i_1, \ldots, i_m, j_1, \ldots, j_n)$

The shuffle product preserves the relative order of the first m elements and the last n elements, That is, if a and b are two blocks of elements consisting of m and n elements respectively, a shuffle σ rearranges these two blocks without mixing the order within each block. Hence, the shuffle product of two sequences (or words) a and b is the sum of all sequences formed by "shuffling" a and b in all possible ways, while preserving the order within each original sequence. The shuffle product plays a crucial role in computing and understanding the signature because of its relation to iterated integrals. The relationship between the shuffle product and Chen's identity is essential for understanding how paths are combined and interact through their signatures. The Chen's identity is fundamental in the study of paths and their iterated integrals. In rough path theory, concatenating two paths refers to joining them end-to-end. The Chen's identity is used for such concatenation:

Definition 2.7. Given two paths X and Y defined on intervals $[0, T_1]$ and $[0, T_2]$ respectively. We can construct a concatenated path $X * Y$ on $[0, T_1 + T_2]$ as follows:

$$(X * Y)(t) = \begin{cases} X(t) & \text{for } t \in [0, T_1] \\ X(T_1) + (Y(t - T_1) - Y(0)) & \text{for } t \in [T_1, T_1 + T_2] \end{cases} \tag{23}$$

Chen's identity is crucial in describing how the signature of a concatenated path $X * Y$ relates to the signatures of X and Y individually. It allows us to compute the signature of a long path by breaking it into segments, calculating the signatures of each segment, and then combining them through tensor products. This recursive structure is central to the study of rough paths, as it enables the development of powerful tools for analyzing complex paths by understanding simpler components. Therefore, the signature $S(X)$ of a path X is an element in the tensor algebra over the path's values, generally denoted as $T((\mathbb{R}^d))$ when X takes values in $\mathbb{R}^d$. Let $S(X) = \sum_{k=0}^{\infty} S(x)_k$ where $S(X)_k$ is the k-th level tensor, then Chen's identity states that the signature of the concatenated path $S(X * Y)$ can be expressed as the tensor product of the signatures of X and Y:

$$S(X * Y) = S(X) \otimes S(Y)$$

In expanded form, this implies that each term in the signature of the concatenated path $S(X * Y)$ at level k can be expressed by 24, which shows that the k-th level of the signature of the concatenated path $X * Y$ is obtained by taking the tensor products of all pairs of signature levels from X and Y that add up to k, and then summing these tensor products [11]:

$$S(X * Y)_k = \sum_{i=0}^{k} S(X)_i \otimes S(Y)_{k-i} \tag{24}$$

Definition 2.8. A continuous map $X = (1, X^1, X^2)$ from the simplex $\Delta = \{(s, t) : 0 \leq s \leq u \leq t \leq 1\}$ into the truncated tensor algebra, $X : \Delta \to T^{(2)}(\mathbb{R}^d)$ is said to be a rough path if the following two conditions are satisfied [12]:

(i) (Chen's identity) For any $0 \leq s \leq u \leq t \leq 1$

$$X^{(1)}_{s,t} = X^{(1)}_{s,u} + X^{(1)}_{u,t} \qquad\qquad X^{(2)}_{s,t} = X^{(2)}_{s,u} + X^{(2)}_{u,t} + X^{(1)}_{s,u} \otimes X^{(1)}_{u,t}$$

(ii) (finite p-variation) $\left\|X^1\right\|_p < \infty, \quad \left\|X^2\right\|_{p/2} < \infty$

This means that the first-level increment $X^{(1)}_{s,t}$ (the path itself) is additive over segments, that is the total variation over $[s,t]$ is the sum of increments over $[s,u]$ and $[u,t]$. The second-level increment encapsulates the nontrivial structure in rough path theory: not only do we add the second-level increments of the individual segments, but we also include an additional term given by the tensor product of the first-level increments [11].

Example 2. Y. Inahama [12], For a continuous path $x : [0,1] \to \mathbb{R}^d$ of 1-variation that starts from 0 and $(s,t) \in \Delta$:

$$X^1_{s,t} = \int_s^t dx_{t_1} = x_t - x_s \qquad\qquad X^2_{s,t} = \int_{s \le t_1 \le t_2 \le t} dx_{t_1} \otimes dx_{t_2} = \int_s^t (x_u - x_s) \otimes dx_u$$

2.3 Signature of a Path

The signature of a path is intrinsically multidimensional [3]. Its computational architecture is highly flexible and dependable on the underlying dataset. There are at least four steps to consider before calculating the signatures:

(i) *Path dimension* d, number of columns or channels, which can vary depending on the number of variables and transformation methods used.

(ii) *Signature levels* as each level corresponds to the binomial expansion of powers, making it more computationally expansive. In this paper, we are interested in truncated signature levels of up to 5.

(iii) *Augmentation method* or transformation of input data to paths. For example. for a lag parameter of 1, the lead-lag method doubles the column dimension, making it more computationally expansive whereas, the basepoint transformation keeps the column dimension unchanged. For us, both basepoint and lead-lags augmentations are relevant.

(iv) The window method helps capture locality. We consider three methods: a) sliding window, b) expanding window, and c) Dyadic or hierarchical.

Given a path $X : [0,T] \to \mathbb{R}^d$, the signature of X over the interval $[0,T]$ is the sequence of all iterated integrals of X up to a certain order. These iterated integrals act as descriptors of the path and capture all non-linear dependencies within the path's structure. [13]

Definition 2.9. For a path $X : [0,T] \to \mathbb{R}^d$, its signature $S(X)$ is given by the collection of iterated integrals 25, where each iterated integral in the sequence represents a term of the signature:

$$S(X) = \left(1, \int_0^T dX_t, \int_0^T \int_0^{t_1} dX_{t_2} \otimes dX_{t_1}, \int_0^T \int_0^{t_1} \int_0^{t_2} dX_{t_3} \otimes dX_{t_2} \otimes dX_{t_1}, \dots\right) \tag{25}$$

The n-th order term in the signature sequence is given by the tensor product of each integral of the increments dX_t over the simplex $0 \le t_n \le t_{n-1} \le \cdots \le t_1 \le T \in \mathbb{R}^d$:

$$S(X) = \int_0^T \int_0^{t_1} \cdots \int_0^{t_{n-1}} dX_{t_n} \otimes dX_{t_{n-1}} \otimes \cdots \otimes dX_{t_1} \tag{26}$$

The factors we use come from three different datasets:

1. Unique to every stock, fundamental data aka dataset 1 e.g.: price-to-book ratio, pe, etc.
2. Common to all stocks, macro data aka dataset 2 e.g.: Inflation, GDP, etc.
3. Common to all stocks, market data aka dataset 3 e.g.: ETFs, msci acwi, etc.

Given this information, it is advisable to choose the augmentation and window methods based on the dataset. Nonetheless, for every stock, we use the signature levels with the best adjusted R^2 to determine the exponent level of the polynomial equation. One major contribution of this PhD thesis is to determine the exponent level of the polynomial or polymodel equation from the signature level. Following our last conversation, I integrated the signature genetic algorithm code with the polymodels code, whereby, previously all stocks had a fixed polynomial exponent level of 4. Now, each stock has an exponent level of 1, 2, 3, or 4 determined by the adjusted R^2 of the signature level. This section will include math definitions of path/path integral, tensor, and signatures, along with the results done on three datasets independently, and all datasets taken together to validate the approach.

3 Genetic Algorithm

We implement the integer-based genetic algorithm process because of the nature of our optimization or selection process, which is to select factors or columns each indexed to an integer-based number. We use the GA Python library PyGAD. In addition, to validate PyGAD results, we created an entirely customized GA process in Python with standalone functions and functionality similar to PyGAD but completely transparent with no hidden layers for deeper understanding and clarity, especially while fine-tuning the parameters of the steps below. The GA consists of the following steps:

1. Initial population: randomly construct 100 sets of a 5-genome chromosome or 100 sets of a 5-col vector of unique integers between 1-200, e.g.: $[1, 33, 156, 199, 7], [22, 39, 116, 99, 19]$
2. Fitness, suitable evaluation, or objective function: in our case, the fitness function takes the input data, converts it into paths, partitions the paths into windows, computes the signatures, and then estimates the fitted values using the partial least square or random forest method depending on the underlying dataset; for instance, for the serially correlated dataset such as dataset 1, it is not wise to use random forest, whereas for a highly random or a white noise dataset such as dataset 3, it could be more suitable to use random forest versus PLS. Evaluate the fitness function of every member of the population, then rank based on the selection function.
3. Selection function: It is responsible for selecting individuals or solutions from the population to be parents for the next generation. We consider three selection methods: a) Roulette wheel, where individuals are selected with a probability proportional to their fitness. Better individuals have a higher chance of being selected. b) Stochastic universal sampling is a variation of the roulette wheel with a more uniform selection of individuals. c) Tournament selection involves selecting a small group (tournament) at random and then the best individual from this group is chosen as a parent. For our application, we find the tournament selection method most suitable.
4. Crossover function: we use a two-point crossover, which involves combining/swapping genes or integer-based numbers from two parents to generate new offspring or a new set of integers e.g: $[1, 33, 116, 99, 7]$.
5. Mutation function: we use the uniform mutation function, where a gene is randomly replaced with a value from the possible range of values. Similar to crossover, it introduces random variations in the genetic material of individuals (solutions) in a population. It helps maintain diversity in the population and avoid stagnation around local optima. e.g.: $[1, 33, 158, 199, 7]$
6. Replacement: determines which individuals move to the next generation, by replacing some of the old population members with the new offspring.
7. Repeat: set a large enough number of generations and for each generation, loop through steps 2-6 until a termination condition is met while allowing the population to change for each generation.

The ***initial population*** function generates a list (population) of lists (individuals), where each individual has a specified number of genes, each randomly chosen from a uniform integer distribution between the given lower and upper bounds. Take population P as the set of N individuals, where each individual is a vector of G genes:

$$\mathcal{P} = \{\mathbf{x}_i \mid \mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iG}),\ x_{ij} \sim \text{UniformDiscrete}(L, U) \text{ for all } j\} \tag{27}$$

where $i = (1, \ldots, N)$ and $j = (1, \ldots, G)$

The ***fitness function*** evaluates individual members of the population based on a cost or performance metric. This is where the signatures (coefficients) are computed, and then their relationships are estimated through an appropriate algorithm, i.e.: partial least squares, random forest, etc, depending on the dataset. See section 4 for more details.

The ***selection function*** determines how individuals (solutions) are chosen to participate in creating the next generation. The selected individuals (parents) then mate and recombine to produce offspring, forming the next generation (population). Python's genetic algorithm library PyGAD offers several selection methods such as tournament selection, stochastic universal sampling, and roulette wheel selection, each with distinct characteristics. We use the tournament selection method: each individual $\mathbf{x}_i \in \mathcal{P} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ has an associated fitness $f(\mathbf{x}_i)$, where $f : \mathcal{P} \to \mathbb{R}$. Let T be the tournament size, which determines the number of individuals randomly selected from the population in each tournament.

- ***Tournament Sampling***: For each selection, draw a subset of T individuals uniformly at random from the population. Let $\mathcal{T}_k$ be the subset for the k-th tournament, where each $\mathbf{x}_{i_j}$ is randomly chosen from $\mathcal{P}$ without replacement:

$$\mathcal{T}_k = \{\mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \ldots, \mathbf{x}_{i_T}\}$$

- ***Tournament Winner***: For each tournament subset $\mathcal{T}_k$, select the individual $\mathbf{x}_{i^*} \in \mathcal{T}_k$ with the best fitness. This individual is the "winner" of the tournament and is added to the selected population:

$$\mathbf{x}_{i^*} = \arg\max_{\mathbf{x} \in \mathcal{T}_k} f(\mathbf{x})$$

- ***Repeat and Build Selected Population***: Repeat the tournament process until a new population of N individuals is created. Denote $\mathcal{P}_{selected}$ the selected population, where each $\mathbf{x}_{i_k^*}$ is the winner of the k-th tournament:

$$\mathcal{P}_{selected} = \{\mathbf{x}_{i_1^*}, \mathbf{x}_{i_2^*}, \ldots, \mathbf{x}_{i_N^*}\} \tag{28}$$

The ***crossover function*** generates offspring by pairing individuals from a set {mom, dad}, where each pair represents two consecutive individuals in $\mathcal{P}_{selected}$. This process is repeated until either all pairs in $\mathcal{P}_{selected}$ are used or a specified number of offspring, such as 80%, have been produced. We present a custom crossover function that combines common and unique genes from two parents to produce offspring. The formulation ensures that the offspring inherit a balanced mix of genetic material from both parents while maintaining genetic diversity within the population, which is crucial for the effectiveness of genetic algorithms in exploring the solution space.

- ***Random Role Assignment***: Introduces variability in parent contributions.
- ***Chromosome Segmentation***: Identifies shared and distinct genetic material.
- ***Random Shuffling and Selection***: Ensures diversity and unbiased gene selection.
- ***Chromosome Assembly and Sorting***: Creates a coherent and consistent offspring genome.

Denote parents by $\mathcal{P}_1 = (p_1^{(1)}, p_2^{(1)}, \ldots, p_G^{(1)})$, $\mathcal{P}_2 = (p_1^{(2)}, p_2^{(2)}, \ldots, p_G^{(2)})$ (mom and dad respectively), offspring $\mathcal{C}_i$ (crossover child), $G = 5$ number of genes per parent, C common chromosomes, U unique chromosomes, and H half-length for selection. The random role assignment introduces variability by randomly deciding with equal probability which parent contributes as "mom" and "dad" in the crossover process:

$$Pr(\mathcal{P}_{mom} = \mathcal{P}_1) = Pr(\mathcal{P}_{dad} = \mathcal{P}_2) = \frac{1}{2} \tag{29}$$

Chromosome segmentation identifies common and unique chromosomes, genes present in both parents, those unique to mom, and those unique to dad:

$$C = \mathcal{P}_{mom} \cap \mathcal{P}_{dad} = \{x \mid x \in \mathcal{P}_{mom} \text{ and } x \in \mathcal{P}_{dad}\} \tag{30}$$
$$U_{mom} = \mathcal{P}_{mom} \setminus \mathcal{P}_{dad} = \{x \mid x \in \mathcal{P}_{mom} \text{ and } x \notin \mathcal{P}_{dad}\} \tag{31}$$
$$U_{dad} = \mathcal{P}_{dad} \setminus \mathcal{P}_{mom} = \{x \mid x \in \mathcal{P}_{dad} \text{ and } x \notin \mathcal{P}_{mom}\} \tag{32}$$

The length H helps determine how many unique genes to select from each parent to contribute to the child. Variable $|U_{mom}|$ is the cardinality (number of gene elements) in the U_{mom} set :

$$H = \left\lceil \frac{|U_{mom}|}{2} \right\rceil \tag{33}$$

Shuffle unique chromosomes and select subsets of permuted elements without replacement. Randomize the order of unique genes to ensure that the selection is unbiased, then select the first H genes from each shuffled set to contribute to the child:

$$U'_{mom} = \text{Shuffle}(U_{mom}) \qquad U'_{dad} = \text{Shuffle}(U_{dad}) \tag{34}$$

The selected subsets are:

$$\begin{aligned} S_{mom} &= \{U'_{mom}(1), U'_{mom}(2), \ldots, U'_{mom}(H)\} \\ S_{dad} &= \{U'_{dad}(1), U'_{dad}(2), \ldots, U'_{dad}(H)\} \end{aligned} \tag{35}$$

Assembly child chromosome $\mathcal{C}_i$ combines the common and selected unique genes to form the crossover offspring. Then, to mix the genes thoroughly, randomly shuffle the assembled chromosomes and sort to maintain consistent gene order:

$$\begin{aligned} \mathcal{C}_i &= C \cup S_{mom} \cup S_{dad} \\ \mathcal{C}_i &= \text{Shuffle}(\mathcal{C}_i, G) \\ \mathcal{C}_i &= \text{Sort}(\mathcal{C}_i) \end{aligned} \tag{36}$$

$$\mathcal{X}[i,:] = \mathcal{C}_i$$

The ***mutation function*** generates offspring where elements (genes) of an individual (solution) are altered with a certain probability while the values remain within specified bounds. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be the input individual with n genes, $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$, p_m the mutation rate, L and U lower and upper bounds for the genes, $R(i) \sim \text{Uniform}(L, U)$ be a random integer generated for mutation. Let $r_i \sim \text{Uniform}(0, 1)$ for $i = 1, 2, \ldots, n$ be a random value determining whether mutation occurs for the i-th gene, then the mutation process is defined as:

$$x'_i = \begin{cases} R(i) & \text{if } r_i < p_m \\ x_i & \text{otherwise,} \end{cases} \tag{37}$$

$$\mathcal{M}[j,:] = \mathbf{x}'_j$$

The mutation value x'_i is clipped to the bounds $[L, U]$, ensuring: $x'_i = \max(\min(x'_i, U), L)$. The mutated individual is $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_n)$ where each gene has been potentially replaced by a new random value within the bounds based on the mutation probability p_m.

4 Signature Genetic Algorithm

Implementing the signature over the genetic algorithm optimizer occurs within the fitness function, which consists of five main steps:

1. The augmentation approach creates a smooth continuous path(s) and removes the signature invariance to translation and/or reparametrization. At a minimum, we need two consecutive data points to calculate the signatures. Indeed, we can calculate the signatures on the entire length of the path(s) also known as the global window, but this will then project the first output observation onto the whole domain of the paths, which can be reasonable when each output observation represents an image and in turn each image is represented by a vector or an array of coordinates, like an airplane or a flower. When dealing with paths derived from time series, it is more suitable to compute the signatures over sub-windows, such as sliding, expanding, or hierarchical windows.
2. Window methods [2]: a) The sliding window method uses the same scale to compute all signatures. The sliced paths have fixed lengths but shifted in time. b) Expanding windows produces paths of increasing lengths, similar to the history processes of stochastic analysis. c) Dyadic or hierarchical captures information at different scales.
3. Signature at truncated levels: The signatures can be thought of as the polynomial expansion of paths. The signature levels help capture stylized facts of financial time series without imposing an explicit probability distribution on the future returns [14]. The signature length or the number of coefficients obtained from computing the signatures is exponential in the truncation level. For example, the level-4 signatures produced by a path of two dimensions (2-column array) lead to 30 coefficients, and 780 for a path of dimension equal to 5 (5-column array). For a 5-gene Chromosome, we compute the signatures at levels 1, 2, 3, and 4. Then using the adjusted-R^2 performance measure, we select the polynomial exponent for the polymodels based on the signature level with the highest adjusted-R^2. This means, that every stock will have a unique polynomial expansion and a lag parameter depending on the signature results.
4. Estimation algorithm, or objective function: Consistent with our datasets and the ultimate goal of constructing a 5-factor model for each stock with unique polynomial representation, we alternate between partial least square (PLS) and random forest (RF) models with the number of estimators no larger than the length of the path(s). For serially correlated data, such as dataset 1, we do time-parametrized, lead-lag path(s) augmentation, under PLS with 10 estimators. For heteroskedastic data such as dataset 3, where the variance varies over time, we use time-parametrized, basepoint augmentation (least transformation) under RM with 10 decision trees.
5. Performance - statistical, measure, or cost function: We use the adjusted-R^2 to measure the improvements of the fitted functions. The number of generations is fixed based on the stopping condition of at least 10 generations of no improvement in the out-of-sample adjusted-R^2.

The signature lengths i.e.: the number of coefficients, that is the signature dimensions at every level is given by:

$$\sum_{k=0}^{K} d^k = \frac{d^{K+1} - 1}{d - 1} - 1 \quad d \neq 1 \tag{38}$$

4.1 The Fitness Function

What is the optimal fixed-size combination and formulaic representation of a subset of approximately 200-250 explanatory variables to model a stock's path? Furthermore, under what conditions can we describe the stock's stream of increments (returns) as quadratic or of order 3, 4, and so on? The signature captures information about the path at different levels of complexity. Lower-order terms (e.g., levels 1 and 2 in the signature) correspond to linear and quadratic relationships, while higher-order terms capture increasingly complex interactions. The signature expansion can be truncated at a certain level (or depth), which might implicitly reflect the level of polynomial interactions needed to approximate the stock return's behavior. As such, deploy the signature as a feature extraction method in a regression problem by examining which levels (or orders) of the signature contribute most significantly to explaining stock returns. From this analysis, infer the polynomial order that best fits the data (stock returns) using the original factors data or returns before augmentation.

Define the mapping $\zeta : F \to \mathcal{G} = \zeta(F)$, where $F \in \mathbb{R}^{T\times M}$ represents the standardized factor returns with T time steps and M explanatory variables. Let $\mathcal{G} \in \mathbb{R}^{T\times G} \subseteq \{1, 2, \ldots, M\}$ denote the proposed genetic algorithm solution, which is a subset of indices $\in \mathbb{N}$ corresponding to the selected factors. An initial population of 100 individuals is randomly chosen, with G representing the number of genes. Denote $r \in \mathbb{R}^{T'\times C}$ as the time-parameterized array of standardized factor returns transformed under one of the augmentation methods discussed above 2.1.2, C corresponds to the number of channels or the dimension of r. Let x be a multivariate matrix of factor returns, where x_i represents the time series or stream of data as a list of tuples comprising time and values. Denote y_i the output corresponding to the input stream x_i. We can express:

$$\begin{gathered} x = \{x_1, x_2, \ldots, x_T\}, \\ x_i = \{(t_{i1}, v_{i1}), (t_{i2}, v_{i2}), \ldots, (t_{im}, v_{im})\}, \\ y = \{y_1, y_2, \ldots, y_T\}. \end{gathered}$$

Finally, the mapping can be written as:

$$\mathcal{G} = \zeta(x_1, x_2, \ldots, x_T) \tag{39}$$

The function ζ encapsulates the fitness function, which encompasses embedding r, partition into windows w, signature Sig, regression (β, adj-R^2), and genetic algorithm optimizer Ga. Let $\Xi : \mathrm{G} \to \mathrm{E}$ be the mapping that transforms the solution $\mathcal{G}$ into an augmented representation $r := C_{emb} \in \mathrm{R}^{T'\times C}$. Denote $\varpi : \mathrm{E} \to \mathrm{Z}$ the mapping that partitions the augmented solution r into overlapping sliding windows of size w, where $z = \{z_1, z_2, \ldots, z_{n_w}\},\ z_i \in \mathbb{R}^{T'_w\times C}, n_w = T' - w + 1$ is the number of windows, and c_{ig} is the channel associated with genetic algorithm solution. If the lead-lag parameter is 1 for a 5-variable model then the number of channels is 11 including the time-parametrization channel. We can write:

$$r = \Xi\big(\mathcal{G}(x_i)\big) \tag{40}$$

$$z = \varpi(r) = \varpi\big((t'_{i1}, c_{i1}), (t'_{i2}, c_{i2}), \ldots, (t'_{ig}, c_{ig})\big) \tag{41}$$

Now, split the data into train and test samples: $\mathrm{Z}_{train}, \mathrm{Z}_{test}, \mathrm{y}_{train}, \mathrm{y}_{test}$, where Z_{train} contains 70% of the windows, and Z_{test} contains the remaining 30%. For each input z_i, apply a signature transformation $\mathrm{Sig}(z_i, k)$ to extract features up to a specified order k. The signature can be represented in 42, where $\mathrm{Sig}_j(z_i)$ represents the j-th level of the signature. The signatures of up to level 3 of an array of 11 channels yield 1463 features for each z_i, i.e.: $\mathrm{Sig}((w \times 11), 3) = 1463$ coefficients, for a sample size of 66. the training set of z_i has 43 rows $0.7 \times (66 - 5 + 1)$. Subsequently, construct the feature matrix $\mathbf{X}$ by applying the signature transformation to all input streams, where each row $\mathrm{Sig}(z_i, k)$ is the feature vector for the corresponding z_i

$$\mathrm{Sig}(z_i, k) = \{\mathrm{Sig}_1(z_i), \mathrm{Sig}_2(z_i), \ldots, \mathrm{Sig}_k(z_i)\} \tag{42}$$

$$\mathbf{X} = \begin{bmatrix} Sig(z_1, k) \\ Sig(z_2, k) \\ \vdots \\ Sig(z_{n_w}, k) \end{bmatrix}$$

At this stage, we are ready to train the model, evaluate its performance, and iterate through each individual (a 5-gene element) in the population and subsequent generations of the genetic algorithm (GA) until specific termination criteria are met. These criteria include reaching a maximum of 100 generations or observing no improvement in fitness values for 20 consecutive generations. A regression model is trained on the dataset $(\mathbf{X}, \mathbf{y})$, where $\mathbf{y} = [\mathrm{y}_1, \mathrm{y}_2, \ldots, \mathrm{y}_{n_w}]$ is the

output vector. The choice of the regression model depends on the dataset and is primarily the modeler's discretion; in this case, we use the partial least squares (PLS) with d components. The PLS regression [15] $\mathrm{X} = \mathrm{TP}^\top + \mathrm{E}, \mathrm{y} = \mathrm{T}q + e$ decomposes X into latent scores $\mathrm{T} \in \mathbb{R}^{n_w \times d'}$ and loading $\mathrm{P} \in \mathbb{R}^{p \times d'}$, and loading $\mathrm{Q} \in \mathbb{R}^{1 \times d'}$ for y. It aims to maximize the covariance between T and y while reducing X and y to latent components. The model is trained by solving 43. Let $\mathrm{X} \in \mathbb{R}^{n_w \times p}$ be the input matrix of n_w samples and p features, $\mathrm{y} \in \mathbb{R}^{n_w \times 1}$ the output vector. The first step is to compute the latent scores T for X. This is done by finding a weight vector w such that the projection $\mathrm{t} = \mathrm{Xw}$ maximizes the covariance with y:

$$\mathrm{w} = \arg\max_{\mathrm{w}} \mathrm{cov}(\mathrm{Xw}, \mathrm{y})$$

subject to $\|\mathrm{w}\| = 1$. Once the w is found, the latent scores are computed $\mathrm{t} = \mathrm{Xw}$. The second step finds the regression coefficients between t and y. The loading for y denoted q is the regression coefficient of y onto the latent score t. The loading p for X are given:

$$\mathrm{p} = \frac{\mathrm{X}^\top \mathrm{t}}{\mathrm{t}^\top \mathrm{t}}, \qquad\qquad q = \frac{\mathrm{y}^\top \mathrm{t}}{\mathrm{t}^\top \mathrm{t}}$$

Thirdly, deflate both X and y to remove the information explained by the current latent component. The process iterates over d components, progressively deflating both X and y. Finally construct the PLS regression model, where w_i and q_i are the weights and loadings corresponding to the i-th latent component [16]:

$$\mathbf{X} \longleftarrow \mathrm{X} - \mathrm{tp}^\top, \qquad\qquad \mathbf{y} \longleftarrow \mathrm{y} - \mathrm{t}q^\top,$$

$$\hat{\mathrm{y}} = \mathrm{Xw}_1 q_1 + \mathrm{Xw}_2 q_2 + \cdots + \mathrm{Xw}_\mathrm{d} q_d$$

Compactly, given the outputs vector $\mathbf{y} = [y_1, y_2, \ldots, y_{n_w}]^\top$, the partial least squares regression with d components, $\beta \in \mathbb{R}^d$ coefficients, ϵ residual error can be expressed:

$$\mathrm{y} = \mathbf{X}\beta + \epsilon \qquad\qquad \hat{\beta} = \min_{\beta} \|\mathrm{y} - \mathrm{X}\beta\|^2 \tag{43}$$

Consider the signature-based genetic algorithm learning model $\mathcal{M}$ (SigGA) i.e.: the function ζ discussed above. Train on X_{train} and y_{train}. Then, make predictions and evaluation (adj-R^2) using X_{test} and y_{test}, n_{test} the length of the test set, μ the average of y_{test}, and d' the number of estimators or effective parameters such as $d' \geq G + 1$:

$$\mathcal{M} : \mathrm{X}_{train} \to \mathrm{y}_{train} \qquad\qquad \hat{\mathrm{y}} = \mathcal{M}(\mathrm{X}_{test}) \tag{44}$$

$$\bar{R}^2 = 1 - (1 - R^2) \times \frac{(n-1)}{(n - d' - 1)} \qquad\qquad R^2 = 1 - \frac{(\mathrm{y}_{test} - \hat{\mathrm{y}}_{test})^\top \cdot (\mathrm{y}_{test} - \hat{\mathrm{y}}_{test})}{(\mathrm{y}_{test} - \mu)^\top \cdot (\mathrm{y}_{test} - \mu)}$$

The effective number of parameters d' is at least as large as the number of genes plus the time augmentation is different from the total number of channels $d = G(LL + 1) + 1 = 5(2) + 1 = 11$ when the lead-lag parameter $LL = 1$ and the total number of coefficients or features $p = (d^{k+1} - 1)/(d - 1) - 1 = 1463$ for $k = 3$ signature level. Because of limitations due to sample length and polynomial regression, we limit d' to the values $d' = [6, 10, 15, 20, 25]$. The flexibility of the SigGA "supervised learning" fitness function comes with a cost related to the various, pre-processing (embedding) and post-processing (windows) transformations, signatures computed at truncated levels (the number of features increases exponentially with the levels of signatures), the effective number of estimators for the evaluation function (PLS, random forest, etc.), in addition to the original number of genes or variables desired in the study of stock's path. Furthermore, the genetic algorithm has notable parameters that need to be configured. In this analysis, we set the same GA configuration for all signature computations. The default GA setting consists of a population of size 100 with 100 generations, one mutated gene per offspring, the tournament (parent) selection function, and 60 parent mating. Hence, given a GA configuration, we want to use the SigGA process to make statements about the stocks' paths nonlinear forms, capturing and measuring the interaction effects, the lagged relationship of stocks with the relevant factors, and whether a particular dataset adds value or signal to the process.

- ***Nonlinear Form***: Which polynomial order (2, 3, or 4) best fits each stock in our universe?
- ***Lead-Lag***: Which lagged level $(1, \text{or } 2)$ best relates each stock with its selected factors?

- ***Feature Engineering***: What are the benefits of combining three datasets (fundamentals, economic/financial, and market data i.e. ETFs) compared to analyzing each dataset separately?

A. Fermanian in [2] chapter 3 p. $62, 63$ finds embedding methods time + basepoint and lead-lag to yield better results than the other methods. Similarly, the author finds dyadic and expanding windows methods to be significantly better than sliding windows. In this approach, two consecutive windows overlap entirely except for the first row of the first window and the last row of the succeeding window. The expanding window, consistent with the historical look-back method of stochastic analysis, produces arrays of increasing lengths. The dyadic window method captures information at different scales by breaking sequential data into overlapping (fixed windows) and non-overlapping segments that align with a hierarchical multi-scale partition of the interval of interest. A dyadic partition divides an interval $[0, T]$ into sub-intervals of equal length, where the size of each sub-interval is $T/2^q$ for some q. For instance, at level $q = 2$, the interval $[0, 1]$ would be partitioned into $[0, 0.25], [0.25, 0.5], [0.5, 0.75], [0.75, 1]$. For each segment (window), extract the portion of the path corresponding to the partition i.e. $[0, 0.25]$, compute the signature for the extracted partition, and then combine the computed signatures to represent the path. The method allows for local analysis of paths making it computationally feasible to handle long or complex time-series data. We begin with the combined dataset, which includes the entire universe of explanatory variables (approximately 200 per stock), and apply expanding windows as the default method, a commonly used in financial time series analysis. Repeat the computation under sliding and dyadic windows. We proceed as follows, given the concatenated dataset and a window method, we run the SigGA process nine times as table 1 illustrates the number of coefficients estimated for expanding and sliding windows. These numbers are doubled under dyadic window when the subinterval division is set to $n = 2$. Then, do the same for each dataset separately i.e.: Fundamental—balance sheet data only, economic/financial data only, and market data, ETFs only.

Sig Level \ Augmentation	Time + BP	LL1 + Time	LL2 + Time
Order 1	6	11	16
Order 2	42	132	272
Order 3	258	1463	4368

Table 1: Number of coefficients of a 5-gene chromosome

4.2 Deciding on the Polynomial Order

Use the signature genetic algorithm results to find an optimal multivariate polynomial representation for each stock. Multivariate polynomial equations provide powerful tools to derive risk measures for stock returns. Polynomials of orders 3 and 4 go beyond the mean and the variance up to the skewness and kurtosis measures. Together, the four moments provide a comprehensive description of the statistical properties of asset returns. Including skewness and kurtosis in portfolio analysis can offer deeper insights into the behavior of asset returns, particularly in capturing the impact of asymmetric risks and extreme events. The path-dependence nature of the signatures and the higher levels incorporate many stylized facts about asset prices such as fat tails, skewness, and volatility clustering [14]. In this section, we bridge from the SigGA to multivariate polynomials of five variables of orders $2, 3$ and 4, first to validate the SigGA results, and second to select the best polynomial form for each stock.

For a polynomial with n variables and degree d, the number of terms i.e.: the number of combinations, is given by the binomial coefficient formula:

$$\text{Number of terms} = \binom{n+d}{d} = \frac{(n+d)!}{n!\,d!}$$

Example 1. The full expansion of a multivariate polynomial of order 3 in 3 variables (x, y, z) includes all terms where the sum of the exponents of x, y and z in each term is less than equal to 3. It looks like this:

$$\begin{aligned} P(x, y, x) = {} & c_0 + c_1x + c_2y + c_3z \\ & + c_4x^2 + c_5y^2 + c_6z^2 + c_7xy + c_8xz + c_9yz \\ & + c_{10}x^3 + c_{11}y^3 + c_{12}z^3 + c_{13}x^2y + c_{14}x^2z + c_{15}y^2x + c_{16}y^2z + c_{17}z^2x + c_{18}z^2y + c_{19}xyz \end{aligned}$$

Where:

1. Constant term (c_0): a standalone constant.

2. Linear terms $(c_4x^2, c_5y^2, c_6z^2, c_7xy, c_8xz, c_9yz)$: degree of 1 terms.
3. Linear terms $(c_{10}x^3, c_{11}y^3, c_{12}z^3, c_{13}x^2y, c_{14}x^2z, c_{15}y^2x, c_{16}y^2z, c_{17}z^2x, c_{18}z^2y, c_{19}xyz)$: degree of 3 terms.

4.3 Data Definitions

We analyze stocks against a comprehensive global dataset, which is a concatenation of three distinct datasets: fundamental dataset, economic and financial data, and market data i.e. exchange-traded funds, all expressed quarterly. The fundamental data capture companies' balance sheet information, cash flow statements, income statements, and other metrics as defined in appendix E. The economic and financial dataset consists of macroeconomic indicators such as the balance of trade, building permits, business confidence, capital flows, etc as defined in appendix F. The exchange-traded funds (ETF) dataset defined in appendix G captures domestic and international equity markets and attributes such as size and style. Together, these datasets provide a panoramic view of stock relationships to both idiosyncratic and universal variables. While the fundamental dataset is made available quarterly the other two are available at multiple frequencies: daily, weekly, and monthly. In this study, we only consider quarterly data. However, it's worth noting that the signatures are completely agnostic to the underlying frequency. Before conducting statistical analysis on the datasets, we recognize that the fundamental and economic & financial datasets may exhibit serial correlation, making them suitable candidates for examining lagged relationships with stocks. However, it's important to note that this lagged behavior may or may not apply to the study of ETFs. Despite this knowledge, we begin with the concatenated dataset because we are motivated to study the effectiveness of the signature genetic algorithm ('SigGA') model as a computational tool for dimension reduction.

4.4 Learning from Signatures Genetic Algorithm: Short and Long-Term Trends

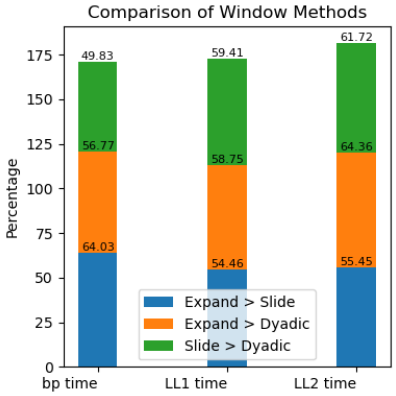
The signature's flexible architecture can be highly beneficial for stock trend analysis, behavior assessment, and subsequently stock classification. With regard to choosing a window method to compute the signatures, data-slicing techniques allow analysts to break down datasets into manageable and focused segments, which can reveal valuable insights for stock trend analysis. A fixed-length window considers a rolling, fixed-duration segment of the path for computing the signature. For example, we might compute the signatures over the most recent 2, 3, or 5 data points, sliding the window as time progresses. The short-term focus of the sliding window method helps capture local patterns and transient behaviors in the data as well as react quickly to changes in the dynamics (e.g., market shocks, abrupt shifts in trends), which can benefit momentum-based strategies or detect high-frequency fluctuations. If signatures from fixed windows show significant variability, this could indicate volatile or rapidly changing dynamics, and if they are stable over fixed windows, this may suggest a consistent local tend. The expanding window includes all data from the beginning up to the current time point. For example, at time t, the window includes all data from t_0 to t. This method promotes long-term focus because it captures cumulative trends and global features of the path as well as smooths out short-term noise to focus on the overarching trajectory, which can be good for mean-reversion strategies or identifying structural trends. Using both fixed-length and expanding windows together can help capture multi-scale dynamics. Fixed windows detect short-term signals that might be noise or reflect temporary shifts while expanding windows detect persistent, long-term changes that might be masked in short-term analysis. This approach is useful in financial time-series modeling, where short-term volatility often overlays long-term trends [17]. The dyadic window [18] refers to a specific partitioning of time intervals, such as subdivisions of time into halves, quarters, eighths, etc., defining windows over which the signature of a path is computed. By splitting the path into dyadic windows, the signature can be computed piecewise, and the results can be combined to represent the entire path. This simplifies the computation of signatures for long paths by reducing the integral domains into smaller, manageable intervals. Dyadic windows allow for hierarchical or multi-scale analysis of the path, enabling efficient approximation schemes or analysis at different resolutions, and are useful tools in studying path regularity. They are fundamental in constructing higher-order rough paths, where detailed local behavior of the path at different time scales matters. A dyadic partition divides the time interval $[0, T]$ into subintervals of length $W_k^n = T/2^n$, where n is the dyadic level. For example, at $n = 2$, the time interval $[0, T]$ is divided into four intervals: $[0, T/4], [T/4, T/2], [T/2, 3T/4].[3T/4, T]$. Let $T > 0$ be the total time interval, and let $n \in \mathbb{N}$ denote the dyadic level, where the union of all dyadic windows at a given level n forms the entire interval $[0, T] = \bigcup_{k=0}^{2^{n-1}} W_k^n$. The dyadic partition of $[0, T]$ at level n is given by:

$$\mathcal{P}_n = \left\{ \left[\frac{kT}{2^n}, \frac{(k+1)T}{2^n} \right] : k = 0, 1, \ldots, 2^n - 1 \right\}$$

With table 1 in mind, we use 10 latent variables in the Partial Least Squared regressions, except for the first case $n_{latent} = 6$. For each stock, the initial setup of our SigGa involves running 27 regressions (3 window methods, 3

augmentation methods, and 3 signature levels $1, 2, 3$), each of 100 generations, where each generation consists of 100 populations. This amounts to $270,000$ regressions per stock.

Comparison of window methods: The table and bar chart in Figure 5 compare stocks' adj-R^2 of the 100th generation of a window method, by counting the number of times they exceed the adj-R^2 of the competing method. This comparison is done for all 101 stocks under consideration, across basepoint and lead-lag levels 1 & 2 augmentations. The results show that the expanding window method has more stocks with higher adj-R^2 than the sliding and dyadic methods, with the sliding window outperforming the dyadic. This could be attributed to the fact that for sliding and dyadic methods, we utilized very similar window lengths of 5 and 6, respectively. Nonetheless, these results suggest the relevance of compartmentalizing stocks with the highest adj-R^2 values obtained from the expanding window method. Appendix H figure 10 shows the results of the window methods where each augmentation is expanded across degrees. They suggest deteriorating adj-R^2 values of the expanding window method at higher degrees (number highlighted in red). The dyadic method's adj-R^2 tends to improve as the signature's level increases (number highlighted in blue).



(a) across Augmentation

Aug \ Window	E > S	E > D	S > D
BP time	64	57	50
LL1 time	54	59	59
LL2 time	55	64	62
ave.	**58**	**60**	**57**

(b) Expand (E), Slide (S), Dyadic (D)

Figure 5: Window method across augmentation

Comparison of augmentation methods: The results in Appendix H Figure 11 suggest that incorporating lead-lag augmentation improves the adjusted R^2 across all windows, although lead-lag at level 2 does not add much value over lead-lag 1. Additionally, the sliding window benefits the most from the lead-lag transformation.

Comparison of signature levels: The results in Appendix H Figure 12 suggest that adjusted R^2 values increase as signature levels rise from 1 to 2 and 3, with degree 2 outperforming 3 for both sliding and dyadic window methods. In other words, while both sliding and dyadic windows perform better at higher degrees, level 2 signatures are optimal. On the other hand, the first-level signatures' adjusted R^2 decrease as signature levels rise from 1 to 2 and 3, with degree 3 outperforming degree 2 under the expanding method.

Conclusion – signature-learned configuration: Select the top-performing stocks based on adjusted R^2 values using the expanding window method with lead-lag 1 and first-order signatures. Evaluate the remaining stocks using the sliding window method with lead-lag 1 and second-order signatures, and select the top-performing stocks in terms of adjusted R^2 values. Finally, evaluate the remaining stocks using the dyadic window method with lead-lag 1 and signatures at levels 2 and 3, while changing the dyadic window length from 6 to 30.

4.5 Learning from SigGa: Market, Economic, and Fundamental Trends

The hypothesis posits that improved stock prediction accuracy is contingent upon increasingly refined categorization of stocks, their influencing factors, and the associated linear and non-linear causal and interaction effects. This holds true within the framework of the weak form of the Efficient Market Hypothesis (EMH). Recall that the weak-form EMH precludes profit generation solely through the exploitation of historical stock prices or technical analysis, given their public availability. However, it remains agnostic regarding the potential for profit derived from fundamental analysis or insights into specific or general economic or financial conditions. The semi-strong form extends this prohibition to include fundamental analysis, while the strong form precludes all forms of price-predictive information, including insider knowledge. If market inefficiencies exist, stock trading biases can arise from the systematic (common to the entire financial system) and idiosyncratic (unique to a single institution) processing of circumstantial (stochastic or deterministic) information. In this section, we use SigGa results to analyze the relevance of the concatenated factor set of three datasets: market, economic/financial, and fundamental. For each stock, we select the results of the 100th

generation associated with each of the window methods (sliding, expanding, dyadic) and each of the augmentation methods (basepoint time, lead-lag 1 time, lead-lag 2 time) all computed under the signature levels of $1, 2$ and 3. This yields 27 SigGA outputs per stock (2727 per 101 stocks) where each output consists of stock's adj-R^2, 5-gene chromosome (final five selected factors), and the stock identifier i.e. ticker. Tables 2 and 3 provide summary statistics of the adjusted R^2 of the three datasets combined per window method. Table 3 removes the five worst adj-R^2 observations, which belong to Starbucks (SBUX) and Silicon Valley Bank (SIVB) stocks. The results in the tables below grouped by window method, were produced by regressing stocks' returns against signature coefficients evaluated under various configurations. Overall, the SigGa model selected factors that consistently exhibited moderate signal-to-noise ratio with an average adj-R^2 of 29% and an average count of 80% of the results being positive.

Statistics \ Window	Slide	Expand	Dyadic
count	909	909	909
mean	0.26	0.29	0.21
std	0.56	0.58	0.90
min	-10.81	-9.14	-18.99
25%	0.11	0.13	0.09
50%	0.33	0.38	0.30
75%	0.53	0.56	0.48
max	0.88	0.94	0.88
# $obs. < 0$	155	150	166
% $obs. < 0$	21	20	22

Table 2: Summary entire sample

Statistics \ Window	Slide	Expand	Dyadic
count	904	904	904
mean	0.29	0.32	0.27
std	0.30	0.35	0.30
min	-0.99	-1.82	-1.31
25%	0.12	0.14	0.09
50%	0.33	0.38	0.30
75%	0.53	0.56	0.49
max	0.88	0.94	0.88
# $obs. < 0$	150	145	161
% $obs. < 0$	20	19	22

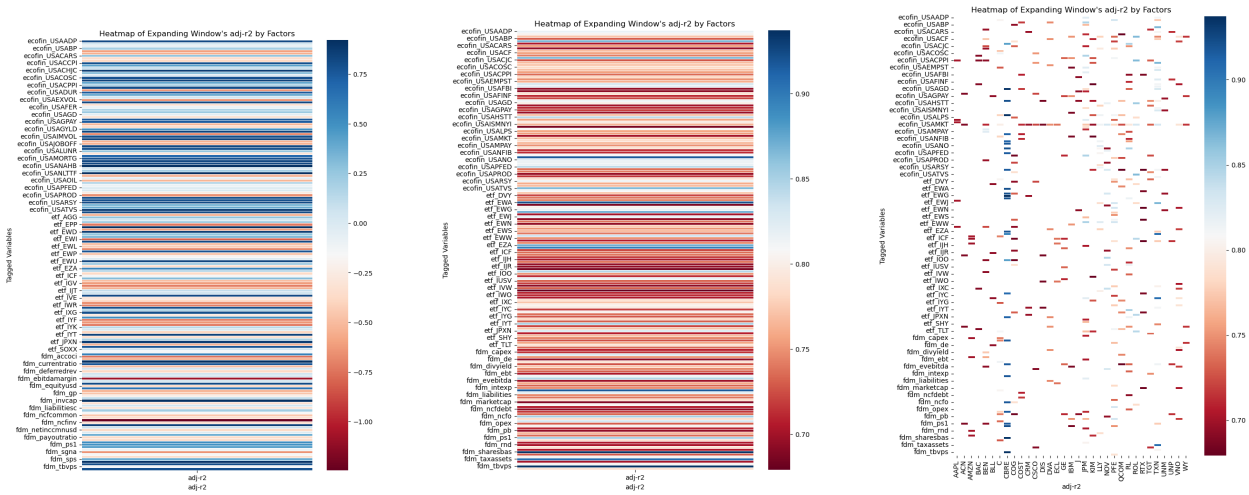
Table 3: Summary ex 5 worst observations

We use the heatmap and one-hot-encoder methods to understand the categorical distribution of the 5-gene chromosome results produced by the SigGa. The former is a graphical representation of data where values are depicted as colors in a matrix, often used to visualize correlations, relationships, or patterns within data. The latter counts how often each tagged variable appears across all gene columns. It is a technique used to represent categorical data in numerical format by creating new binary features (columns) for each category within a categorical variable. The corresponding feature is set to 1 for a given category, while all others are set to 0. One-hot-encoder can be more informative than a heatmap because it gives a more focused view of the frequency occurring features in the dataset, especially when showing only those that appear more than a specified number of times such as three or four etc., occurences across all gene columns. Considering the expanding window method, Figure 6a depicts the top and bottom fifty observations in terms of their adjusted R-squared values. The darker the color and the wider the stripe the larger the values. The blue and red colors indicate positive and negative values respectively. The macroeconomic data, i.e., the category labeled Ecofin, is predominantly represented by the dark blue color. The fundamental data (i.e., the category labeled FDM) appears to be lighter in colors (both blue and red) than the ETF data, although it exhibits small patches of dark red color. Figure 6b depicts the top 100 observations in terms of their adjusted R-squared values. The blue and red colors indicate positive values, with blue representing more positive than red. The macroeconomic and fundamental data appear to have higher adjusted R-squared than the ETF data. Figure 6c is the same as 6b but adds stocks to the picture. We observe that stocks have a diversified selection of factors, with the stock market indicator (USAMKT) being selected by most stocks.

Using the expanding window, the one-hot encoding method identified 185 factors that occurred at least once among the top 100 stocks sorted by their adjusted R-squared. Remember, each stock had five factors selected by SigGa. This suggests that to achieve a high signal-to-noise ratio (adj-R^2 of 0.71), SigGa utilized nearly all of the 244 available factors per stock. Table 4 presents the percentage count of those factors appearing three times or more by category within the top, center, and bottom-ranked adjusted R-squared values. The macroeconomic (ECOFIN) data is selected the most, and it shows decreasing occurrences as we move from top to bottom (50% to 40%), while ETF and fundamental (FDM) data exhibit increasing occurrences (33% to 37%) and (17% to 23%) respectively. Comparing the ETF data to the FDM data, the former accounted for a higher proportion of selected factors than the latter.

Category \ count occurrence >2	Top 100	Center 100	Bottom 100	Bottom 140
ECOFIN	50	45	44	40
ETF	33	37	40	37
FDM	17	18	16	23

Table 4: % count by category across top, center, bottom adj-R^2, expand window



(a) Top/bottom (50) total of (100) (b) Top (100) (c) Top (100) across stocks

Figure 6: Expanding window Heatmap by factors and stocks

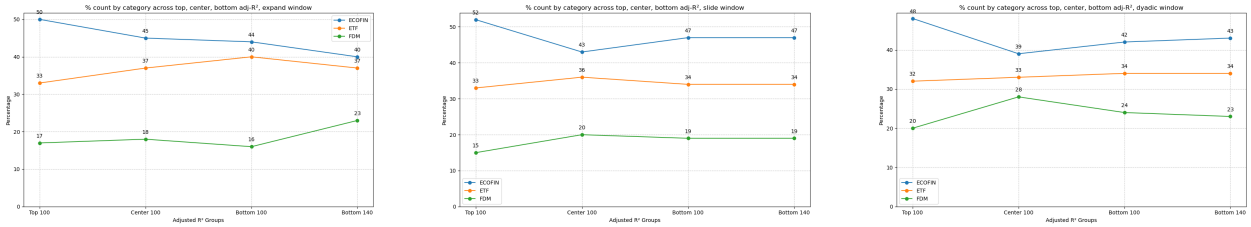
Tables 5 and 6 expand on the information presented in Table 4. Instead of counts, they display symbols of factors appearing four or more times in the top-ranked and bottom-ranked adjusted R-squared values, respectively, for the expanding window method. Symbols in blue indicate factors common to both top and bottom rankings. In summary, the SigGa results demonstrate a moderate signal-to-noise ratio (32%) can be achieved, as evidenced by the adjusted R-squared values of the relationship between stock returns and a diverse set of selected factors represented by a 5-gene chromosome. While the macroeconomic category was predominant, the five-gene chromosomes for most, if not all, stocks incorporated factors from at least two categories. This diversity underscores the importance of a multifaceted approach to risk factors in achieving robust predictive power, where the composition of the 5-gene chromosomes, representing the selected factors, varied significantly across all stocks. Figure 7 shows the results of Table 4, Table 10, and Table 11. Table 11 in Appendix I shows an increasing proportion of FDM chosen factors, especially for the center 100 ranked adj-R^2. This suggests as stocks' paths become more irregular and exhibit higher-order relationships, they tend to have a relatively higher tilt toward fundamental factors.

Ecofin	Ecofin (cont.)	ETF	FDM
USACARS	USAGPAY	AGG	ebitdausd
USACCPI	USAGYLD	DVY	ev
USACF	USALC	EWD	payables
USACJC	USAMKT	EWJ	ppnenet
USACNCN	USAMPAY	ICF	ps1
USACPMI	USANHS	IEF	
USACPPI	USAOPT	IUSG	
USAEMPST	USAPHS	IXC	
USAFOET	USAPSAV	IYE	
USAGD	USATOT	SOXX	
		TLT	

Table 5: Top 100, count occurrence > 3. expand wd.

Ecofin	Ecofin (cont.)	ETF	FDM
USABOT	USAIMPX	DVY	accoci
USABR	USAJCLM	EPP	ev
USACCONF	USALC	EWK	evebitda
USACNCN	USAMKT	EWL	grossmargin
USACOR	USANATGSC	EWM	ncfcommon
USAFER	USANO	EWP	opinc
USAFOET	USAOPT	EWS	ps1
USAGD	USAPCEPI	EWW	sharesbas
USAGSP	USAPFED	EWY	tbvps
USAGYLD	USAPHS	EWZ	
	USAPPIC	FXI	
	USAPROD	IBB	
	USATOT	ICF	
		IJS	
		IXC	
		IXP	
		IYH	
		IYM	
		IYW	
		TIP	

Table 6: Bottom 100, count occurrence > 3, expand wd.



(a) expand (b) slide (c) dyadic

Figure 7: One-hot encoding categories

4.6 Literature Overview: Linear and Nonlinear Models on Macroeconomic and Fundamental Data

Academic studies [19] on the adjusted R^2 of linear regression models for stock returns against fundamental and macroeconomic data generally report that the adjusted R^2 values are positive but low, the predictive power of fundamental data reside in few factors, and macroeconomic variables provide modest improvements to the adjusted R^2. Most studies find that the linear regression models typically explain a small fraction of the variance in stock returns, with adjusted R^2 values ranging between 1% and 10%. Positive adjusted R^2 values indicate that these models explain more variance than would be expected by chance, but their explanatory power is limited due to the inherent noise and randomness in stock returns. Fundamental variables like price-to-earnings ratio (P/E), book-to-market ratio (B/M), dividend yields, and cash flows have been shown to have some explanatory power of stock returns. In [20] Fama and French (1992) found that the book-to-market ratio and size factors had a statistically significant relationship with stock returns, with adjusted R^2 values in the low single digits. In [21] Campbell and Shiller (1998) showed that dividend-price and earning-price ratios could explain stock returns, but again with relatively low R^2. The macroeconomic variables like GDP growth, interest rates, inflation, and unemployment rates provide modest improvements to the adjusted R^2. In [22] Chen, Roll, and Ross (1986) found that macroeconomic factors like industrial production, interest rates, and inflation could explain some variation in stock returns, but the R^2 remained low. In [23] Lettau and Ludvidson (1999) introduced a consumption-wealth ratio (CAY) that explained some stock return variability with an adjusted R^2 around $5\% - 7\%$. In a slightly different context, the cross-sectional studies e.g., Fama-MacBeth regressions [24] often yield higher adjusted R^2 values for explaining differences in returns across stocks than time-series regressions focused on predicting returns for a single stock or index over time. Four main reasons could explain the low adjusted R-squared in linear or nonlinear models: a) The high noise-to-signal ratio in stock returns, characterized by high volatility and significant random fluctuations, severely hinders the ability of any model to explain their variability. b) The omitted variables issue, as linear models often fail to capture other relevant factors (e.g., investor sentiment, market microstructure effects) that drive stock returns. c) The nonstationarity issue in fundamental and macroeconomic relationships with stock returns can vary over time, making consistent predictions difficult. d) The nonlinear relationships of stocks with factors since macroeconomic and fundamental data may have nonlinear [25] or interaction effects with stock returns, which linear models cannot capture [26]. Nonlinear models (e.g., polynomial regression [27], machine learning algorithms [28]) tend to achieve higher R^2 values when compared to linear models, but the improvements are typically modest. Nonlinear transformations of fundamental and macroeconomic [29] data (e.g., interaction terms or log transformation) can slightly improve the adjusted R^2 but not dramatically. For instance, nonlinear approaches that account for regime changes or thresholds (e.g., threshold models, Markov switching models) may yield higher R^2 values. Overall, polynomial and nonlinear regressions can capture relationships missed by linear models, particularly when stock returns respond to fundamental and macroeconomic variables in a non-monotonic way. Algorithms like random forests, support vector machines, and neural networks [30] often report higher R^2 values when compared to traditional nonlinear regressions. However, the reported adjusted R-squared, still tends to remain relatively low to modest, indicating that fundamental and macroeconomic data capture only a fraction of the variation in stock returns.

The implications of low adjusted R-squared is consistent with the efficient market hypothesis (EMH), which posits that most available information is already incorporated into stock prices (or returns), leaving little room for predictability. Additionally, it underscores the challenge of explaining or predicting stock returns using only fundamental and macroeconomic data. In conclusion, while linear models often yield positive adjusted R-squared values when using fundamental and macroeconomic data, the values are generally small ($1\% - 10\%$). These results highlight the limited explanatory power of linear models in capturing the complex, dynamic, and noisy nature of stock returns. A list of relevant papers covering domestic and international markets under linear models is provided in the references section [31] [32] [33] [34] [35] [36] [37] [38].

5 Polymodels General Description

The mathematical concept of polymodels generalizes the one-dimensional univariate method to multidimensional, multivariable analysis. Theoretically, a polymodel entails four steps: a) An unsupervised learning procedure that operates on each risk factor independently b) A nonlinear basis formulation or data transformation c) A supervised learning regression of each risk factor against each dependent variable e.g: stock returns d) A statistical measure or evaluation function such as expectation or variance (value-at-risk/expected shortfall). In the polymodels approach, a random variable Y_t is dependent on some collection of variables X_t through some nonlinear functions f. Consequently, polymodels offer flexibility in specification (domain-specific dependencies) and application in various fields. In particular, polymodels allow for the specification of different functions for different sets of variables or domains [39]. It can handle varying relationships and interactions between variables in various domains, providing a tailored approach to modeling dependencies. For instance, in a multi-domain or multi-input variables system, you might need different mathematical functions to describe relationships in each domain. This flexibility is crucial when dealing with complex systems with varying dependencies in various domains. In addition, this capability is beneficial in fields such as machine learning, statistics, finance, and engineering, where models often need to accommodate diverse and complex relationships between variables. More precisely, given a set of $n = 1, \ldots, N$ variables, approximate the output Y_t by a collection of functions $F_1, \ldots, F_N \in F_n : \mathbb{R} \to \mathbb{R}$ for any nonlinear function f_{nk}. Then, find a nonlinear approximation of the output Y_t through a multivariable function $F(X_1, \ldots, X_N)$ joining all functions $F_1(X_1), \ldots, F_N(X_N) \in F : \mathbb{R}^N \to \mathbb{R}$ 46 guaranteeing dependency between variables. To simplify the notation, set $Y \equiv Y_t, X \equiv X_t$:

$$Y \simeq F_n(X_n) \qquad F_n(X_n) = \sum_{k=0}^{K} a_k f_k(X_n) \tag{45}$$

$$F(X_1, \ldots, X_N) = \sum_{n=1}^{N} F_n(X_n) = \sum_{n=1,k=0}^{N,K} a_{nk} f_{nk}(X_n) \qquad Y \simeq F(X_1, \ldots, X_N) \tag{46}$$

5.1 Formulation

The above expressions state for every output variable $Y_i,\ i = 1, \ldots, I \in \mathbb{N}$ and a collection of factors $X,\ n = 1, \ldots, N \in \mathbb{N}$, find an ordered subset $\mathcal{J}_i \subseteq \mathcal{J} \in X_{J_i} := Y_i$ estimated through the nonlinear specification φ_n using a cost function $\mathcal{C}$. This study takes φ_n as the univariate function of monomials $f_k(x_n) := x^k$ 45, and the minimum squared errors as the cost function. The multi-factor function φ is an aggregate equation (mean, minimum, sum) consisting of a set of independent variables x and a nonlinear function φ_n linking x to y, plus an error term ϵ. Compactly:

$$y = \varphi(x_1, \ldots, x_N) = \begin{cases} \varphi_1(x_1) + \epsilon_1 & \text{for } x_1 \in \mathcal{J}_1 \\ \varphi_2(x_2) + \epsilon_1 & \text{for } x_2 \in \mathcal{J}_2 \\ \quad \vdots & \\ \varphi_N(x_N) + \epsilon_N & \text{for } x_N \in \mathcal{J}_N \end{cases} \tag{47}$$

$$y = \varphi(x_1, \ldots, x_N) + \epsilon \qquad y = \sum_{n=1,k=0}^{N,K} \beta_{nk} f_{nk}(x_n) + \epsilon \tag{48}$$

However, this formulation is ill-conditioned due to colinearity in the powers of x, and autocorrelation in the variables type (nontraded, traded). Hence, we must obtain an orthogonal base(s) before estimating as described in Section 5.5

5.2 Scale, Compute Returns

Scale all input levels and return data to the same base. Denote the assets and integer time indices by $i \in \{1, 2, \ldots, I\}$ and $t \in \{1, 2, \ldots T\}$. Denote standardized returns on stock i by $R_i, \forall\, i \in \mathcal{I} = \{1, 2, \ldots, I\}$ and standardized returns on factor j by $F_j, \forall j \in \mathcal{J} = \{1, 2, \ldots, J\}$. The natural logarithm returns of stocks and factors r, f respectively. T corresponds to the most recent observation and is indexed to the number of observations. Set $Y_i \equiv R_i$ and $X_j \equiv F_j$. For each stock and risk factor, we compute a divider as follows:

$$
\begin{aligned}
p_{i,t=1:T} &= levels, prices \\
l_{i,t=1:T} &= |p_{i,t=1:T}| \\
d_{i=1:I,T} &= 10^{log_{10}(\max(l_{i,t=1:T}))} \\
L_{i,t=1:T} &= 100 + \frac{p_{i,t=1:T}}{d_{i=1:I,T}}
\end{aligned}
$$

$$r_{i,t} = \ln(L_{i,t}) - \ln(L_{i,t-1}) \qquad f_{i,t} = \ln(L_{i,t}) - \ln(L_{i,t-1}) \tag{49}$$

$$R_{i,t} = \frac{r_{i,t} - \overline{\mu}}{\sigma} \qquad F_{i,t} = \frac{f_{i,t} - \overline{\mu}}{\sigma} \tag{50}$$

$$\sigma = \sqrt{\frac{1}{\tau}\sum_{i=1}^{T}(x_{i,t} - \overline{\mu})^2} \qquad \overline{\mu} = \frac{1}{\tau}\sum_{t=1}^{T} x_{i,t}$$

Equivalently, the expression in 5.1 can be written as:

$$R = \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}} \beta_{j,k} F_j^k + \epsilon_i \tag{51}$$

5.3 Risk Factor Long-Term Distribution

For each series[3], obtain the parameterized probability distribution at the cutoff points 1%, 16%, 50%, 84%, and 99% of the cumulative distribution function, by quantile for the central points and power law for the tail points. Intuitively, the quantile function associates with a range at and below a probability input the likelihood that a random variable is realized in that range for some probability distribution. The power law distribution associates positive and negative tail values with exponential growth at and beyond extreme probability events (1% 99%).

Take $f_X(x)$ the PDF probability density function of a continuous random variable x :

$$f_X(x) = \lim_{x \to 0^+} \frac{P(x < X \leq x + \Delta)}{\Delta} \qquad P(x < X \leq x + \Delta) = F_X(x + \Delta) - F_X(x) \tag{52}$$

By the Fundamental Theorem of Calculus, if $F_X(x)$ is differentiable at x :

$$f_X(x) = \frac{\partial F_X(x)}{\partial x} = \frac{\partial}{\partial x}\left[\int_{-\infty}^{x} f_X(t)dt\right] = F'_X(x) \qquad F_X(x) = \int_{-\infty}^{x} f_X(t)dt \tag{53}$$

Definition: For a given value x, CDF is the probability P that a random variable X is less than or equal to x :

$$F_X(x) = P(X \leq x) \qquad F_X(x + \Delta) - F_X(x) = \int_{x}^{x+\Delta} f_X(x)dx \tag{54}$$

Denote $D_{l,q}$ the set of all finite partitions of $[l, q]$:

$$D_{l,q} = \{(q_0, ..., q_k) \,|\, k > 0,\, l = q_0 < q_1 < ... < q_{k-1} < q_k = q\} \tag{55}$$

For a finite population of N equally probable indices $1, \ldots N$ arranged from highest to lowest, the k^{th} q-quantile can be computed via $I_p = Nk/q$, For continuous r.v. x, $k/q = p$. If I_p is not an integer, then round up to the next integer to

[3] standardized log difference returns.

get the appropriate index; the corresponding data value is the k^{th} q-quantile. If I_p is an integer, then the quantile can be any number from the data value at that index to the data value of the next.

For a given $p \in (0,1)$, the k^{th} quantile is the value x where the distribution function crosses (jumps over) p; is the value x such that its probability is less than or equal to an input probability. For a discrete r.v. where $p = k/q$, the q-quantile partitions (ordered division) the distance between the largest and smallest values into q subsets of (nearly) equal size and probability. Hence, x is the k^{th} q-quantile for a variable X if:

$$P(X < x) \leq p \, (\equiv P(X \geq x) \geq 1 - p) \quad \& \quad P(X \leq x) \geq p \tag{56}$$

The q-quantile is a generic term, the sample's median is the 2-quantile, the quartile is the 4-quantile (four groups), etc, and percentiles are the 100-quantile (100 groups). There are $(q-1)$ partitions of the q-quantiles, one for each integer k satisfying $0 < k < q$, e.g. 4-quantile partitions at three points 25%, 50%, 75% resulting in four $\approx$ identical groups.

Definition: Given a continuous and strictly monotonic cumulative distribution function F_X, q-quantile Q_X of a random variable X is the inverse of the CDF function, mapping the input $p \in \{1/q, 2/q, ..., (q-1)/q\}$ to a threshold value x so that the probability of X being less than or equal to x is greater or equal to p. In terms of the distribution function F_X, the quantile Q_X returns the value x such that:

$$F_X : \mathbb{R} \to [0,1] \quad F_X(x) := \; P(X \leq x) = p \quad \& \quad Q_X : [0,1] \to \mathbb{R} \quad Q_X(p) := F_X^{-1}(p) = x \tag{57}$$

Definition: The PDF distribution of the tail is given by the indicator function $1_T(x)$ which collects return values above (below) a certain threshold q such that the cumulative distribution function is less than or equal to p respectively $(1-p)$. The function $1_T(x)$ is the union set of highest (positive) and lowest (negative) returns, excluding the center points:

$$1_T(x) = 1_{H \cup L} = max\{1_H, 1_L\} = 1_H + 1_L - 1_H \cdot 1_L \tag{58}$$

Thus, the indicator function $1_H : X \to \{0, x\}$ of tail observations subset H of factor returns set $\{X\}$:

$$1_H(x) := \begin{cases} x & if \; x > \max(0, Q_X(p)) \\ 0 & otherwise \end{cases} \tag{59}$$

Similarly, the indicator function $1_L : -X \to \{0, -x\}$ of the tail observations subset L of factor returns set $\{-X\}$:

$$1_L(-x) := \begin{cases} -x & if \; -x > \max(0, Q_{-X}(p)) \\ 0 & otherwise \end{cases} \tag{60}$$

The percentile rank R_X of a given value x is the percentage of values in its frequency distribution that are lower than or equal to it. The percentile ranks are not on an equal-interval scale:

$$R_X = \frac{F_X(x) - (0.5 \times n_i)}{N} \times 100 \tag{61}$$

Definition: Tail probabilities are percentile rank statistics, where $F_X(x)$ is given by the index count I_p, and n_i the frequency is 1. The function $\lceil . \rceil$ rounds up to the nearest integer in $\mathbb{Z}^+$.

$$\tilde{p} = \frac{I_p - 0.5}{N} \qquad\qquad n = N - \lceil Np \rceil \tag{62}$$

$$N = \text{sample size} \quad n = \text{tail size}$$

Example: $N = 51,\ p = 0.8,\ n = N - \lceil N \times p \rceil = 51 - 41 = 10.$
$I_p + 1 = \lceil N \times p \rceil + 1 = 42,\ \tilde{p}_{42} = \frac{42-0.5}{51} = 0.81,\ \tilde{p}_{51} = \frac{51-0.5}{51} = 0.99$

5.3.1 Power Law Parameterized Distribution

Take the cumulative distribution function of the exponential(λ), where λ is the intensity and $1/\lambda$ the expected value:

$$F(x;\lambda) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0, \\ 0 & x < 0, \end{cases} \tag{63}$$

Using results from previous sections; substituting Q for x we can write:

$$1 - e^{-\lambda x} = F(x;\lambda) = p \quad \& \quad Q(p;\lambda) = x \quad \longrightarrow \quad 1 - e^{-\lambda Q} = p \tag{64}$$

Taking the natural logarithm, we get the quantile function for exponential(λ):

$$Q(p;\lambda) = \frac{-ln(1-p)}{\lambda} \tag{65}$$

Consider the tail indicator function $\{1_T(x)\} := x$ and probabilities $\tilde{p}$. Write an OLS regression equation in the coefficients α and β with $\epsilon \mid X,\ \epsilon \mid Z \sim \mathcal{N}(0,\, \sigma^2\mathbf{I})$:

$$ln(1-\tilde{p}) = \alpha + \beta \times ln\{1_T(x)\} + \epsilon \tag{66}$$

Rearrange and get a formula for x as a function of α and β:

$$\begin{aligned} &ln(1-\tilde{p}) = \alpha + \beta \times ln(x) \\ &ln(1-\tilde{p}) - ln(e^{\alpha}) = ln(x^{\beta}) \\ &ln(\frac{1-\tilde{p}}{e^{\alpha}}) = ln(x^{\beta}) \\ &\frac{1-\tilde{p}}{e^{\alpha}} = x^{\beta} \iff x^{-\beta} = \frac{e^{\alpha}}{(1-\tilde{p})} \\ &x = \left[\frac{e^{\alpha}}{(1-\tilde{p})}\right]^{-1/\beta} \end{aligned} \tag{67}$$

Obtain formulae for quantile estimators corresponding to positive and negative tails, respectively, under power law by:

$$\omega_{\phi \leq 1\%} := Q_X(p;\lambda) = \left[\frac{e^{\alpha}}{(1-p)}\right]^{-1/\beta} \qquad \omega_{\phi \geq 99\%} := Q_{-X}(p;\lambda) = -\left[\frac{e^{\alpha}}{(1-p)}\right]^{-1/\beta} \tag{68}$$

5.3.2 Value-at-Risk, Quantile Estimator

For each risk factor, derive a parameterized probability distribution independently of stocks at five cut-off points 1%, 16%, 50%, 84%, and 99%. Extreme values are captured through the power law quantile estimator while the percentile function captures the central points.

Definition: Let X be the risk factor standardized[4] return series (negative, positive), the VaR at level $\phi \in (0,1)$ is the smallest (respectively the largest) number y such that $P(Y := -X \leq y) \geq 1 - \phi$ ($P(Y := X \geq y) \geq \phi$):

$$VaR_{\phi}(X) = -inf\{x \in \mathbb{R} : Q_X^{-1}(\phi) := F_X(x) > \phi\} = Q_Y(1-\phi) = F_Y^{-1}(1-\phi) \tag{69}$$

Definition: The function Ω concatenates sets of power law and percentile quantile estimator values at $1\%, 99\%$, and $16\%, 50\%, 84\%$ confidence levels respectively, as follows:

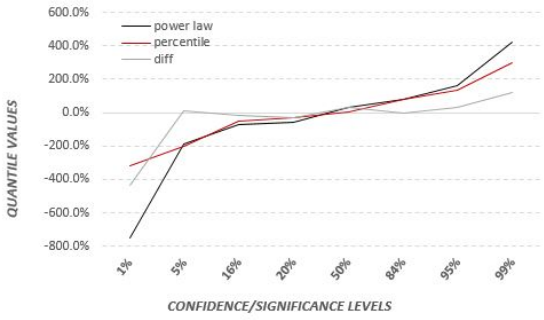
[4] all factors are scaled to the same base 100, before computing standardized returns.

$$\omega(x;\phi) = \begin{cases} -\left[\frac{e^{\alpha}}{(1-(1-\phi)}\right]^{-1/\beta} & \text{for } \phi \leq 1\%, \\ \left[\frac{e^{\alpha}}{(1-\phi)}\right]^{-1/\beta} & \text{for } \phi \geq 99\%, \\ Q_X(1-\phi) = F_X^{-1}(1-\phi) & \text{for } 1\% < \phi < 99\% \end{cases} \tag{70}$$

For an arbitrary measure $\phi \in \mathbb{R}^+$, the re-parameterized five points distribution of each risk factor is:

$$\Omega = \{\phi \in (0,1) : \omega_{\phi \leq 1\%} \cup \omega_{1\% < \phi < 99\%} \cup \omega_{\phi \geq 99\%}\} \tag{71}$$

Figure 8a shows a [Balance Sheet] risk factor with the identifier "accoci", a component of [EQUITY] representing the accumulated change in equity from transactions and other events.



(a) quantile vs. percentile

C/S levels	power law	percentile	difference
1%	-749	-317%	-432%
16%	-71%	-52%	-19%
50%	34%	5%	29%
84%	80%	80%	0%
99%	422%	301%	121%

(b) quantile estimator confidence (significance) levels

5.4 Polynomial Interpolation

We want to approximate the output values of a variable Y using some non-linear function $\varphi(.)$ of a certain variable X. Given $\mathcal{P}_n$ the set of all polynomials of degree n in one variable and real coefficients, consider a basis $\{p_0, \ldots, p_n\}$ such as the monomials basis $P_i(x) = x^i$. Formally, given $Y : [a,b] \to \varphi$ and points $\{x_0, x_1 \ldots, x_T\}$ for $T > n+1$ satisfying $a \leq x_0 < x_1 < \cdots < x_T \leq b$, determine a polynomial $\varphi_n \in \mathbb{R}_n[X]$ such as:

$$Y_t = \varphi_n(x_t) \quad \forall\, t = 0, 1, \ldots, T \tag{72}$$

Choose φ_n as a linear combination of $(n+1)$ basis functions $p_0(x), \ldots, p_n(x)$. By condition (72) get $\Phi c = Y$ and solve for the coefficients $c_0, \ldots, c_n$, where Φ is $(T \times n+1)$ Vandermonde matrix V_x, which is ill-conditioned due to co-linearity, auto-correlation, and larger values of n:

$$\varphi_n(x_t) = c_0 p_0(x_t) + c_1 p_1(x_t) \ldots + c_n p_n(x_t) = \sum_{i=0}^{n} c_i p_i(x_t) \tag{73}$$

$$\begin{pmatrix} p_0(x_0) & p_1(x_0) & \ldots & p_n(x_0) \\ p_0(x_1) & p_1(x_1) & \ldots & p_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ p_0(x_T) & p_1(x_T) & \ldots & p_n(x_T) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_T \end{pmatrix} = \begin{pmatrix} Y_0 \\ Y_1 \\ \vdots \\ Y_T \end{pmatrix} \tag{74}$$

5.4.1 Orthogonal Basis:

We are interested in polynomials of at most degree $n = 4$ enough to capture the skewness and kurtosis statistics (3^{rd} and 4^{th} moments). We chose basis functions depending on the underlying risk factor's distribution[39], non-traded assets (Gram-Schmidt) and traded assets (Hermite). A basis that can be easily inverted, integrated, and differentiated is paramount in building multivariate models from univariate models. To determine the nonlinear function $\varphi(.)$, orthogonalize each risk factor by the Hermite polynomials generation function 77. The resulting basis $\mathcal{H}$ consists of orthogonal vectors $\{H_0, \ldots, H_n\}$ corresponding to the matrix Q in $V_x = QR$ a.ka. Gram-Schmidt decomposition, with dimension $(T \times (n+1))$, where T is the total number of observations, and $(n+1)$ columns of polynomial terms:

$$\mathcal{H}_n(x) = [H_0 \quad H_1 \quad \ldots \quad H_n] \tag{75}$$

Definition 5.1. A sequence of polynomials $\{\varphi_i(x)\}_{i=0}^n$ in one variable and real coefficients where deg $\varphi_n = n$, and $\varphi_0 \neq 0$, is orthogonal on $[a, b]$ with respect to weight function $w(x) > 0$, if the inner product $\langle \varphi_n, \varphi_m \rangle = 0$ for $n \neq m$ or $\langle \varphi_n, \varphi_m \rangle = h_n$ nonzero constant, and $\delta_{nm} = 1$ otherwise. If $h_n = 1 \ \forall n$, the set of functions is ***orthonormal*** $\forall\, m, n = 1, 2 \ldots, n \leq m - 2$. The Hermite orthogonal polynomials defined on $[a, b] = (-\infty, \infty),\ w(x) = e^{-x^2/2}$:

$$\langle \varphi_n, \varphi_m \rangle = \int_a^b \varphi_n(x)\varphi_m(x)w(x)dx = \int_{-\infty}^{\infty} e^{\frac{-x^2}{2}} H_n(x)H_m(x)dx = \begin{cases} n!\sqrt{2\pi}\delta_{nm} \neq 0 & n = m \\ 0 & n \neq m \end{cases} \tag{76}$$

Then, generate the orthogonal **Hermite basis** by:

$$H_n(x) = (-1)^n e^{\frac{x^2}{2}} \frac{d^n}{dx^n}(e^{\frac{-x^2}{2}}) = \left(x - \frac{d}{dx}\right)^n .1 \qquad \varphi_n(x) = \frac{1}{(n!\sqrt{2\pi})^{1/2}} H_n(x) \tag{77}$$

For $w(x) = e^{-x^2/2}, [a, b] = [-\infty, \infty]$, the first five orthogonal Hermite polynomials $H_n(x)$, normalized by $\sqrt{n!}$ are:

$$H_{e0}(x) = 1 \tag{78}$$
$$H_{e1}(x) = x \tag{79}$$
$$H_{e2}(x) = (x^2 - 1)/\sqrt{2} \tag{80}$$
$$H_{e3}(x) = (x^3 - 3x)/\sqrt{6} \tag{81}$$
$$H_{e4}(x) = (x^4 - 6x^2 + 3)/\sqrt{24} \tag{82}$$

Example 1. For $w(x) = 1, [a, b] = [-1, 1]$ obtain the Legendre system of orthogonal polynomials with general form:

$$\varphi_n(x) = \frac{1}{2^n} \frac{d^n}{dx^n}(x^2 - 1)^n \tag{83}$$

$$\begin{aligned} \varphi_0(x) &= 1 \\ \varphi_1(x) &= x \\ \varphi_2(x) &= x^2 - \frac{1}{3} \\ \varphi_3(x) &= x^3 - \frac{3}{5}x \end{aligned} \tag{84}$$

Adjusted Hermite: The values of variables in the powers of x increase exponentially with $i,\ (x^i, i = 1, \ldots n)$. To adjust for aberration in variable (tail) values, interpolate the Hermites of large negative and positive observations by the first-order derivative of the functions $H_n(x)$. Derive thresholds using $\{\alpha, \beta\}$ the constant and slope Pareto parameters

by 5.3.1 of both tails $\pm \left[e^{\alpha}/(1-p)\right]^{-1/\beta}$. Then interpolate the bottom and top 20% i.e.: $1-p=0.2$ observations, that is, $\forall\ x \in (-x < -\tau)$ and $(x > \rho)$ compute $\overleftarrow{H}$ and $\overrightarrow{H}$ respectively. The general forms of the interpolated $\overleftarrow{H}_n(-x < -\tau)$ and Hermite polynomial $H_n(x) := H_{(x<-\tau)} \cup H_{(x\geq -\tau\ \&\ x\leq\rho)} \cup H_{(x>\rho)}$:

$$\overleftarrow{H}_n(-x) = \left[H_n(-\tau) - H_n^{'}(-\tau) \times (-\tau)\right] + H_n^{'}(-\tau) \times (-x) \tag{85}$$

$$\overleftarrow{H}_0(-x) = 1 \tag{86}$$

$$\overleftarrow{H}_1(-x) = -x \tag{87}$$

$$\overleftarrow{H}_2(-x) = -\tau^2 - 1 + 2\tau x \tag{88}$$

$$\overleftarrow{H}_3(-x) = 2\tau^3 - 3\tau^2 x + 3x \tag{89}$$

$$\overleftarrow{H}_4(-x) = -3\tau^4 + 6\tau^2 + 3 + 4\tau^3 x - 12\tau x \tag{90}$$

$$H_n(x) = \overleftarrow{H}_n(x) \cup \overline{H}_n(x) \cup \overrightarrow{H}n(x) \tag{91}$$

5.5 Estimation

This section illustrates a single-factor nonlinear polynomial approach with regularization and cross-validation fine-tuning of weights, sample set, and parameters, before generalizing to the multivariable polymodel (multi-factor models) setup. Define the basis function φ_n as the weighted average of the orthogonal polynomial vector of n^{th} order $H_n[x] \in \mathcal{H}$ in 75. Then, combine with 72, add an error term to get:

$$\begin{aligned} y &= \varphi_n(x_n) + \epsilon \\ &= \sum_{k=0}^{K} a_k x^k + \epsilon = \sum_{k=0}^{K} a_k f_k(x) + \epsilon \end{aligned} \tag{92}$$

$$y = \sum_{k=0}^{K} \alpha_k H_k(x) + \epsilon \tag{93}$$

By definition, we have $\langle f, g\rangle = \langle g, f\rangle = f^T g = \sum_j f_j g_j = \int_a^b f(x)g(x)dx$. Minimize the sum of square residuals between the orthogonal polynomial φ_n in basis $H_0, \ldots, H_n \in \mathbb{R}_n[X]$ and y, i.e.: find α_i such that $\partial E/\partial\alpha_i = 0$. This leads to the $n+1$ systems of equations in the coefficients, called normal equations, in matrix form:

$$\begin{aligned} E(\alpha_0, \ldots, \alpha_n) &= \int_a^b (y - \varphi(x))^2 dx \\ &= \langle y, y\rangle - 2\sum_{i=0}^{n} \alpha_i \langle y, H_i\rangle + \sum_{i=0}^{n}\sum_{j=0}^{n} \alpha_i\alpha_j \langle H_i, H_j\rangle \end{aligned} \tag{94}$$

$$\begin{pmatrix} \langle H_0, H_0\rangle & \langle H_0, H_1\rangle & \ldots & \langle H_n, H_n\rangle \\ \langle H_1, H_0\rangle & \langle H_1, H_1\rangle & \ldots & \langle H_1, H_n\rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle H_n, H_0\rangle & \langle H_n, H_1\rangle & \ldots & \langle H_n, H_n\rangle \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix} = \begin{pmatrix} \langle y, H_0\rangle \\ \langle y, H_1\rangle \\ \vdots \\ \langle y, H_n\rangle \end{pmatrix} \tag{95}$$

Equivalently, $\langle y, H_i\rangle = \sum_{j=0}^{n} \alpha_j \langle H_i, H_j\rangle \rightarrow \alpha_i = \langle H_i, H_i\rangle^{-1}\langle y, H_i\rangle$. Or compactly, $e = y - H\alpha \rightarrow e'e = (y - H\alpha)'(y - H\alpha) = y'y - 2\alpha' H'y + \alpha' H'H\alpha$. Take $\frac{\partial}{\partial\alpha}\left[E(\alpha_0, \ldots, \alpha_n)\right]\big|_{=0}$ then obtain:

$$\hat{\alpha} = (H'H)^{-1}H'y \tag{96}$$

5.5.1 Regularization and Cross-validation

The main idea behind Ridge regularization, also known as L^2 regularization, is to shrink the coefficients towards zero but not exactly zero, thus preventing overfitting caused by higher-order polynomial terms [40]. Regularization may also help improve the generalization of the model to unseen data. It involves adding a shrinkage term λ to the equation of coefficients $\hat{\alpha}$. Furthermore, $\lambda = \{0.1, \ldots, 10.5\}$ with 0.5 increments is typically multiplied by the identity matrix or any other square matrix the analyst deems appropriate. For example, in our case, we set Ω as a penalty term proportional to a certain magnitude of higher-order polynomials. Assuming x is a Gaussian copula [41] we can compute the normalized Hermite with $w(x) = e^{-x^2/2}$ as in expression 127 and 77, factor out the divisor $n!$ from ψ_i to obtain the diagonal elements of the diagonal matrix $\Omega = [0, 1, 2, 6, 24]$, see appendix C. Although Ridge regularization introduces bias into the estimates, it can significantly reduce the variance (bias-variance trade-off), leading to a lower overall mean squared error (MSE) on unseen data. Ultimately, Ridge regularization helps stabilize the coefficient estimates by mitigating issues arising from highly correlated predictors (multicollinearity). Then the penalized residuals sum of squares (RSS) equation for estimating the coefficients can easily be derived from the linear counterpart: $L(\alpha) = \sum_{i=1}^{\tau}(y_i - x_i^T\alpha)^2 + \lambda\sum_{j=1}^{p}\alpha_j^2 \; \forall \, i, j = 1, \ldots, \tau, p \in \mathbb{N}$ observations and predictors respectively. As a result of this operation, given a $\lambda = 21$ penalty multipliers and a sample of τ observations, the regression produces 21 coefficient estimates. Denote $J(\alpha)$ the nonlinear form in matrix notation:

$$J(\alpha) = (\mathbf{y} - \mathbf{H}\alpha)^T(\mathbf{y} - \mathbf{H}\alpha) + \lambda\alpha^T\mathbf{\Omega}\alpha \tag{97}$$

$$\hat{\alpha}(\lambda) = \arg\min_{\alpha} J(\alpha) = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{\Omega})^{-1}\mathbf{H}^T\mathbf{y} \tag{98}$$

Training and tuning the model parameters to unseen data happens with cross-validation. One can find many ways to divide the data into training and testing samples. First, decide on a reasonable number of observations to use for testing, e.g. 20%, partition the data into κ folds (5 =1/0.2) of approximately equal length $\ell \approx \tau/\kappa$ testing sets such as $\{1 \ldots \tau\} = \{1, \ldots, \ell\} \cup \{\ell + 1, \ldots, 2\ell\} \cup \cdots \cup \{(\kappa - 1)\ell + 1, \ldots, \tau\}$ their union covers the entire τ domain. Then, for each κ fold use a training subsample $(\tau - \ell)$ to estimate the coefficients, and the remaining testing ℓ observations to compute the mean squared error. Formally, given the finite partitions $\mathcal{D}_i$ such as $i \subset [1, \tau] \; \forall i = 1, \ldots, d$ and d is the total number of folds, denote, $\mathcal{D}_{1,\tau} = \{(t_1, \ldots, t_{\kappa\ell}) \,|\, \ell \approx \tau/\kappa, \; 1 = t_1 < \cdots < t_\ell < \cdots < t_{\tau-\ell+1} < \cdots < t_\tau = \tau\}$, $\{\nu_i, \kappa_i\}$ training and testing sets respectively. The cross-validated out-of-sample (testing sample) MSE of the κ^{th} fold is η_κ: (see alsoD):

$$\eta_\kappa(\lambda) = \frac{1}{\ell}\sum_{(x,y)\in\mathcal{D}_\kappa}\left[\frac{y - \hat{H}_\kappa(x, \lambda)}{1 - tr[I_\kappa]/\ell}\right]^2 = \frac{1}{\ell}\sum_{i=1}^{\ell}\left[\frac{y_i - \hat{H}_\kappa(x_i, \lambda)}{1 - tr[I_\kappa]/\ell}\right]^2 \tag{99}$$

$$\hat{H}_\kappa(x, \lambda) = \hat{H}_\nu(y|x, \lambda)H_\kappa(x) = \frac{\partial}{\partial\alpha}\left[E_\nu(\alpha_0, \ldots, \alpha_n)\right]\Big|_{=0} \times H_\kappa(x) \tag{100}$$

$$I_\kappa(x, \lambda) = H_\kappa(H_\kappa^T H_\kappa + \lambda\Omega)^{-1}H_\kappa^T \tag{101}$$

$$\hat{y}_\kappa(x, \lambda) = I_\kappa(\lambda)y = \hat{H}_\kappa(x, \lambda) \tag{102}$$

The function $\hat{H}_\nu(y|x, \lambda)$ represents the estimated coefficients of the system using the training data, which is the same as 98, where the first term, the regularized inverse matrix is computed twice; once to estimate the coefficients using training data and another to compute the fisher information matrix I_κ using testing data. The function $H_\kappa(x)$ corresponds to the Hermite transformation matrix $(\ell \times n)$ of the test data (out-of-sample). Therefore, the predicted values are captured by $\hat{H}_\kappa(x, \lambda)$. Generally, the value of λ controls the strength of the regularization. A higher value of λ increases the penalty on the magnitude of the coefficients, leading to more shrinkage. The optimal λ is typically chosen through cross-validation. Finally, select the coefficients associated with the MSE of all the λ across all the folds κ. Subsequently, assess the statistical significance of each minimum squared error $\eta_\kappa(\lambda)$ using adjusted $\overline{R^2}$ and F test statistics, i.e.: analysis of variance (ANOVA table).

$$R^2 = \frac{\sum \hat{y}^2}{\sum y^2} = 1 - \frac{\sum e^2}{\sum y^2} \tag{103}$$

$$\overline{R^2} = 1 - \frac{(\sum e^2)/(\ell - k - 1)}{(\sum y^2)/(\ell - 1)} \tag{104}$$

$$F = \frac{(\sum \hat{y}^2)/(k)}{(\sum e^2)/(\ell - k - 1)} \tag{105}$$

6 Portfolio Construction and Results

To be completed

7 Conclusion

To be completed

References

[1] A. Cherny, Douady, and S. Molchanov, "On measuring nonlinear risk with scarce observations," in *Finance Stoch (2010) 14: 375–395*. Springer-Verlag 2009, 2009, pp. DOI 10.1007/s00 780–009–0107–y, received: 25 March 2008 / Accepted: 15 January 2009 / Published online: 7 November 2009.

[2] A. Fermanian, "Learning time-dependent data with the signature transform," in *En vue de l'obtention du doctorat de Sobonne Université, Ecole Doctoral Sciences Mathematiques de Paris Center*. Laboratoire de Probabilités Statistique and Modélisation, 15 Oct 2021.

[3] T. Lyons and A. D. McLeod, "Signature methods in machine learning," in *University of Oxford, Radcliffe Observatory, A. W. Bld, Woodstock Rd, Oxford, OX2 6CG, UK*. arXiv:2206.14674v5, [stat.ML], 29 Jan 2024.

[4] I. Chevyrev and A. Kormilitzin, "A primer on the signature method in machine learning," in *Mathematical Institute, University of Oxford, A. W. Bld, Woodstock Rd, Oxford, OX2 6CG, UK*. arXiv:1603.037888v1, [stat.ML], 11 Mar 2016.

[5] B. Hambly and T. J. Lyons, "Uniquenss for the signature of a path of bounded variation and the reduced path group," in *Mathematical Institute, Oxford University, 24-29 St. Giles, Oxford OX1 3LB, England*. arXiv:math/0507536v2, [math.CA], 19 Dec 2006.

[6] P. Foster, "A brief introduction to path signatures," in *The content of this notebook draws substantially from the excellent primer by Chevyrev and Kormilitzin (2016)*. The Alan Turing Institute, 26 Jun 2020, copyright 2020 Peter Foster.

[7] ChatGPT and version 4 released 2024, "What is rectilinear transformation math," in *OpenAI*. An artificial intelligence (AI) research company, 28 Oct 2024.

[8] G. FLint, B. Hambly, and T. Lyons, "Discretely sampled signals and the rough hoff process," in *Mathematical Institute, University of Oxford, Woodstock Road, OX2 6GG, UK*. arXiv:2308.15135v2 [q-fin.PM], 02 Dec 2015.

[9] ChatGPT and version 4 released 2024, "Give me a mathematical definition of tensor products of vector spaces," in *OpenAI*. An artificial intelligence (AI) research company, 31 Oct 2024.

[10] C. Cuchiero, G. Gassani, and S. Svaluto-Ferro, "Signature-based models: theory and calibration," in *Vienna University, Department of Statistics and OR, University of Verona, Department of Economics*. arXiv:2207.13136v1 [q-fin.PM], 28 Jul 2022, financial support by the FWF project I 3852 and through grant Y 1235 of the FWF START-program.

[11] ChatGPT and version 4 released 2024, "How does chen identify in rough path theory concatenates two paths," in *OpenAI*. An artificial intelligence (AI) research company, 09 Nov 2024.

[12] Y. INAHAMA, "Rough path theory and stochastic calculus," in *Graduate School of Mathematics, Kyushu University, Motooka 744, Nishi-ku, Fukuoka,819-0395, Japan*. arXiv:1602.03255v1, [math.PR], 10 Feb 2016.

[13] ChatGPT and version 4 released 2024, "Give me a thorough mathematical definition of the signatures of the rough path theory," in *OpenAI*. An artificial intelligence (AI) research company, 10 Nov 2024.

[14] O. Futter, B. Horvath, and M. Wiese, "Signature trading: A path-dependent extension of the mean-variance framework with exogenous signals," in *Imperial College London, University of Kaiserslautern, Departments of Mathematics, University of Oxford, Oxford-Man Insititute, The Alan Turing Institute*. arXiv:2308.15135v2 [q-fin.PM], 30 Aug 2023.

[15] G. Sanchez, E. Marzban, and P. AI, "All models are wrong: Concepts of statistical learning," in *Perplexity AI v3.2.0, released on November 26 2024*. 2020 Sanchez, Marzban. All Rights Reserved, 28 Nov 2024, give me the mathematical derivation of partial least squares in the case of one output variable.

[16] ChatGPT and version 4 released 2024, "What is the mathematical formulation of this python plsregression function," in *OpenAI*. An artificial intelligence (AI) research company, 28 Nov 2024.

[17] ——, "Does fixed length window vs. expanding window of signature of rough path theory tell us something about short term versus long term trends," in *OpenAI*. An artificial intelligence (AI) research company, 07 Jan 2025.

[18] ——, "What do you know about the dyadic window of the signatures of the rough path theory?" in *OpenAI*. An artificial intelligence (AI) research company, 07 Jan 2025.

[19] ChatGPT, GPT-4, and (v2) Jan 2025, "What do academic studies say about the adjusted r-squared of linear and nonlinear regression models of stock returns against fundamental data and macroeconomic data? do they say the average r squared is positive, negative, etc?" in *OpenAI*. An artificial intelligence (AI) research company, 19 Jan 2025.

[20] E. F. Fama and K. R. French, "The cross-section of expected stock returns," in *The Journal of Finance, Vol XLVII, No 2*. Graduate School of Business, University of Chicago, 1101 East 58th Street, Chicago, IL 60637, June 1992, research supported by National Science Foundation (Fama), and Center for Research in Security Prices (French).

[21] J. Y. Campbell and R. J. Shiller, "Stock prices, earnings and expected dividends," in *Econometric Research Program*. Research Memorandum No. 334, January 1998, princeton University, Yale University.

[22] N.-F. Chen, R. Roll, and S. A. Ross, "Economic forces and the stock market," in *Journal of Business, vol. 59, no. 3*. All rights reserved 0021-9398/86/5903-0001, 1986, university of Chicago, UCLA, Yale University.

[23] M. Lettau and S. Ludvigson, "Consumption, aggregate wealth and expected stock returns," in *Federal Reserve Bank of New York*. Research Department, June 8, 1999, 33 Liberty St. New York, NY 10045.

[24] E. F. Fama and J. D. MacBeth, "Risk, return, and equilibrium: Empirical tests," in *National Science Foundation*. University of Chicago, Sept 2, 1972.

[25] J. Bruin, "Stock prediction model with financial ratios and macroeconomic variables using machine learning," in *TILBURG UNIVERSITY, Netherlands*. Supervisor: Dr. S. Collin, Second reader: Dr. M. de Sisto, 03-11-2021.

[26] C. Hiemstra and C. Kramer, "Nonlinearity and endogeneity in macro-asset pricing," in *University of Strathclyde, International Monetary Fund*. MITPress, Quarterly Journal Vol. 2, No. 3, October 1997, studies in Nonlinear Dynamics and Econometrics.

[27] B. Cai and J. Gao, "A simple nonlinear predictive model for stock returns," in *Huazhong University of Science and Technology, China. Monash University, Australia*, October 14, 2017.

[28] M. Chen, M. X. Hanauer, and T. Kalsbach, "Design choices, machine learning, and the cross-section of stock returns," in *TUM School of Management, Technical University of Munich*, November 2024.

[29] M. Qi, "Nonlinear predictability of stock returns using financial and economic variables," in *Journal of Business & Economic Statistics*. Vol. 17, No. 4, pp. 419-429 (11 pages), October 1999, https://www.jstor.org/stable/1392399.

[30] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," in *University of Chicago Booth School of Business, Yale University*. AQR Capital Management, and NBER, September 13, 2019.

[31] A. Dahlman, "Macroeconomic variables and the stock market, a u.s. study of their relationship," in *Copenhagen Business School*. Master's Thesis (CFIVO1009E) - Contract No. 31435, May 15, 2023, supervisor, Peter Belling.

[32] A. Humpe and P. Macmillan, "Can macroeconomic variables explain long term stock market movements? a comparison of the us and japan," in *Center for Dynamic Macroeconomic Analysis*. University of St Andrews, Ocotober 2007, working paper series CDMA07/20.

[33] D. Filipović and P. Pasricha, "Empirical asset pricing via ensemble gaussian process regression," in *´École Polytechnique Fédérale de Lausanne and Swiss Finance Institute, Indian Institute of Technology*. arXiv:2212.01048v2 [q-fin.RM], January 2, 2025.

[34] R. Morck, B. Yeung, and W. Yu, "R-squared and the economy," in *National Bureau of Economic Research*. Working Paper 19017, May 2013, http://www.nber.org/papers/w19017.

[35] R. Priestley, "Short interest, macroeconomic variables and aggregate stock returns," in *Norwegian Business School*. SSRN, 30 May 2019, https://ssrn.com/abstract=3384620.

[36] E. Palmgren and N. Nanakorn, "The impact of macroeconomic variables on stock return in different industries - a multiple linear regression," in *Royal Institute of Technology School of Engineering Sciences*. Examiner at KTH: Jörgen Säve-Söderbergh, 08 2019, supervisors at KTH: Tatjana Pavlenko och Julia Liljegren.

[37] R. Seth and V. Tripathi, "Stock market performance and macroeconomic factors: The study of indian equity market," in *University of Delhi, Global Business Review*. 15(2) 291–316 © 2014 IMI SAGE Publications DOI: 10.1177/0972150914523599, May 2014, http://gbr.sagepub.com.

[38] R. Seethalakshmi, "Analysis of stock market predictor variables using linear regression," in *Sastra University, International Journal of Pure and Applied Mathematics*. Volume 119 No. 15, 369-378, ISSN: 1314-3395 (on-line version), January 2018, https://www.researchgate.net/publication/326253896.

[39] J. Zhang, "Statistical arbitrage based on stock clustering using nonlinear factor model," in *in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy*. Stony Brook University, 2019.

[40] Y. Guan, "Polymodel: Application in risk assessment and portfolio consstruction," in *in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy*. Stony Brook University, August 2019, quantitative finance, Applied Mathematics and Statistics.

[41] X. Ye, "Systemic risk indicators based on nonlinear polymodel," in *in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy*. Stony Brook University, May 2018, quantitative finance, Applied Mathematics and Statistics.

[42] T. H. Koornwinder, "Orthogonal polynomials, a short introduction," in *Korteweg-de Vries Institute, University of Amsterdam*. arXiv:1303.2825v2, 11 Nov 2021, math.CA.

A Appendix Auto-correlation

Our universe of risk factors consists of exchanged-traded funds and non-traded fundamental data. The latter suffers accuracy issues due to stale operations, e.g. delay or failed update. This implies that the correlation between a variable and its past values at different lags is not zero, and varies as a function of the time lag:

$$\rho_{xx}(\tau) = E[X_{t+\tau}\bar{X}_t] \qquad lag := \tau = (t_{i+1} - t_i) \qquad \rho_{xx}(0) = 1 \qquad \Theta_{xx}(0) = \sigma^2 \tag{106}$$

B Appendix Gram-Schmidt

Gram Schmidt basis for polynomials Let x be the random variable of a single risk factor data. Consider the vector space $\mathcal{P}$ of all polynomials in one variable and real coefficients of the sequence of monomials $x^0, x^1, x^2 \dots$ that are linearly independent. Let $\mathcal{X}$ be a vector subspace of $\mathcal{P}$ spanned by vectors $\{x^0, x^1, x^2 \dots, x^n\}$. Orthogonalize $\mathcal{P}$ w.r. to $\mathcal{Q}$ inner product space[42] a.k.a. Gram-Schmidt decomposes X into an upper triangular square matrix R and an orthonormal basis Q:

$$\langle x, y\rangle = x^T y = \sum_{j=1}^{n} x_j y_j \qquad \|x\| = \sqrt{\langle x, x\rangle} \qquad \text{proj}_p(x) = \frac{\langle x, p\rangle}{\langle p, p\rangle}\, p \tag{107}$$

Compute Q as follows:

$$p_0(x) = x^0 = 1 \qquad \to q_0 = \frac{1}{\|p_0\|} p_0 \tag{108}$$

$$p_1(x) = x^1 - \text{proj}_{p_0}\left(x^1\right) \qquad \to q_1 = \frac{1}{\|p_1\|} p_1 \tag{109}$$

$$p_2(x) = x^2 - \text{proj}_{p_0}\left(x^2\right) - \text{proj}_{p_1}\left(x^2\right) \qquad \to q_2 = \frac{1}{\|p_2\|} p_2 \tag{110}$$

$$\vdots$$

$$p_n(x) = x^n - \text{proj}_{p_0}\left(x^n\right) \cdots - \text{proj}_{p_{n-1}}\left(x^n\right) \qquad \to q_n = \frac{1}{\|p_n\|} p_n \tag{111}$$

The general forms in terms of p and q:

$$p_n(x) = x^n - \sum_{k=0}^{n-1} \frac{\langle x^n, p_k\rangle}{\langle p_k, p_k\rangle}\, p_k(x) \tag{112}$$

$$p_n(x) = x^n - \sum_{k=0}^{n-1} \langle x^n, q_k\rangle\, q_k(x) \tag{113}$$

This produces a mutually orthogonal sequence $p_0(x), p_1(x) \dots p_n(x)$ of polynomials in x, and $p_0(x) = 1$. The orthonormal basis $\{q_0, q_1 \dots q_n\}$ is the orthogonal unit length basis of constituent vectors:

$$q_{\mathrm{n}}^{\mathrm{T}} q_{\mathrm{m}} = \begin{cases} 1 & n = m \\ 0 & n \neq m \end{cases} \tag{114}$$

In fact, $p_n(x)$ is a linear combination[42] of $x^0, x^1 \dots x^n$, and

$$\langle p_n, p_j\rangle = \langle x^n, p_j\rangle - \sum_{k=0}^{n-1} \frac{\langle x^n, p_k\rangle}{\langle p_k, p_k\rangle} \langle p_k, p_j\rangle = \langle x^n, p_j\rangle - \frac{\langle x^n, p_j\rangle}{\langle p_j, p_j\rangle} \langle p_j, p_j\rangle = 0 \quad (j = 0, 1, \dots, n-1) \tag{115}$$

Express the projection equation, and deduce formulae for $\{x^0, x^1, x^2 \dots, x^n\}$ B in terms of q:

$$\text{proj}_q(x) = \langle x, q\rangle\, q \qquad\qquad \|p_j\| = \left\langle x^j, q_j \right\rangle \tag{116}$$

$$x^0 = \left\langle x^0, q_0 \right\rangle q_0 \tag{117}$$
$$x^1 = \left\langle x^1, q_0 \right\rangle q_0 + \left\langle x^1, q_1 \right\rangle q_1 \tag{118}$$
$$x^2 = \left\langle x^2, q_0 \right\rangle q_0 + \left\langle x^2, q_1 \right\rangle q_1 + \left\langle x^2, q_2 \right\rangle q_2 \tag{119}$$
$$\vdots$$
$$x^n = \langle x^n, q_0\rangle\, q_0 + \langle x^n, q_1\rangle\, q_1 + \cdots + \langle x^n, q_{n-1}\rangle\, q_{n-1} + \langle x^n, q_n\rangle\, q_n \tag{120}$$

Rewrite the above in matrix form, and obtain the upper triangular decomposition R:

$$\begin{bmatrix} x^0 & x^1 & x^2 & \dots & x^n \end{bmatrix} = \begin{bmatrix} q_0 & q_1 & q_2 & \dots & q_n \end{bmatrix} \begin{bmatrix} \|p_0\| & \left\langle x^1, q_0 \right\rangle & \left\langle x^2, q_0 \right\rangle & \dots & \langle x^n, q_0\rangle \\ 0 & \|p_1\| & \left\langle x^2, q_1 \right\rangle & \dots & \langle x^n, q_1\rangle \\ 0 & 0 & \|p_2\| & \dots & \langle x^n, q_3\rangle \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \|p_n\| \end{bmatrix} \tag{121}$$
$$\text{or, } X = QR$$

Or alternatively,

Gram-Schmidt orthogonalization Given $\{p_0, \dots p_n\}$ basis $\in \mathbb{R}_n[X]$ of vector subspace $\mathcal{P}$, where the degree of p_n is exactly n, obtain orthogonal $\varphi := \{\varphi_0, \dots, \varphi_n\}$ and orthonormal $\psi := \{\psi_0 \dots \psi_n\}$ bases relative to the inner product:

$$\begin{aligned} &\varphi_0(x) = p_0 = 1 \\ &\text{for } i = 1, \dots, n \\ &\varphi_i(x) = p_i - \sum_{j=0}^{i-1} \frac{\langle p_i, \varphi_j\rangle}{\langle \varphi_j, \varphi_j\rangle}\, \varphi_j(x) \end{aligned} \tag{122}$$

Given $\varphi_0, \dots \varphi_n$ set of orthogonal polynomials, the degree of the next orthogonal polynomial $\varphi_{n+1} = x\varphi_n(x)$ is $n+1$ since the degree of φ_n is exactly n. Create a basis of $\mathbb{R}_{n+1}[X]$ by Gram-Schmidt on $\varphi_0(x), \dots \varphi_n(x), x\varphi_n(x)$. Notice that $x\varphi_j(x) \in \mathbb{R}_{j+1}[X] \to \langle \varphi_n(x), x\varphi_j(x)\rangle = 0$ for $j < n-1$ substantially simplifying the G-Schmidt procedure:

$$\varphi_{n+1}(x) = x\varphi_n(x) - \sum_{j=0}^{n} \frac{\langle x\varphi_n(x), \varphi_j(x)\rangle}{\langle \phi_j(x), \varphi_j(x)\rangle}\, \varphi_j(x) \tag{123}$$

$$\begin{aligned} \langle x\varphi_n(x), \varphi_j(x)\rangle &= \int_a^b x\varphi_n(x)\varphi_j(x)w(x)dx \\ &= \int_a^b \varphi_n(x)x\varphi_j(x)w(x)dx \\ &= \langle \varphi_n(x), x\varphi_j(x)\rangle \end{aligned} \tag{124}$$

QR decomposition: Get $V = QR$ by B, Q matrix columns form an orthogonal basis spanned by the k columns of V, $dim(Q) = dim(V)$ with the number of columns $k = n+1$ and first column of Q the constant. The R matrix of coefficients of variations and co-variations of basis vectors (monomials) with projection vectors, is an upper triangular square matrix used to reconstruct V from Q, and guarantees that the first k columns of V can be represented as a linear combination of the first k columns of Q

$$
\begin{aligned}
&n = 0 \\
&H_0(x) = P_0(x) \\
&\varphi_0(x) = \frac{H_0(x)}{\sqrt{\int_a^b H_0^2(x) w(x) dx}}
\end{aligned}
\tag{125}
$$

$$
\begin{aligned}
&n = 1 \\
&H_1(x) = P_1(x) + \alpha_{10}\varphi_0(x) \\
&H_1(x) \perp \varphi_0(x) \rightarrow \int_a^b H_1(x)\varphi_0(x)w(x)dx = 0 \\
&0 = \int_a^b P_1(x)\varphi_0(x)w(x)dx + \alpha_{10}\int_a^b [\varphi_0(x)]^2 w(x)dx \\
&\alpha_{10} = -\int_a^b P_1(x)\varphi_0(x)w(x)dx \\
&\varphi_1(x) = \frac{H_1(x)}{\sqrt{\int_a^b H_1^2(x) w(x) dx}}
\end{aligned}
\tag{126}
$$

$$
\begin{aligned}
&n = i \\
&H_i(x) = P_i(x) + \sum_{j=i-1}^{n} \alpha_{ij}\varphi_j(x) \\
&H_i(x) \perp \varphi_j(x) \rightarrow \int_a^b H_i(x)\varphi_j(x)w(x)dx = 0 \\
&0 = \int_a^b \{P_i(x) + \alpha_0\varphi_0(x) + \alpha_{i1}\varphi_1(x) + \alpha_{i2}\varphi_2(x) + \cdots + \alpha_{i,i-1}\varphi_{i-1}(x)\}\varphi_j w(x)dx \\
&\varphi_i(x) \perp \varphi_j(x) \rightarrow 0 = \int_a^b P_i(x)\varphi_j w(x)dx + \alpha_{i,i-1}\int_a^b \varphi_{i-1}(x)\varphi_j w(x)dx \\
&0 = \int_a^b P_i(x)\varphi_j w(x)dx + \alpha_{ij}\int_a^b [\varphi_j(x)]^2 w(x)dx \\
&\alpha_{ij} = -\int_a^b P_i(x)\varphi_j(x)w(x)dx \\
&\varphi_i(x) = \frac{H_i(x)}{\sqrt{\int_a^b H_i^2(x) w(x) dx}}
\end{aligned}
\tag{127}
$$

C Appendix Regularization

Take $w(x) = e^{-x^2/2}, [a, b] = [-\infty, \infty]$. To understand where the factor $n!\sqrt{2\pi}$ comes from, calculate the normalized Hermite for $\varphi_0(x) = 1/\sqrt{\int_a^b 1^2.\, e^{-x^2/2}dx} = 1/\sqrt{2\pi}$ and for higher order φ_n:

$$
\lambda = \begin{cases} 0.1, \ldots, 10.5 & \text{regularization} \\ 0.5 & \text{with increments} \end{cases}
\qquad
\Omega = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 24 \end{bmatrix}
\tag{128}
$$

$$\varphi_1(x) = \frac{x}{\sqrt{\int_{-\infty}^{+\infty}(x)^2 w(x)dx}} = \frac{x}{\sqrt{2\pi}} \qquad \varphi_2(x) = \frac{x^2-1}{\sqrt{\int_{-\infty}^{+\infty}(x^2-1)^2 w(x)dx}} = \frac{x^2-1}{2\sqrt{2\pi}} \tag{129}$$

$$\varphi_3(x) = \frac{x^3-3x}{\sqrt{\int_{-\infty}^{+\infty}(x^3-3x)^2 w(x)dx}} = \frac{x^3-3x}{6\sqrt{2\pi}} \qquad \varphi_4(x) = \frac{x^4-6x^2+3}{\sqrt{\int_{-\infty}^{+\infty}(x^4-6x^2+3)^2 w(x)dx}} = \frac{x^4-6x^2+3}{24\sqrt{2\pi}} \tag{130}$$

D Appendix Cross Validation

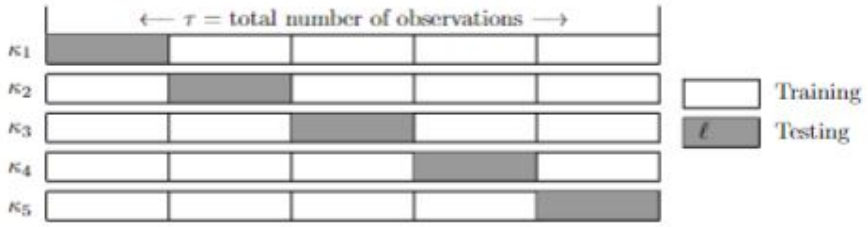
Define the finite partition $\mathcal{D}_{1,\tau} = \{(t_1, \ldots, t_{\kappa\ell}) \,|\, \ell \approx \tau/\kappa,\ 1 = t_1 < \cdots < t_\ell < \cdots < t_{\tau-\ell+1} < \cdots < t_\tau = \tau\}$. Denote ν_i training and κ_i testing sets $\forall i = 1, \ldots, d$ where d is the total number of partition. The model's overall cross-validated error is derived as follows:

$$MSE = \frac{1}{\kappa}(MSE_1 + MSE_2 + \cdots + MSE_\kappa)$$

$$D_1 : \nu_1 = \{\ell+1, \ldots, \tau\} \qquad \kappa_1 = \{1, \ldots, \ell\} \qquad MSE_1 = \frac{1}{\ell}\sum_{t=1}^{\ell}\epsilon_t^2$$

$$D_2 : \nu_2 = \{1, \ldots, \ell\} \cup \{2\ell+1, \ldots, \tau\} \qquad \kappa_2 = \{\ell+1, \ldots, 2\ell\} \qquad MSE_2 = \frac{1}{\ell}\sum_{t=\ell+1}^{2\ell}\epsilon_t^2$$

$$\xi^{(\lambda)} = \frac{1}{\kappa}\sum_{\kappa}\xi_\kappa^{(\lambda)} \tag{131}$$



(a) training vs. testing samples

E Appendix Fundamental Data Definition

Idx	Indicator	Title	Description
1	cashnequsd	Cash and Equivalents (USD)	[Balance Sheet] [CASHNEQ] in USD, converted by [FXUSD].
2	debtusd	Total Debt (USD)	[Balance Sheet] [DEBT] in USD, converted by [FXUSD].
3	equityusd	Shareholders Equity (USD)	[Balance Sheet] [EQUITY] in USD, converted by [FXUSD].
4	taxassets	Tax Assets	[Balance Sheet] A component of [ASSETS] representing tax assets and receivables.

5	ppnenet	Property, Plant & Equipment Net	[Balance Sheet] A component of [ASSETS] representing the amount after accumulated depreciation, depletion and amortization of physical assets used in the normal conduct of business to produce goods and services and not intended for resale.
6	inventory	Inventory	[Balance Sheet] A component of [ASSETS] representing the amount after valuation and reserves of inventory expected to be sold, or consumed within one year or operating cycle, if longer.
7	intangibles	Goodwill and Intangible Assets	[Balance Sheet] A component of [ASSETS] representing the carrying amounts of all intangible assets and goodwill as of the balance sheet date, net of accumulated amortization and impairment charges.
8	investments	Investments	[Balance Sheet] A component of [ASSETS] representing the total amount of marketable and non-marketable securities, loans receivable, and other invested assets.
9	receivables	Trade and Non-Trade Receivables	[Balance Sheet] A component of [ASSETS] representing trade and non-trade receivables.
10	accoci	Accumulated Other Comprehensive Income	[Balance Sheet] A component of [EQUITY] representing the accumulated change in equity from transactions and other events and circumstances from non-owner sources, net of tax effect, at period end. Includes foreign currency translation items, certain pension adjustments, unrealized gains and losses on certain investments in debt and equity securities.
11	retearn	Accumulated Retained Earnings (Deficit)	[Balance Sheet] A component of [EQUITY] representing the cumulative amount of the entity's undistributed earnings or deficit. May only be reported annually by certain companies, rather than quarterly.
12	taxliabilities	Tax Liabilities	[Balance Sheet] A component of [LIABILITIES] representing outstanding tax liabilities.
13	deferredrev	Deferred Revenue	[Balance Sheet] A component of [LIABILITIES] representing the carrying amount of consideration received or receivable on potential earnings that were not recognized as revenue, including sales, license fees, and royalties, but excluding interest income.
14	deposits	Deposit Liabilities	[Balance Sheet] A component of [LIABILITIES] representing the total of all deposit liabilities held, including foreign and domestic, interest and non-interest bearing. May include demand deposits, saving deposits, Negotiable Order of Withdrawal, and time deposits, among others.
15	payables	Trade and Non-Trade Payables	[Balance Sheet] A component of [LIABILITIES] representing trade and non-trade payables.
16	assetsnc	Assets Non-Current	[Balance Sheet] Amount of non-current assets, for companies that operate a classified balance sheet. Calculated as the difference between Total Assets [ASSETS] and Current Assets [ASSETSC].
17	assets	Total Assets	[Balance Sheet] Sum of the carrying amounts as of the balance sheet date of all assets that are recognized. Major components are [CASHNEQ], [INVESTMENTS], [INTANGIBLES], [PPNENET], [TAXASSETS], and [RECEIVABLES].
18	liabilities	Total Liabilities	[Balance Sheet] Sum of the carrying amounts as of the balance sheet date of all liabilities that are recognized. Principal components are [DEBT], [DEFERREDREV], [PAYABLES], [DEPOSITS], and [TAXLIABILITIES].

19	assetsc	Current Assets	[Balance Sheet] The current portion of [ASSETS], reported if a company operates a classified balance sheet that segments current and non-current assets.
20	debtc	Debt Current	[Balance Sheet] The current portion of [DEBT], reported if the company operates a classified balance sheet that segments current and non-current liabilities.
21	investmentsc	Investments Current	[Balance Sheet] The current portion of [INVESTMENTS], reported if the company operates a classified balance sheet that segments current and non-current assets.
22	liabilitiesc	Current Liabilities	[Balance Sheet] The current portion of [LIABILITIES], reported if the company operates a classified balance sheet that segments current and non-current liabilities.
23	debtnc	Debt Non-Current	[Balance Sheet] The non-current portion of [DEBT] reported if the company operates a classified balance sheet that segments current and non-current liabilities.
24	investmentsnc	Investments Non-Current	[Balance Sheet] The non-current portion of [INVESTMENTS], reported if the company operates a classified balance sheet that segments current and non-current assets.
25	liabilitiesnc	Liabilities Non-Current	[Balance Sheet] The non-current portion of [LIABILITIES], reported if the company operates a classified balance sheet that segments current and non-current liabilities.
26	ebitdausd	EBITDA (USD)	[Metrics] [EBITDA] in USD, converted by [FXUSD].
27	ps1	Price to Sales Ratio	[Metrics] An alternative calculation method to [PS], that measures the ratio between a company's [PRICE] and its [SPS].
28	pe1	Price to Earnings Ratio	[Metrics] An alternative to [PE] representing the ratio between [PRICE] and [EPSUSD].
29	ebt	Earnings before Tax	[Metrics] Earnings Before Tax is calculated by adding [TAXEXP] back to [NETINC].
30	ev	Enterprise Value	[Metrics] Enterprise value is a measure of the value of a business as a whole, calculated as [MARKETCAP] plus [DEBTUSD] minus [CASHNEQUSD].
31	fcf	Free Cash Flow	[Metrics] Free Cash Flow is a measure of financial performance calculated as [NCFO] minus [CAPEX].
32	fcfps	Free Cash Flow per Share	[Metrics] Free Cash Flow per Share is a valuation metric calculated by dividing [FCF] by [SHARESWA].
33	grossmargin	Gross Margin	[Metrics] Gross Margin measures the ratio between a company's [GP] and [REVENUE].
34	invcap	Invested Capital	[Metrics] Invested capital is an input into the calculation of [ROIC], and is calculated as: [DEBT] plus [ASSETS] minus [INTANGIBLES] minus [CASHNEQ] minus [LIABILITIESC]. Please note this calculation method is subject to change.
35	bvps	Book Value per Share	[Metrics] Measures the ratio between [EQUITY] and [SHARESWA].
36	evebitda	Enterprise Value over EBITDA	[Metrics] Measures the ratio between [EV] and [EBITDAUSD].
37	evebit	Enterprise Value over EBIT	[Metrics] Measures the ratio between [EV] and [EBITUSD].
38	de	Debt to Equity Ratio	[Metrics] Measures the ratio between [LIABILITIES] and [EQUITY].
39	pb	Price to Book Value	[Metrics] Measures the ratio between [MARKETCAP] and [EQUITYUSD].
40	tbvps	Tangible Assets Book Value per Share	[Metrics] Measures the ratio between [TANGIBLES] and [SHARESWA].

41	ps	Price Sales (Damodaran Method)	[Metrics] Measures the ratio between a company's [MARKETCAP] and [REVENUEUSD].
42	ebitdamargin	EBITDA Margin	[Metrics] Measures the ratio between a company's [EBITDA] and [REVENUE].
43	netmargin	Profit Margin	[Metrics] Measures the ratio between a company's [NETINCCMN] and [REVENUE].
44	marketcap	Market Capitalization	[Metrics] Represents the product of [SHARESBAS], [PRICE] and [SHAREFACTOR].
45	sps	Sales per Share	[Metrics] Sales per Share measures the ratio between [REVENUEUSD] and [SHARESWA].
46	payoutratio	Payout Ratio	[Metrics] The percentage of earnings paid as dividends to common stockholders. Calculated by dividing [DPS] by [EPSUSD].
47	currentratio	Current Ratio	[Metrics] The ratio between [ASSETSC] and [LIABILITIESC], for companies that operate a classified balance sheet.
48	tangibles	Tangible Asset Value	[Metrics] The value of tangible assets calculated as the difference between [ASSETS] and [INTANGIBLES].
49	workingcapital	Working Capital	[Metrics] Working capital measures the difference between [ASSETSC] and [LIABILITIESC].
50	sharesbas	Shares (Basic)	[Entity] The number of shares or other units outstanding of the entity's capital or common stock or other ownership interests, as stated on the cover of related periodic report (10-K/10-Q), after adjustment for stock splits.
51	ncff	Net Cash Flow from Financing	[Cash Flow Statement] A component of [NCF] representing the amount of CF (outflow) from financing activities, from continuing and discontinued operations. Principal components of financing CF are: issuance (purchase) of equity shares, issuance (repayment) of debt securities, and payment of dividends, other cash distributions.
52	ncfi	Net Cash Flow from Investing	[Cash Flow Statement] A component of [NCF] representing the amount of CF (outflow) from investing activities, from continuing and discontinued operations. Principal components of investing CF are: capital (expenditure) disposal of equipment [CAPEX], business (acquisitions) disposition [NCFBUS] and investment (acquisition) disposal [NCFINV].
53	ncfo	Net Cash Flow from Operations	[Cash Flow Statement] A component of [NCF] representing the amount of CF (outflow) from operating activities, from continuing and discontinued operations.
54	ncfdiv	Dividends Payment, Other Cash Distributions	[Cash Flow Statement] A component of [NCFF] representing dividends and their equivalents paid on common stock, and restricted stock units.
55	ncfcommon	Issuance (Purchase) of Equity Shares	[Cash Flow Statement] A component of [NCFF] representing the net CF (outflow) from common equity changes. Includes additional capital contributions from share issuances and exercise of stock options, and outflow from share repurchases.
56	ncfdebt	Issuance (Repayment) of Debt Securities	[Cash Flow Statement] A component of [NCFF] representing the net cash inflow (outflow) from issuance (repayment) of debt securities.
57	ncfbus	NCF - Business Acquisitions, Disposals	[Cash Flow Statement] A component of [NCFI] representing the NCF (outflow) associated with the acquisition, disposal of businesses, joint-ventures, affiliates, and other named investments.

58	ncfinv	NCF - Investment Acquisitions, Disposals	[Cash Flow Statement] A component of [NCFI] representing the NC inflow (outflow) associated with the acquisition, disposal of investments, including marketable securities and loan originations.
59	capex	Capital Expenditure	[Cash Flow Statement] A component of [NCFI] representing the net cash inflow (outflow) associated with the acquisition, disposal of long-lived, physical, intangible assets that are used in the normal conduct of business to produce goods and services and are not intended for resale. Includes cash inflows/outflows to pay for construction of self-constructed assets, software.
60	sbcomp	Share Based Compensation	[Cash Flow Statement] A component of [NCFO] representing the total amount of noncash, equity-based employee remuneration. This may include the value of stock or unit options, amortization of restricted stock or units, and adjustment for officers' compensation. As noncash, this element is an add back when calculating net cash generated by operating activities using the indirect method.
61	ncfx	Effect of Exchange Rate Changes on Cash	[Cash Flow Statement] A component of Net Cash Flow [NCF] representing the amount of increase (decrease) from the effect of exchange rate changes on cash and cash equivalent balances held in foreign currencies.
62	depamor	Depreciation, Amortization, and Accretion	[Cash Flow Statement] A component of operating CF representing the aggregate net amount of depreciation, amortization, and accretion recognized during an accounting period. As a non-cash item, the net amount is added back to net income when calculating cash provided by or used in operations using the indirect method.
63	ncf	NCF - Change in Cash Cash Equivalents	[Cash Flow Statement] Principal component of the CF statement representing the amount of increase (decrease) in cash and cash equivalents. Includes [NCFO], investing [NCFI] and financing [NCFF] for continuing and discontinued operations, and the effect of exchange rate changes on cash [NCFX].
64	ebitusd	Earning Before Interest, Taxes (USD)	[Income Statement] [EBIT] in USD, converted by [FXUSD].
65	epsusd	Earnings per Basic Share (USD)	[Income Statement] [EPS] in USD, converted by [FXUSD].
66	netinccmnusd	Net Income Common Stock (USD)	[Income Statement] [NETINCCMN] in USD, converted by [FXUSD].
67	revenueusd	Revenues (USD)	[Income Statement] [REVENUE] in USD, converted by [FXUSD].
68	rnd	Research and Development Expense	[Income Statement] A component of [OpEx] representing the aggregate costs incurred in a planned search or critical investigation aimed at discovery of new knowledge with the hope that such knowledge will be useful in developing a new product or service.
69	sgna	Selling, General and Administrative Expense	[Income Statement] A component of [OpEx] representing the aggregate total costs related to selling a firm's product and services, as well as all other general and administrative expenses. Direct selling expenses (for example, credit, warranty, and advertising) are expenses that can be directly linked to the sale of specific products. Indirect selling expenses are expenses that cannot be directly linked to the sale of specific products, for example telephone expenses, Internet, and postal charges. General and administrative expenses include salaries of non-sales personnel, rent, utilities, communication, etc.

70	gp	Gross Profit	[Income Statement] Aggregate revenue [REVENUE] less cost of revenue [COR] directly attributable to the revenue generation activity.
71	taxexp	Income Tax Expense	[Income Statement] Amount of current income tax expense (benefit) and deferred income tax expense (benefit) pertaining to continuing operations.
72	intexp	Interest Expense	[Income Statement] Amount of the cost of borrowed funds accounted for as interest expense.
73	opex	Operating Expenses	[Income Statement] Operating expenses represents the total expenditure on [SGnA], [RnD] and other operating expense items, it excludes [CoR].
74	opinc	Operating Income	[Income Statement] Operating income is a measure of financial performance before the deduction of [INTEXP], [TAXEXP] and other Non-Operating items. It is calculated as [GP] minus [OPEX].
75	cor	Cost of Revenue	[Income Statement] The aggregate cost of goods produced and sold and services rendered during the reporting period.
76	consolinc	Consolidated Income	[Income Statement] The portion of profit or loss for the period, net of income taxes, which is attributable to the consolidated entity, before the deduction of [NetIncNCI].
77	shareswa	Weighted Average Shares	[Income Statement] The weighted average number of shares or units issued and outstanding that are used by the company to calculate [EPS], determined based on the timing of issuance of shares or units in the period.
78	shareswadil	Weighted Average Shares Diluted	[Income Statement] The weighted average number of shares or units issued and outstanding that are used by the company to calculate [EPSDil], determined based on the timing of issuance of shares or units in the period.

F Appendix Economics and Financial Data Definition

Index	Indicator Code	Indicator Name
1	ADP	Adp Employment Change
2	BCONF	Business Confidence
3	BOT	Balance Of Trade
4	BP	Building Permits
5	BR	Bankruptcies
6	CA	Current Account
7	CARS	Car Registrations
8	CBBS	Central Bank Balance Sheet
9	CCONF	Consumer Credit
10	CCPI	Core Consumer Prices
11	CF	Capital Flows
12	CFNAI	Chicago Fed National Activity Index
13	CHJC	Challenger Job Cuts
14	CJC	Continuing Jobless Claims
15	CNCN	Consumer Confidence
16	COR	Crude Oil Rigs
17	COSC	Crude Oil Stocks Change
18	CP	Corporate Profits
19	CPIC	Inflation Rate
20	CPMI	Chicago PMI
21	CPPI	Core Pce Price Index
22	CSP	Consumer Spending
23	CU	Capacity Utilization

24	DINV	Changes In Inventories
25	DPINC	Disposable Personal Income
26	DUR	Durable Goods Orders
27	EHS	Existing Home Sales
28	EMPST	Ny Empire State Manufacturing Index
29	EXPX	Export Prices
30	EXVOL	Exports
31	FACT	Factory Orders
32	FBI	Foreign Bond Investment
33	FDI	Foreign Direct Investment
34	FER	Foreign Exchange Reserves
35	FINF	Food Inflation
36	FOET	Factory Orders Ex Transportation
37	GAGR	GDP Annual Growth Rate
38	GASSC	Gasoline Stocks Change
39	GBVL	Government Budget Value
40	GCP	GDP Constant Prices
41	GD	GDP Deflator
42	GFCF	Gross Fixed Capital Formation
43	GGR	GDP Growth Rate
44	GPAY	Government Payrolls
45	GREV	Government Revenues
46	GSP	Government Spending
47	GYLD	Government Bond 10y
48	HSTT	Housing Starts
49	IMPX	Import Prices
50	IMVOL	Imports
51	ISMNYI	Ism New York Index
52	JCLM	Initial Jobless Claims
53	JOBOFF	Job Offers
54	JVAC	Job Vacancies
55	LC	Labour Costs
56	LPS	Loans To Private Sector
57	LUNR	Long Term Unemployment Rate
58	M0	Money Supply M0
59	M1	Money Supply M1
60	M2	Money Supply M2
61	MANWG	Wages In Manufacturing
62	MKT	Stock Market
63	MORTG	Mortgage Rate
64	MP	Manufacturing Production
65	MPAY	Manufacturing Payrolls
66	NAHB	Nahb Housing Market Index
67	NATGSC	Natural Gas Stocks Change
68	NFIB	Nfib Business Optimism Index
69	NHS	New Home Sales
70	NLTTF	Net Long Term Tic Flows
71	NMPMI	Non Manufacturing PMI
72	NO	New Orders
73	OIL	Crude Oil Production
74	OPT	Economic Optimism Index
75	PCEPI	Pce Price Index
76	PFED	Philadelphia Fed Manufacturing Index
77	PHS	Pending Home Sales
78	PPIC	Producer Prices Change
79	PROD	Productivity

80	PSAV	Personal Savings
81	RSM	Retail Sales Mom
82	RSY	Retail Sales Yoy
83	TOT	Terms Of Trade
84	TOUR	Tourist Arrivals
85	TVS	Total Vehicle Sales
86	UNR	Unemployment Rate
87	UNRY	Youth Unemployment Rate
88	WAGE	Wages

G Appendix Exchange Traded Funds ETFs Definitions

No.	Ticker	ETF Name
1	IVV	iShares Core S&P 500 ETF
2	IJK	iShares S&P Mid-Cap 400 Growth ETF
3	OEF	iShares S&P 100 ETF
4	IWO	iShares Russell 2000 Growth ETF
5	IUSG	iShares Core S&P U.S. Growth ETF
6	IWV	iShares Russell 3000 ETF
7	IWB	iShares Russell 1000 ETF
8	ITOT	iShares Core S&P Total U.S. Stock Market ETF
9	IJJ	iShares S&P Mid-Cap 400 Value ETF
10	IVW	iShares S&P 500 Growth ETF
11	IWN	iShares Russell 2000 Value ETF
12	IOO	iShares Global 100 ETF
13	IUSV	iShares Core S&P U.S. Value ETF
14	IYY	iShares Dow Jones U.S. ETF
15	IWR	iShares Russell Mid-Cap ETF
16	AGG	iShares Core U.S. Aggregate Bond ETF
17	IJH	iShares Core S&P Mid-Cap ETF
18	IVE	iShares S&P 500 Value ETF
19	IJT	iShares S&P Small-Cap 600 Growth ETF
20	IWP	iShares Russell Mid-Cap Growth ETF
21	IWM	iShares Russell 2000 ETF
22	IJR	iShares Core S&P Small-Cap ETF
23	DVY	iShares Select Dividend ETF
24	IJS	iShares S&P Small-Cap 600 Value ETF
25	IWS	iShares Russell Mid-Cap Value ETF
26	EPP	iShares MSCI Pacific ex Japan ETF
27	EEM	iShares MSCI Emerging Markets ETF
28	ILF	iShares Latin America 40 ETF
29	EWA	iShares MSCI Australia ETF
30	EWO	iShares MSCI Austria ETF
31	EWK	iShares MSCI Belgium ETF
32	EWC	iShares MSCI Canada ETF
33	EWQ	iShares MSCI France ETF
34	EWG	iShares MSCI Germany ETF
35	EWH	iShares MSCI Hong Kong ETF
36	EWI	iShares MSCI Italy ETF
37	EWJ	iShares MSCI Japan ETF
38	JPXN	iShares JPX-Nikkei 400 ETF
39	EWN	iShares MSCI Netherlands ETF
40	EWS	iShares MSCI Singapore ETF
41	EWP	iShares MSCI Spain ETF
42	EWD	iShares MSCI Sweden ETF
43	EWL	iShares MSCI Switzerland ETF

Continued on next page

Table 9 – *Continued from previous page*

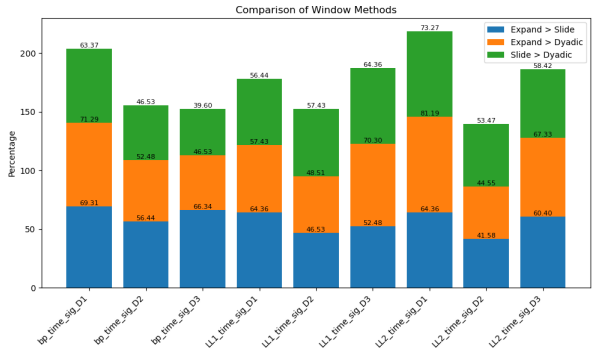
No.	Ticker	ETF Name
44	EWU	iShares MSCI United Kingdom ETF
45	FXI	iShares China Large-Cap ETF
46	EWZ	iShares MSCI Brazil ETF
47	EWM	iShares MSCI Malaysia ETF
48	EWW	iShares MSCI Mexico ETF
49	EZA	iShares MSCI South Africa ETF
50	EWY	iShares MSCI South Korea ETF
51	EWT	iShares MSCI Taiwan ETF
52	SHY	iShares 1-3 Year Treasury Bond ETF
53	IEF	iShares 7-10 Year Treasury Bond ETF
54	TLT	iShares 20+ Year Treasury Bond ETF
55	TIP	iShares TIPS Bond ETF
56	LQD	iShares iBoxx $ Investment Grade Corporate Bond ETF
57	IYK	iShares U.S. Consumer Staples ETF
58	IYC	iShares U.S. Consumer Discretionary ETF
59	IXC	iShares Global Energy ETF
60	IYE	iShares U.S. Energy ETF
61	IXG	iShares Global Financials ETF
62	IYG	iShares U.S. Financial Services ETF
63	IYF	iShares U.S. Financials ETF
64	IXJ	iShares Global Healthcare ETF
65	IBB	iShares Biotechnology ETF
66	IYH	iShares U.S. Healthcare ETF
67	IYT	iShares U.S. Transportation ETF
68	IYJ	iShares U.S. Industrials ETF
69	IYM	iShares U.S. Basic Materials ETF
70	IYR	iShares U.S. Real Estate ETF
71	ICF	iShares Cohen & Steers REIT ETF
72	IGM	iShares Expanded Tech Sector ETF
73	IGV	iShares Expanded Tech-Software Sector ETF
74	SOXX	iShares Semiconductor ETF
75	IYW	iShares U.S. Technology ETF
76	IXP	iShares Global Comm Services ETF
77	IYZ	iShares U.S. Telecommunications ETF
78	IDU	iShares U.S. Utilities ETF

H Appendix Compare Window Methods Across Augmentations and Degrees

The table and bar chart in Figure 10 are similar to Figure 5 where the augmentations are expanded across degrees. The values in blue indicate the dyadic outperforms expanding and sliding window methods at higher degrees. The values in red show deterioration in the expanding window method as the signature level increases.

The Table and bar chart in Figure 11 compare augmentation methods. Incorporating lead-lag augmentation improves the adjusted R^2 across all windows, with lead-lag at level 2 very close to lead-lag 1 (ave. **53**). The sliding window benefits the most from the lead-lag transformation. The three Charts in 11c d, e show sorted augmentation methods per window, confirm lead-lag 1 and 2 yield higher adjusted R^2 than basepoint transformation across all windows.

The Table and bar chart in Figure 12 compare signature levels. The results suggest that adjusted R^2 values increase as signature levels rise from 1 to 2 and 3, with degree 2 outperforming degree 3 for both sliding and dyadic windows. In other words, while both sliding and dyadic windows perform better at higher degrees, level 2 signatures are optimal (ave. **48**, and **53**). On the other hand, the first-level signatures' adjusted R^2 decrease as signature levels rise from 1 to 2 and 3, with degree 3 outperforming degree 2 under the expanding method. The three Charts in 12c d, e show sorted signature levels per window, confirm signature levels 2 and 3 yield higher adjusted R^2 than degree 1 for sliding and dyadic windows.

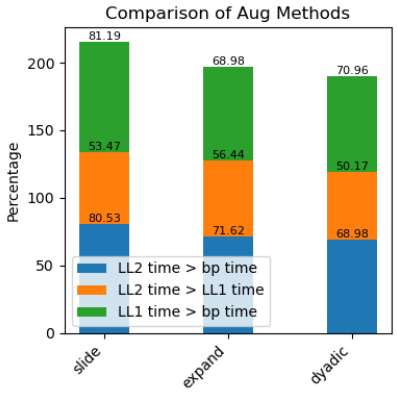


(a) across augmentations and degrees

Aug \ Window	E > S	E > D	S > D
BP time sig D1	69	71	63
BP time sig D2	56	52	47
BP time sig D3	66	47	40
LL1 time sig D1	64	57	56
LL1 time sig D2	47	49	57
LL1 time sig D3	52	70	64
LL2 time sig D1	64	81	73
LL2 time sig D2	42	45	53
LL2 time sig D3	60	67	58
ave.	**58**	**60**	**57**

(b) Expand (E), Slide (S), Dyadic (D)

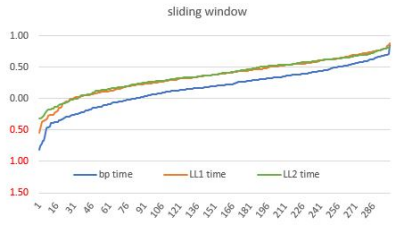
Figure 10: Window methods across augmentations and degrees



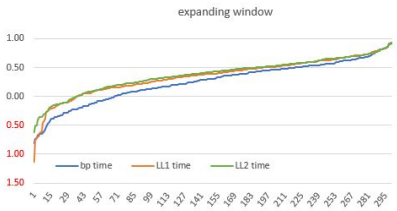
(a) across windows

Window \ Aug	LL2t > bpt	LL2t > LL1t	LL1t > bpt
Slide	81	53	81
Expand	72	56	69
Dyadic	69	50	71
ave.	**74**	**53**	**74**

(b) LeadLag [2, 1] time (LL2t, LL1t), basepoint time (bpt)



(c) Slide



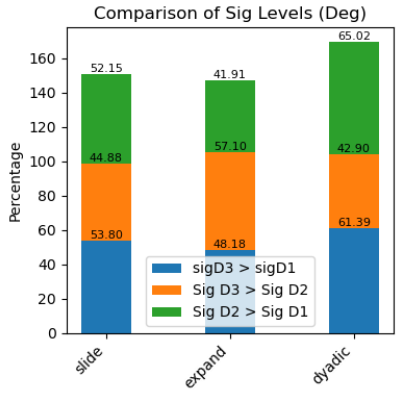
(d) Expand

(e) Dyadic

Figure 11: Augmentation methods across windows

I Appendix Datasets Analysis: Heatmap and One Hot Encoding

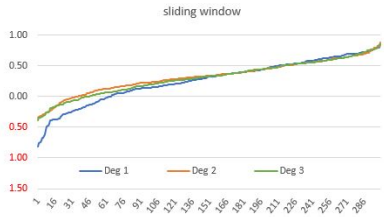
Figure 13a depicts the top and bottom fifty observations in terms of their adjusted R-squared values. The darker the color and the wider the stripe the larger the values. The blue and red colors indicate positive and negative values respectively. The fundamental data, i.e., the category labeled FDM, dominantly exhibits dark-blue and light-blue colors compared to the other two categories. The ETF data appears to be predominantly blue but on the lighter side. The macroeconomic data has more red colors than the other two categories. Figure 13b depicts the top 100 observations in terms of their adjusted R-squared values. The blue and red colors indicate positive values, with blue representing more positive than red. The Fundamental and ETF data appear to have higher adjusted R-squared than the macroeconomic data. Figure 13c is the same as 13b but adds stocks to the picture. We observe that stocks have a diversified selection of factors, that is, depending on the augmentation method and signature level, the same stocks have different sets



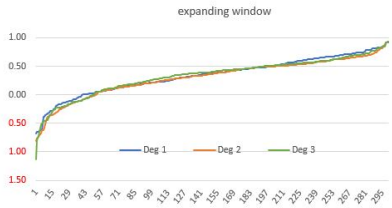
(a) Across windows

Window \ Sig Level	D3 > D1	D3 > D2	D2 > D1
Slide	54	45	52
Expand	48	57	42
Dyadic	61	43	65
ave.	**54**	**48**	**53**

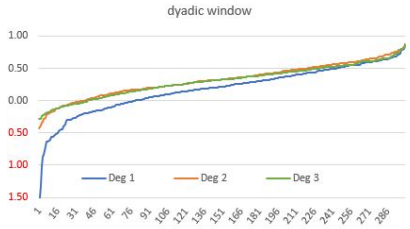
(b) Signature levels [D1, D2,D3]



(c) Slide



(d) Expand



(e) Dyadic

Figure 12: Signature levels across windows

of selected factors. Also observe, that the macroeconomic indicators stock market (USAMKT) and new home sales (USANHS) have been selected by most stocks.

Figure 14a visualizes the top and bottom fifty observations based on their adjusted R-squared values. Color intensity and stripe width represent the magnitude of the values, with darker colors and wider stripes indicating larger values. Blue signifies positive values, while red denotes negative values. Notably, ETF data predominantly exhibits shades of blue, suggesting strong positive relationships. Fundamental data displays a balanced mix of blue and red stripes, while macroeconomic data shows the presence of both dark blue and dark red stripes. Figure 14b focuses on the top 100 observations, using red for positive and blue for more positive values. ETF and fundamental data appear to have generally higher adjusted R-squared values, indicated by a lighter shade of red. Figure 14c extends this analysis by including stock data. Interestingly, stocks exhibit a diverse range of selected factors, with the choice varying significantly across augmentation methods and signature levels. The larger clusters observed around the macroeconomic data suggest that stocks oftentimes select different factors from macroeconomic data compared to ETF and fundamental data.

Table 10 presents the percentage count of those factors appearing three times or more by category within the top, center, and bottom-ranked adjusted R-squared values. The macroeconomic (ECOFIN) data is selected the most, it shows decreasing occurrences as we move from top to bottom (52% to 47%), while ETF and fundamental (FDM) data exhibit increasing occurrences (33% to 34%) and (15% to 19%) respectively. Comparing the ETF data to the FDM data, the former accounted for a higher proportion of selected factors than the latter. Table 11 displays a similar pattern.

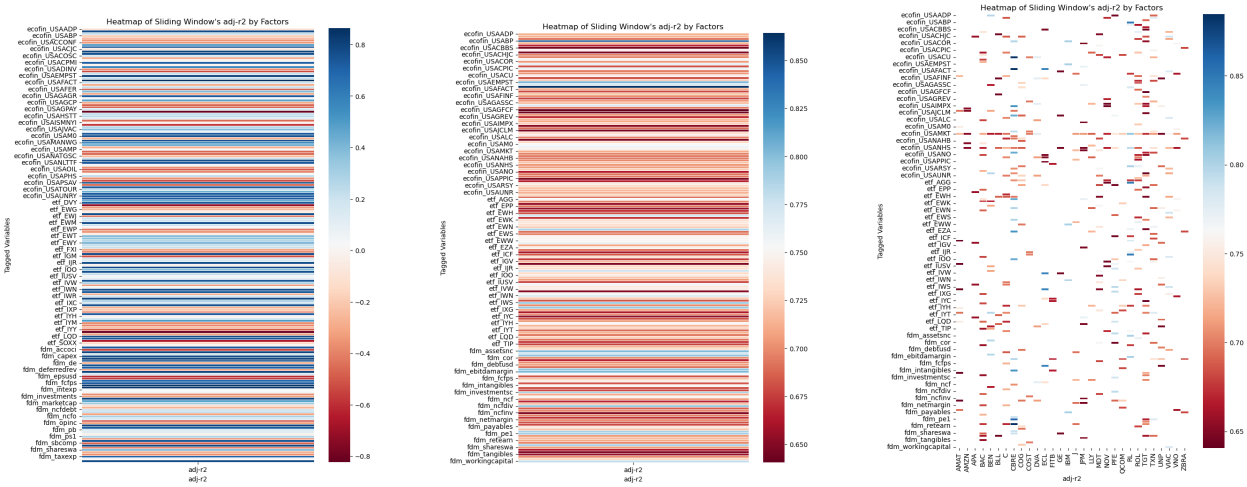
Tables 12 and 13 expand on the information presented in Table 10. Tables 14 and 15 expand on the information presented in Table 11. Instead of counts, they display symbols of factors appearing four or more times in the top-ranked and bottom-ranked adjusted R-squared values, for both sliding and dyadic window methods. Symbols in blue indicate factors common to both top and bottom rankings.

Category \ count occurrence >2	Top 100	Center 100	Bottom 100	Bottom 140
ECOFIN	52	43	47	47
ETF	33	36	34	34
FDM	15	20	19	19

Table 10: % count by category across top, center, bottom adj-R^2, sliding window

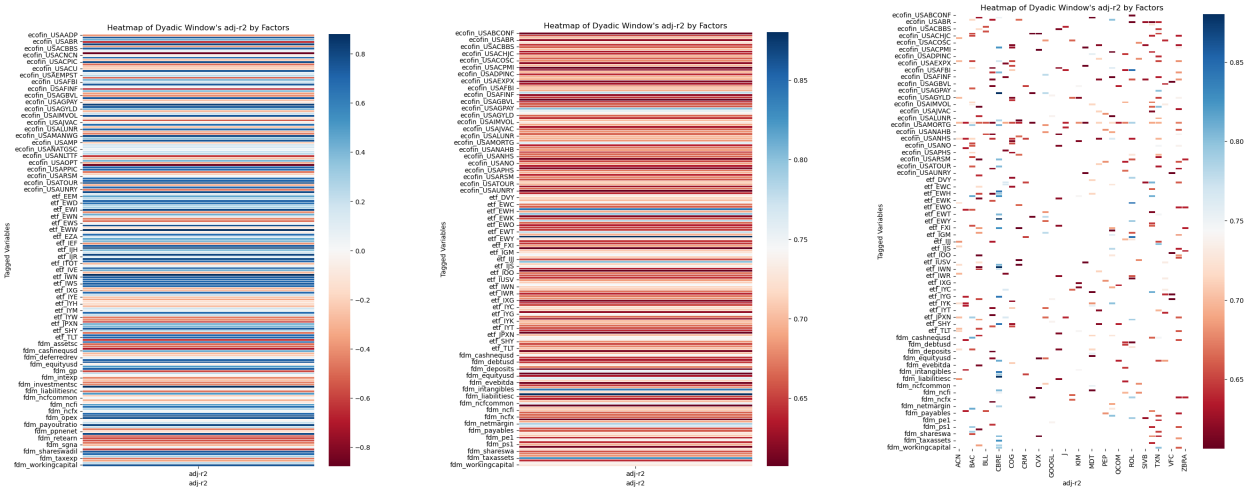
Category \ count occurrence >2	Top 100	Center 100	Bottom 100	Bottom 140
ECOFIN	48	39	42	43
ETF	32	33	34	34
FDM	20	28	24	23

Table 11: % count by category across top, center, bottom adj-R^2, dyadic window



(a) Top/bottom (50) total of (100) (b) Top (100) (c) Top 100 w. stocks

Figure 13: Sliding window Heatmap by factors and stocks



(a) Top/bottom (50) total of (100) (b) Top (100) (c) Top 100 w. stocks

Figure 14: Dyadic window Heatmap by factors and stocks

Ecofin	Ecofin (cont.)	ETF	FDM
USACHJC	USAMKT	DVY	ev
USACPMI	USAMP	EWH	liabilitiesc
USACU	USANHS	EWI	marketcap
USAFDI	USANO	EWM	ncfo
USAFINF	USAPFED	IOO	pe1
USAHSTT	USAPSAV	IWV	ps1
USAIMPX	USATVS	IXG	shareswa
USAISMNYI	USAUNR	IYF	shareswadil
USAJOBOFF	USAWAGE	IYK	
USAMANWG		LQD	

Table 12: Top 100, count occurrence >3, slide wd.

Ecofin	Ecofin (cont.)	ETF	FDM
USAADP	USAMKT	AGG	cashnequsd
USABP	USAMORTG	EWH	equityusd
USABR	USAMP	EWJ	pe1
USACA	USANHS	EWL	ppnenet
USAFDI	USANO	EWP	sgna
USAGD	USAOIL	EWQ	tbvps
USAHSTT	USAOPT	EWS	
USAIMPX	USAPFED	EWZ	
USAISMNYI	USARSM	IBB	
USAJCLM		IEF	
		IGM	
		IJT	
		IYC	
		IYH	
		IYW	
		SOXX	

Table 13: Bottom 100, count occurrence >3, slide wd.

Ecofin	Ecofin (cont.)	ETF	FDM
USABOT	USAGFCF	EEM	depamor
USABP	USAGYLD	EWH	ev
USABR	USAHSTT	EWI	marketcap
USACF	USAIMVOL	FXI	ncfo
USACP	USAMKT	IWN	shareswadil
USACPIC	USANATGSC	IYK	workingcapital
USAEMPST	USANHS	JPXN	
USAFACT	USANO	SHY	
USAFBI	USARSM		
USAFDI			
USAFINF			
USAFOET			

Table 14: Top 100, count occurrence. >3, dyadic wd.

Ecofin	Ecofin (cont.)	ETF	FDM
USABR	USAISMNYI	EWJ	accoci
USACARS	USAMKT	EWY	assetsnc
USACNCN	USAMORTG	EZA	equityusd
USACOSC	USANAHB	IEF	ev
USACPMI	USANHS	IYC	evebitda
USADUR	USANLTTF	IYG	gp
USAEMPST	USAOPT	IYH	invcap
USAFINF	USAPFED	IYK	investments
USAGASSC	USARSM	IYT	liabilitiesnc
USAGPAY	USATVS	IYZ	ncfcommon
USAHSTT		SHY	ncff
			sharesbas
			taxliabilities

Table 15: Bottom 100, count occurr. >3, dyadic wd.