MSC IN MATHEMATICAL SCIENCES

Investigating Stationarity Properties of the Signature Transform

Wojciech Anyszka

Trinity 2025

Abstract

The signature transform, a mathematical tool rooted in rough path theory, has gained prominence in machine learning due to its ability to extract informative features from time series data. This dissertation investigates the predictive utility of sliding signatures derived from stationary processes. We establish a novel theoretical result: the sliding signature transform preserves stationarity. Empirically, we confirm that log increments of forex prices are stationary, consistent with the Black-Scholes model, and demonstrate that sliding signatures of these increments retain stationarity in 51.3% of multidimensional cases and 93.0% of individual components. Predictive experiments indicate that models employing sliding signatures of log increments provide marginal improvements over those trained on raw prices. Although these signature-based models achieve low MSE and near-perfect R^2, their practical profitability is limited due to minimal returns and high variability. These findings suggest that while signature transforms offer theoretical and statistical advantages, translating these into meaningful practical outcomes remains challenging, highlighting the need for further research into their real-world applicability.

Word count: 5934

Contents

1 Introduction

The signature transform of a path is a mathematical tool that has been extensively studied in pure mathematics [Chen, 1954, 1957, 1961]. Initially, its application was limited to paths satisfying certain regularity conditions. However, the development of rough path theory significantly extended its scope, providing a rigorous framework for defining and analyzing controlled differential equations driven by geometric p-rough paths [Lyons et al., 2007, Lyons, 1998, Lyons and Qian, 2002]. While defining signatures for general rough paths requires sophisticated mathematics, the case of continuous, piecewise smooth paths is significantly more straightforward and accessible. This dissertation adopts this simplified context, focusing specifically on leveraging signature transform as feature extraction method in machine learning.

In recent years, the signature transform has attracted increasing attention in machine learning due to its versatility and strong theoretical foundation. It has been employed as a feature extraction method for time series [Lyons et al., 2014, Morrill et al., 2020a], in signature kernel methods [Kiraly and Oberhauser, 2019], which can be used for distribution regression tasks [Lemercier et al., 2021], and in neural controlled (rough) differential equations [Morrill et al., 2021]. Additionally, the signature transform has been integrated

as a layer within neural networks, enabling end-to-end learning architectures that leverage its expressive power [Kidger et al., 2019].

Beyond theoretical advancements, the signature transform has been successfully applied across multiple domains. In finance, it has been applied to market anomaly detection [Gyurkó et al., 2014], predicting the returns of unknown investment strategies based on past performance [Lyons et al., 2014], and pricing exotic options [Lyons et al., 2019]. In medicine, applications include early detection of sepsis onset [Morrill et al., 2020b], distinguishing between bipolar disorder and borderline personality disorder [Perez Arribas et al., 2018], and diagnosing Alzheimer's disease [Moore et al., 2019]. The signature transform has been employed in other notable applications such as malware detection [Cochrane et al., 2021], Chinese and Arabic handwriting recognition [Graham, 2013, Wilson-Nunn et al., 2018], sound compression [Lyons and Sidorova, 2005], and human action recognition [Yang et al., 2022]. These diverse applications highlight the effectiveness of signature transform in extracting essential information from data.

In this dissertation, we employ the signature transform as a feature extraction method for time series analysis. A key theoretical advantage of the signature transform is its universal approximation property —– any continuous (potentially nonlinear) function of a time series can be approximated arbitrarily well by a linear function of its signature. To exploit this property, we introduce the concept of sliding signatures, constructing a sequence of signatures from segments of a stationary time series. We establish a novel result, proving that the sliding signature of a stationary process remains stationary — a finding that, to the best of our knowledge, has not been previously reported.

Our theoretical analysis motivates the hypothesis that sliding signatures derived from stationary sequences might offer more robust, superior predictive performance compared to those from non-stationary sequences. To test this hypothesis, we focus on trend prediction of stock prices, where stationarity plays a crucial role. Under the Black–Scholes model, the log increments of asset prices form a stationary process, contrasting with the non-stationarity typically observed in raw price data. Thus, we empirically compare predictive performances of models trained on log increments and their sliding signatures (both stationary) versus raw prices and their sliding signatures (both non-stationary).

In the first set of experiments, we demonstrated that log increments of forex prices are stationary, as expected, while raw prices are not. We then tested the stationarity-preserving properties of the sliding signature transform. Applying the transform to stationary log increment sequences, we found that 51.3% of the resulting multidimensional signature sequences remained stationary, with 93.0% of their individual components retaining stationarity. This empirically validates our theoretical result that the sliding signature transform preserves stationarity, albeit with some degradation in the multidimensional case.

Predictive experiments show that models utilizing sliding signatures of log increments consistently achieve modest statistical improvements over models trained on raw prices. Despite these improvements, incorporating this model in a simple trading algorithm results in marginal profitability due to low average returns and high variability in trading outcomes. Moreover, there is no clear relation between profitability and low MSE. Thus, translating high statistical performance into practical financial gains presents significant challenges.

This dissertation is structured as follows. We begin by introducing the signature transform of a continuous piecewise differentiable path, exploring its key theoretical properties and geometric interpretation. Next, we discuss how the signature transform can be integrated into machine learning pipelines as a feature extraction method for time-series. Then, we outline augmentation and windowing techniques that enhance interpretability and

effectiveness of signature method. We then review stationarity and the Birkhoff-Khinchin ergodic theorem, and discuss their relevance to time series prediction. We then go on to defining sliding signature sequence and prove that it preserves stationarity. Subsequently, we detail our experimental methodologies, present empirical results, and discuss their implications, highlighting both theoretical strengths and practical limitations. Finally, we conclude by outlining potential directions for future research.

2 Preliminaries

In this section, we define the signature transform of a continuous piecewise differentiable path. We start from basic algebraic concepts essential for its understanding. The presentation closely follows the recent treatments by Cass and Salvi [2024] and [Chevyrev and Kormilitzin, 2025].

2.1 Algebraic Preliminaries

The tensor product is a fundamental construction in linear algebra that combines vector spaces in a way that captures multilinear relationships. Let V and W be finite-dimensional vector spaces over $\mathbb{R}$. The tensor product $V \otimes W$ is a vector space whose elements are formal linear combinations of symbols $v \otimes w$, where $v \in V$ and $w \in W$, subject to bilinearity relations

$$\begin{aligned}(v_1 + v_2) \otimes w &= (v_1 \otimes w) + (v_2 \otimes w),\\ v \otimes (w_1 + w_2) &= (v \otimes w_1) + (v \otimes w_2),\\ (sv) \otimes w &= s(v \otimes w),\\ v \otimes (sw) &= s(v \otimes w),\end{aligned}$$

for all $v, v_1, v_2 \in V$, $w, w_1, w_2 \in W$, and $s \in \mathbb{R}$. These relations ensure that the tensor product is linear in each argument, making it a bilinear operation. The tensor product naturally extends to higher-order tensors. For a vector space V, the k-fold tensor product $V^{\otimes k}$ is defined as:

$$V^{\otimes k} := \underbrace{V \otimes \cdots \otimes V}_{k \text{ times}}.$$

Elements of $V^{\otimes k}$ are finite linear combinations of terms of the form $v_1 \otimes v_2 \otimes \cdots \otimes v_k$, where $v_i \in V$ for $i = 1, \ldots, k$. These terms are subject to multilinearity, i.e. they are linear in each of the k slots. If V has basis $\{e_1, \ldots, e_m\}$ and W has basis $\{f_1, \ldots, f_k\}$, then the tensor product $V \otimes W$ has a basis given by:

$$\{e_i \otimes f_j \mid 1 \le i \le m, 1 \le j \le k\}.$$

Thus, $\dim(V \otimes W) = \dim(V) \cdot \dim(W)$. Similarly, for $V^{\otimes k}$, a basis is formed by all possible k-fold tensor products of the basis vectors of V:

$$\{e_{i_1} \otimes e_{i_2} \otimes \cdots \otimes e_{i_k} \mid 1 \le i_1, i_2, \ldots, i_k \le m\}.$$

When working with tensors, it is often convenient to express them in terms of their components relative to a basis. For a tensor $v \in V^{\otimes k}$, we denote by $v^{i_1,\ldots,i_k}$ the coefficient of v in front of the basis element $e_{i_1} \otimes \cdots \otimes e_{i_k}$. That is,

$$v = \sum_{i_1,\ldots,i_k} v^{i_1,\ldots,i_k}\, e_{i_1} \otimes \cdots \otimes e_{i_k}.$$

The **extended tensor algebra** $T((V))$ of a (finite-dimensional) vector space V is defined as the product space of all finite-order tensor products:

$$T((V)) := \prod_{i=0}^{\infty} V^{\otimes i}.$$

This space is equipped with scalar multiplication and addition defined component-wise, as well as, extended tensor product: For $v = (v_0, v_1, \dots)$ and $w = (w_0, w_1, \dots)$, their tensor product $v \otimes w$ is defined as:

$$v \otimes w = (z_0, z_1, \dots),$$

where each component z_k is given by:

$$z_k = \sum_{i=0}^{k} v_i \otimes w_{k-i}.$$

Here, $\otimes$ denotes the tensor product at each level. Note that this operation is associative, but not commutative. Finally, for practical applications, we often work with a truncated version of the extended tensor algebra. The **truncated tensor algebra** of order N, denoted by:

$$\prod_{i=0}^{N} V^{\otimes i},$$

consists of tensors of rank up to N. The product in this space is defined similarly:

$$v \otimes w = (z_1, \dots, z_N),$$

where:

$$z_k = \sum_{i=0}^{k} v_i \otimes w_{k-i}.$$

2.2 The Signature Transform

In this dissertation, we focus on the simplified setting of piecewise differentiable paths. However, the signature transform can be defined for much more irregular paths, as discussed in Cass and Salvi [2024]. Let $X : [0, T] \to \mathbb{R}^d$ be a continuous piecewise differentiable path[1] with components $X = (X^1, \dots, X^d)$. For a function $f : [a, b] \to \mathbb{R}$ with $[a, b] \subset [0, T]$, we can integrate f along X to obtain a vector in $\mathbb{R}^d$:

$$\int_a^b f(X_t)\, dX_t = \int_a^b f(X_t)\, \dot{X}_t\, dt,$$

where $\dot{X}_t$ denotes the derivative of X at time t. The signature transform is constructed using iterated integrals of the path X against itself. For example, the second-order iterated integral yields a matrix whose components are given by:

$$S^{i,j}(X) = \int_{a<t_1<t_2<b} dX^i_{t_1}\, dX^j_{t_2} = \int_{a<t_1<t_2<b} \dot{X}^i_{t_1}\, \dot{X}^j_{t_2}\, dt_1\, dt_2.$$

[1] In what follows, we denote by capital letters both paths and random variables, while small-case letters refer to points in $\mathbb{R}^d$.

This matrix is denoted by:

$$S(X)^{(2)}_{a,b} = \int_{a<t_1<t_2<b} dX_{t_1} \otimes dX_{t_2},$$

and is an element of the tensor product space $\mathbb{R}^d \otimes \mathbb{R}^d$. More generally, the k-fold iterated integral produces a k-tensor:

$$S(X)^{(k)} = \int_{a<t_1<t_2<\cdots<t_k<b} dX_{t_1} \otimes dX_{t_2} \otimes \cdots \otimes dX_{t_k},$$

whose components are:

$$S^{i_1,\ldots,i_k}_{a,b}(X) = \int_{a<t_1<\cdots<t_k<b} dX^{i_1}_{t_1} \ldots dX^{i_k}_{t_k} = \int_{a<t_1<\cdots<t_k<b} \dot{X}^{i_1}_{t_1} \dot{X}^{i_2}_{t_2} \ldots \dot{X}^{i_k}_{t_k}\, dt_1 dt_2 ... dt_k.$$

Definition 1. The *signature* of a continuous, piecewise differentiable path $X : [0, T] \to \mathbb{R}^d$ over an interval $[a, b] \subset [0, T]$ is defined as the sequence of tensors:

$$S(X)_{a,b} = \left(1, \int_{a<t_1<b} dX_{t_1}, \ldots, \int_{a<t_1<\cdots<t_k<b} dX_{t_1} \otimes \cdots \otimes dX_{t_k}, \ldots\right).$$

This sequence belongs to the extended tensor algebra $\prod_{k\geq 0}(\mathbb{R}^d)^{\otimes k}$. The signature transform to depth N is the truncated version of the signature:

$$\mathrm{Sig}^N(X)_{a,b} = \left(1, \int_{a<t_1<b} dX_{t_1}, \ldots, \int_{a<t_1<\cdots<t_N<b} dX_{t_1} \otimes \cdots \otimes dX_{t_N}\right),$$

which is an element of $\prod_{k=0}^{N}(\mathbb{R}^d)^{\otimes k}$.

Example 1. *Consider a one-dimensional path $X : [a, b] \to \mathbb{R}$. We claim that the n-th signature term is:*

$$S(X)^{(n)}_{a,b} = \frac{(X_b - X_a)^n}{n!}.$$

We prove this by induction. For $n = 1$, we have:

$$S(X)^{(1)}_{a,b} = \int_a^b dX_t = \int_a^b \dot{X}_t\, dt = X_b - X_a.$$

Assuming the result holds for $n - 1$, we compute the n-th term as:

$$S(X)^{(n)}_{a,b} = \int_a^b S(X)^{(n-1)}_{a,t} dX_t = \int_a^b \frac{(X_t - X_a)^{n-1}}{(n-1)!} \dot{X}_t\, dt.$$

Using the chain rule, this simplifies to:

$$S(X)^{(n)}_{a,b} = \int_a^b \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{(X_t - X_a)^n}{n!}\right) dt = \frac{(X_b - X_a)^n}{n!}.$$

Thus, the signature of a one-dimensional path depends only on its increment $X_b - X_a$.

Remark 2. *If $X = (Y, Z)$ is a two-dimensional path, the signature of X contains the signatures of its components Y and Z. Specifically, the signature of X can be decomposed into terms involving Y, Z, and their interactions. This property makes the signature transform particularly useful for analyzing multidimensional paths.*

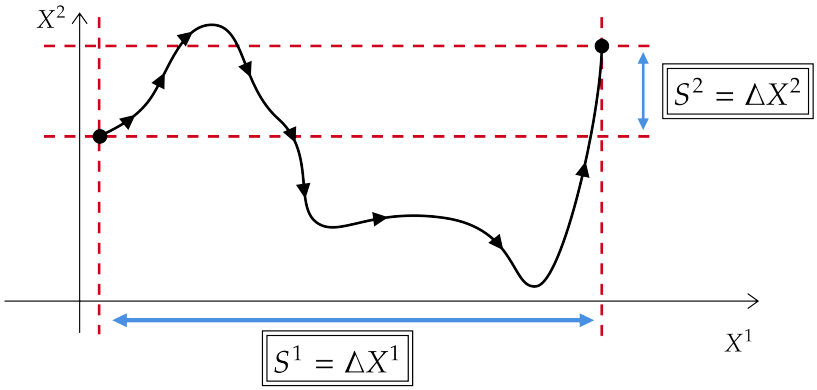

Figure 1: Illustration of the first-order terms of $S(X)$ encoding coordinate-wise increments.

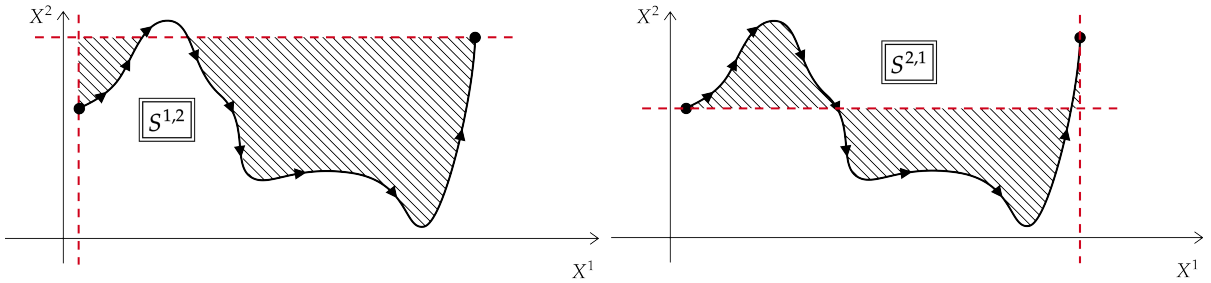

Figure 2: Illustration of the off-diagonal terms of $S(X)$ encoding area above and below the curve.

2.3 Geometric Interpretation

The first few terms of the signature have an intuitive geometric interpretation. First-order terms $S(X)^1_{a,b}, \ldots, S(X)^d_{a,b}$ represent the increment of the path X over the interval $[a, b]$, capturing the "net displacement" of the path in each dimension, see Figure 1. At the second level, the diagonal terms $S(X)^{i,i}$ are always given by $\frac{(X^i(b)-X^i(a))^2}{2}$, so they do not provide new information. The off-diagonal terms $S(X)^{i,j}$ encode the area above and below the curve, see Figure 2, while their difference the signed area enclosed between the path and the straight-line segment connecting its endpoints, known as the Lévy area, and displayed in Figure 3. For a two-dimensional path $\{X^1, X^2\}$, the Lévy area is given by:

$$A_{\text{Lévy}} = \frac{1}{2}\left(S(X)^{1,2}_{a,b} - S(X)^{2,1}_{a,b}\right).$$

This quantity captures the interaction between the components of the path. Intuitively, it measures how much the path "swings" around the straight-line segment connecting its endpoints. A positive Lévy area indicates that the path curves in one direction, while a negative value indicates curvature in the opposite direction. If $A = 0$, the path oscillates symmetrically around the chord, with the area above the chord equal to the area below it. This can be interpreted as the components X^1 and X^2 being linearly correlated with an added oscillatory component.

2.4 Properties of the Signature Transform

The signature transform has several properties that make it a powerful tool for analyzing paths. These include factorial decay, invariance under reparameterizations, translation

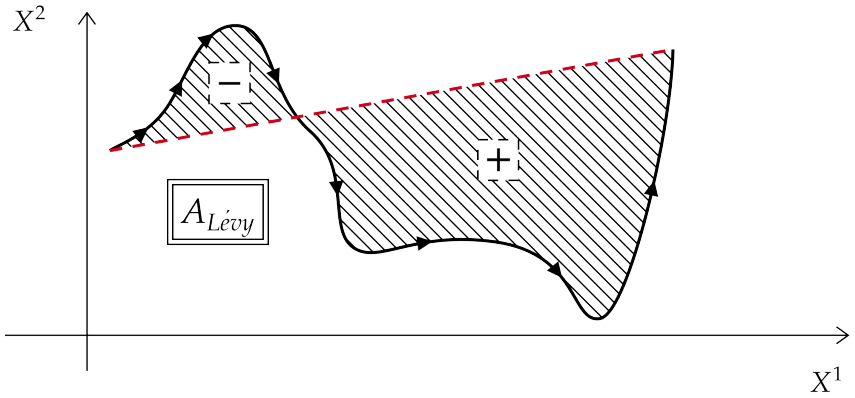

Figure 3: Illustration of the Lévy area of a curve.

invariance, Chen's identity, time-reversal property, uniqueness, and universal approximation. Below, we discuss these properties in detail.

Factorial Decay The terms of the signature transform exhibit factorial decay, meaning that the magnitude of the k-th level signature term $S(X)^k_{a,b}$ is bounded by:

$$\left\|S(X)^k_{a,b}\right\|_{(\mathbb{R}^d)^{\otimes k}} \leq \frac{L^k}{k!},$$

where L is the length of the path X [see Proposition 1.2.3 in Cass and Salvi [2024]]. This rapid decay ensures that truncating the signature transform at a sufficiently high level retains most of the critical information while enabling practical computations.

Invariance Under Reparameterizations The signature transform is invariant under reparameterizations of the path [Lemma 1.2.1 in Cass and Salvi [2024]]. This means that if $\psi : [a, b] \to [a, b]$ is a continuous, nondecreasing surjection (e.g., a time-rescaling function), and $Y_t = X_{\psi(t)}$, then:

$$S(X)_{a,b} = S(Y)_{a,b}.$$

This property removes the infinite-dimensional group of symmetries arising from time reparameterizations, effectively filtering out irrelevant variations in sampling frequency.

Shift Invariance The signature transform is invariant under shifts in the path. Specifically, for any fixed $x \in \mathbb{R}^d$, the signature of the shifted path $X + x$ satisfies:

$$S(X)_{a,b} = S(X + x)_{a,b}.$$

Chen's Identity and Time-Reversal If $X : [a, b] \to \mathbb{R}^d$ and $Y : [b, c] \to \mathbb{R}^d$ are paths, their concatenation $X * Y$ satisfies:

$$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c},$$

Moreover, for the time-reversed path $\overleftarrow{X}_t = X_{b+a-t}$, the following identity holds:

$$S(X)_{a,b} \otimes S(\overleftarrow{X})_{a,b} = 1.$$

For proofs see Lemma 1.3.1 and Lemma 1.3.7 in Cass and Salvi [2024]. Chen's identity is the main theoretical property that enables efficient computation of signatures on a computer, we will come back to this in the next section.

Uniqueness Clearly, reparametrization and shift invariance, and time-reversal property imply that the signature transform does not uniquely determine a path. However, we recover uniqueness under certain conditions. Specifically, if two paths $X, Y : [a, b] \to \mathbb{R}^d$ have the same signature and share a strictly monotone coordinate (e.g., time), then $X = Y$. Formally:

Theorem 3 (Uniqueness of Signatures, Proposition 1.2.4 from Cass and Salvi [2024]). *Let $X, Y : [a, b] \to \mathbb{R}^d$ be continuous, piecewise differentiable paths with $X^1(t) = Y^1(t)$ for all $t \in [a, b]$, where X^1 is strictly monotone. If $X^1(a) = Y^1(a)$, then $X = Y$ if and only if $S(X) = S(Y)$.*

Universal Approximation The signature transform is a universal nonlinearity for continuous functions of paths. Specifically, any continuous function of a path can be approximated arbitrarily well by a linear function of its signature:

Theorem 4 (Universal Approximation, Proposition 1.35 from Chevyrev and Kormilitzin [2025]). *Let $\mathcal{Y}_{a,y}$ denote the set of paths Y with $Y_a = (a, y)$ and $Y_t^1 = t$. For any compact set $K \subset \mathcal{Y}$, continuous function $g : K \to \mathbb{R}$, and $\epsilon > 0$, there exist $n \in \mathbb{N}$, coefficients $\lambda_1, \ldots, \lambda_n \in \mathbb{R}$, and multiindices $I_1, \ldots, I_n$ such that:*

$$\left| g(X) - \sum_{i=1}^{n} \lambda_i S(X)_{a,b}^{I_i} \right| < \epsilon \quad \text{for all } X \in K.$$

This result, which follows from the Stone-Weierstrass theorem, establishes the expressive power of the signature transform and motivates its use as a feature map for time-series.

3 Signature Transform in Practice

In this section, we describe how the signature transform can be applied to time series data, our approach is based on Morrill et al. [2020a]. Let $\mathcal{S}(\mathbb{R}^d)$ denote the space of time series over $\mathbb{R}^d$, defined as:

$$\mathcal{S}(\mathbb{R}^d) = \{(x_1, \ldots, x_n) \mid x_i \in \mathbb{R}^d, n \in \mathbb{N}, n \geq 1\}.$$

For a time series $x = (x_1, \ldots, x_n) \in \mathcal{S}(\mathbb{R}^d)$, we assume access to the associated vector of increasing timestamps $t = (t_1, \ldots, t_n)$.

To compute the signature of a time series, we first construct a continuous path from the discrete data. There are several ways to do this, including: piecewise affine interpolation (connects consecutive points with straight lines), rectilinear interpolation (moves along one coordinate at a time), cubic splines (uses smooth polynomial curves, as in Morrill et al. [2021]). A particularly convenient choice for our purposes is piecewise affine interpolation because the signature of a linear path is particularly simple. Namely, consider a segment

$$X(t) = \frac{x_0 b - x_1 a + t(x_1 - x_0)}{b - a}$$

for $t \in [a, b]$, which connects $x_0 \in \mathbb{R}^d$ to $x_1 \in \mathbb{R}^d$. By reparametrization invariance, we may instead compute the signature of $\tilde{X}(t) = x_0 + (x_1 - x_0)t$ for $t \in [0, 1]$. Thus the n-th signature term of X is given by:

$$S(X)_{a,b}^{(n)} = S(\tilde{X})_{0,1}^{(n)} = \int_{0<t_1<\cdots<t_n<1} d\tilde{X}_{t_1} \otimes \cdots \otimes d\tilde{X}_{t_n}.$$

Since $\dot{\hat{X}}_t = x_1 - x_0$ and the volume of the standard n-simplex is $\frac{1}{n!}$, this simplifies to:

$$S(X)_{a,b}^{(n)} = (x_1 - x_0)^{\otimes n} \int_{a<t_1<\cdots<t_n<b} dt_1 \cdots dt_n = \frac{(x_1 - x_0)^{\otimes n}}{n!}.$$

Thus, the signature of the segment X is:

$$S(X)_{a,b} = \exp(x_1 - x_0),$$

where exp denotes the exponential map in the tensor algebra. With this identity at hand we make the following definition.

Definition 5. Let $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{S}(\mathbb{R}^d)$ be a sequence of points with corresponding time stamps $(t_1, \ldots, t_n)$. The empirical signature[2] $\texttt{Sig}$ of $\mathbf{x}$ is defined as

$$\texttt{Sig}(\mathbf{x}) := S(X)_{t_1,t_n}$$

where $X : [t_1, t_n] \to \mathbb{R}^d$ is the piecewise linear interpolation of $\mathbf{x}$, meaning that $X(t_i) = x_i$ for all $i = 1, \ldots, n$ and $X(t)$ is linearly interpolated between these points. The truncated empirical signature $\texttt{Sig}^N$ of depth N is obtained by truncating $\texttt{Sig}(\mathbf{x})$ at level N.

By applying Chen's identity and our previous computations, the empirical signature of $\mathbf{x} = (x_1, ..., x_n) \in \mathcal{S}$ is

$$\texttt{Sig}(x) = S(X)_{t_1,t_2} \otimes \cdots \otimes S(X)_{t_{n-1},t_n} = \exp(x_2 - x_1) \otimes \exp(x_3 - x_2) \otimes \cdots \otimes \exp(x_n - x_{n-1}),$$

where X is the piecewise linear interpolation of $\mathbf{x}$. For the truncated empirical signature $\texttt{Sig}^N(X)$, the exponential maps are truncated at the N-th level. In our experiments, we use the Python package `signatory` Kidger and Lyons [2021] which leverages the above decomposition to compute empirical signatures efficiently.

Remark 6. *In machine learning applications the $k = 0$ term of the signature (which is always 1) is typically omitted, as it does not provide useful information for learning tasks.*

Remark 7. *The depth-N signature of a time series $\mathbf{x} \in \mathcal{S}(\mathbb{R}^d)$ of length n can be computed using the* `signatory` *library, which operates with a computational complexity of $\mathcal{O}(d^N)$ [Kidger and Lyons, 2021]. Importantly, the size of the depth-N signature is given by $\frac{d^{N+1}-1}{d-1}$, a quantity that is independent of the series length n. This makes the signature transform highly efficient, especially when dealing with high-frequency data. While the exponential growth of complexity with respect to N might initially seem problematic, empirical results show that even small values of N, such as $N = 3$, often yield competitive performance in practice. Thus, the signature remains a practical tool for extracting information from multi-dimensional time series.*

3.1 Path Augmentations

Path augmentations are transformations applied to a time series prior to computing its signature. Careful selection and combination of augmentations allow for more meaningful analysis of time series data. These transformations serve several key purposes:

[2] In what follows, we may refer to the empirical signature simply as the signature. The meaning should be clear from the context.

1. Eliminating Undesirable Invariances: Certain invariances, such as translation or reparameterization invariance, can obscure meaningful information in the signature. Augmentations mitigate these effects.

2. Managing Computational Complexity: The number of terms in a depth-N signature scales as $\mathcal{O}(d^N)$, where d is the dimension of the time series. Some augmentations help control this growth by restructuring the data; intuitively they transfer the information from higher-order signature terms to lower-order terms by increasing the path dimension, i.e. they decrease N at the cost of increasing d.

3. Enhancing Feature Extraction: By modifying the representation of the time series, augmentations can improve the interpretability and effectiveness of the extracted signature features.

Below, we discuss several commonly used augmentations and their respective advantages.

Cumulative Sum The cumulative sum transformation converts a sequence of increments into a cumulative process, making trends and patterns more apparent. Given a sequence $\{x_i\}_{i=1}^N$, the transformed sequence $\{S_i\}_{i=1}^N$ is defined as

$$S_i = \sum_{j=1}^{i} x_j.$$

This transformation is particularly useful when analyzing processes where cumulative effects, rather than individual changes, carry more meaningful information.

Base-Point Augmentation Base-point augmentation removes translation invariance by prepending a fixed reference point—typically the origin—to the sequence. Given a sequence $\{x_i\}_{i=1}^N$, the augmented sequence is

$$\{0, x_1, x_2, \ldots, x_N\}.$$

Without this augmentation, the signature of a time series remains unchanged under uniform shifts $x_i \mapsto x_i + c$ for any constant vector $c \in \mathbb{R}^d$. By introducing a fixed base point, the transformation ensures that the signature depends explicitly on the absolute positions of the data points.

Time Augmentation Time augmentation introduces an additional dimension representing time, transforming a sequence $\mathbf{x} = (x_1, ..., x_N)$ into

$$\{(t_i, x_i)\}_{i=1}^N,$$

where t_i are the time stamps associated with $\mathbf{x}$. This augmentation eliminates reparameterization invariance, ensuring that the signature is sensitive to the rate at which the path is traversed. Additionally, it guarantees that the signature uniquely characterizes the path when time is strictly increasing (see Theorem 3).

Lead-Lag Transformation The lead-lag transformation is designed to encode additional second-order information, such as quadratic variation. It increases the dimensionality of the path while preserving its structure. Given a sequence $\{x_i\}_{i=1}^{N}$, the transformation constructs a new sequence $\{(x_i^{\text{lead}}, x_i^{\text{lag}})\}_{i=1}^{2N-1} \in \mathcal{S}(\mathbb{R}^{2d})$ of twice the dimension and (nearly) twice the length, where:

$$x_i^{\text{lead}} = \begin{cases} x_n & \text{if } i = 2(n-1), \\ x_n & \text{if } i = 2(n-1)+1, \end{cases} \qquad x_i^{\text{lag}} = \begin{cases} x_n & \text{if } i = 2n, \\ x_n & \text{if } i = 2n-1. \end{cases}$$

For example, applying the lead-lag transformation to $\{x_1, x_2\}$ results in the sequence

$$\{(x_1, x_1), (x_2, x_1), (x_2, x_2)\}.$$

This transformation captures interactions between successive data points, making it particularly useful for applications that require second-order structure. The Lead-Lag has been found to perform very-well in practice, whatever the algorithm or dataset used [Fermanian, 2021].

Augmentations can be combined to achieve multiple objectives. For instance, applying time augmentation followed by base-point augmentation removes both translation and reparameterization invariance while ensuring that the signature uniquely determines the path. The choice of augmentation depends on the specific characteristics of the dataset and the objectives of the analysis.

3.2 Windowing

Once a time series has been augmented, the next step is to determine an appropriate segmentation strategy for signature computation. A well-designed windowing approach ensures that both global and local information is effectively captured. Later, we will apply the sliding windowing technique to generate a time series of signatures from the original data and demonstrate that this transformation preserves stationarity. Below, we outline several commonly used windowing methods and their advantages.

Global Window The simplest approach is to compute the signature over the entire time series, a method referred to as the global window. This approach captures the overall structure of the data but may fail to detect finer local variations. It is best suited for stationary processes or when global features are of primary interest.

Expanding Window The expanding window method starts with an initial segment of the series and progressively extends it by adding more data points. Given a time series $\mathbf{x} = (x_1, \ldots, x_N)$, an expanding window with an initial length ℓ and step size k generates the sequence

$$W(\mathbf{x}) = (\mathbf{x}_{1:\ell}, \mathbf{x}_{1:\ell+k}, \mathbf{x}_{1:\ell+2k}, \ldots, \mathbf{x}_{1:N}),$$

where $\mathbf{x}_{a:b} := (x_a, x_{a+1}, \ldots, x_b)$. This method is particularly useful for tracking how information accumulates over time, making it well-suited for forecasting and online learning settings.

Sliding Window The sliding window approach extracts signatures from fixed-length, overlapping segments of the time series. Given a window of length ℓ and step size k, the sliding windows are defined as

$$W(\mathbf{x}) = (\mathbf{x}_{1:\ell}, \mathbf{x}_{k+1:\ell+k}, \mathbf{x}_{2k+1:\ell+2k}, \ldots, \mathbf{x}_{mk+1:N}),$$

where m is chosen such that $mk + \ell \leq N$. Sliding windows are particularly effective for capturing local dependencies and short-term dynamics.

Hierarchical Dyadic Window The hierarchical dyadic window method is designed to capture information at multiple temporal scales. Given a fixed depth $q \in \mathbb{N}$, suppose for simplicity (but it can be overcome) that 2^{q-1} divides the length N of the time-series. This windowing technique constructs q levels of sliding windows, where the window at depth i has length and step size $\frac{N}{2^{i-1}}$. The hierarchical dyadic windowing method is particularly effective when analyzing data with features at multiple time scales, such as financial or physiological time series.

Fermanian [2021] empirically observed that the signature of the global window captures as much information as the signatures of other windows. In contrast, the empirical analysis of Morrill et al. [2020a] suggests that dyadic and expanding windows outperform both global and sliding windows.

Remark 8. *Higher-order signature terms decay factorially in magnitude, suggesting that rescaling each k-th term by $k!$ might improve numerical stability. However, empirical studies indicate that such rescaling does not consistently enhance performance and may even introduce unintended distortions [Morrill et al., 2020a].*

4 Stationarity and the Ergodic Theorem

The strong law of large numbers (SLLN) is a fundamental result in probability theory, forming the basis for the probabilistic interpretation of relative frequencies. In statistical analysis, estimating an unknown distribution typically relies on drawing multiple independent samples. As the number of observations increases, the empirical distribution converges to the true distribution. This principle naturally extends to stochastic processes, which model time-dependent random phenomena. However, in practical applications, we often face situations where only a single realization of a stochastic process is available. For example, when analyzing the historical price trajectory of a financial asset, only one sample path of the underlying stochastic process is observable. Unlike i.i.d. sampling, where multiple independent realizations are available, time series data often consists of a single sequence of dependent observations. Consequently, to conduct meaningful statistical inference, we must impose structural assumptions on the process. A key assumption that enables inference from a single trajectory is stationarity.

A random sequence $\{X_n\}_{n\in\mathbb{N}}$ is said to be stationary if its statistical properties remain invariant under time shifts. Formally, for any $a \in \mathbb{Z}_+$ and any finite sequence of integers $0 \leq i_1 < \cdots < i_n$, the joint distribution of $(X_{i_1}, \ldots, X_{i_n})$ is identical to that of $(X_{i_1+a}, \ldots, X_{i_n+a})$. This property ensures that key characteristics such as mean, variance, and autocovariance do not change over time. Moreover, if f is a measurable function, then the transformed process $\{f(X_t)\}$ is also stationary. There is an equivalent definition of stationarity of a random sequence X_n which states that the process is called stationary if

it can be expressed as

$$X_n = f \circ T^n,$$

where $T : \Omega \to \Omega$ is a measure-preserving transformation, and f is a measurable function (see Lemma 25.1 in Kallenberg [2021]).

4.1 The Birkhoff–Khinchin Theorem

A fundamental result in the study of stationary processes is the Birkhoff–Khinchin ergodic theorem [Theorem 25.6 in Kallenberg [2021]], which provides conditions under which time averages converge to ensemble averages. Specifically, if $\{X_n\}$ is a stationary process, then for any measurable function $g : \mathbb{R} \to \mathbb{R}$ we have

$$\frac{1}{n}\sum_{k=0}^{n-1} g(X_k) = \frac{1}{n}\sum_{k=0}^{n-1} g \circ f \circ T^k \xrightarrow{n\to\infty} \mathbb{E}[g(X_0) \mid \mathcal{I}] \quad \text{a.s.},$$

where $\mathcal{I}$ is the invariant σ-algebra of T. When $f \in L^p(\Omega)$ and $g \in L^p(\mathbb{R})$ the above convergence also holds in L^p. A process is called ergodic if $\mathcal{I}$ is a trivial σ-algebra, i.e. all of its elements have measure 0 or 1. In the ergodic case, we have that

$$\frac{1}{n}\sum_{k=0}^{n-1} g(X_k) \xrightarrow{n\to\infty} \mathbb{E}[g(X_0)],$$

thus the statistical properties such as mean and variance can be inferred from a sufficiently long realization of a single trajectory. This property is crucial in practical applications, as it justifies the use of historical data for statistical estimation. The SLLN can be viewed as a special case of the ergodic theorem, as an independent and identically distributed (i.i.d.) sequence is trivially stationary and is also ergodic.

4.2 Sliding Signatures as Stationarity Preserving Map

Theorem 9. *Let $\{X_n\}$ be a stationary sequence. For any $\ell, k \in \mathbb{Z}_+$, the sequences $Z_n = \mathtt{Sig}(X_{(n-1)k+1:(n-1)k+\ell})$ and $W_n = \mathtt{Sig}^N(X_{(n-1)k+1:(n-1)k+\ell})$ are stationary, where $X_{a:b} := (X_a, X_{a+1}, \ldots, X_b)$. We call Z_n and W_n the* **sliding signatures** *of X_n.*

Proof. Let $\{X_n\}_{n\in\mathbb{N}}$ be a stationary sequence. Fix $\ell, k \in \mathbb{Z}_+$, and define the sequence of sliding windows:

$$Y_n := X_{(n-1)k+1:(n-1)k+\ell} = (X_{(n-1)k+1}, X_{(n-1)k+2}, \ldots, X_{(n-1)k+\ell}).$$

We first show that $\{Y_n\}$ is stationary. For any integer $a > 0$ and indices $0 < i_1 < i_2 < \cdots < i_N$, the joint distribution of the sliding windows satisfies

$$(Y_{i_1}, \ldots, Y_{i_N}) = (X_{(i_1-1)k+1}, \ldots, X_{(i_1-1)k+\ell}, \ldots, X_{(i_N-1)k+1}, \ldots, X_{(i_N-1)k+\ell}).$$

By the stationarity of $\{X_n\}$, this joint distribution is invariant under shifts so

$$(Y_{i_1}, \ldots, Y_{i_N}) \sim (X_{(i_1-1)k+1+ka}, \ldots, X_{(i_1-1)k+\ell+ka}, \ldots, X_{(i_N-1)k+1+ka}, \ldots, X_{(i_N-1)k+\ell+ka}).$$

Rewriting the indices, we obtain

$$\begin{aligned}(Y_{i_1}, \ldots, Y_{i_N}) &\sim (X_{(i_1+a-1)k+1}, \ldots, X_{(i_1+a-1)k+\ell}, \ldots, X_{(i_N+a-1)k+1}, \ldots, X_{(i_N+a-1)k+\ell}) \\ &= (Y_{i_1+a}, \ldots, Y_{i_N+a}).\end{aligned}$$

Thus, $\{Y_n\}$ is stationary. Next, define the sequence $Z_n := \texttt{Sig}(Y_n) = \texttt{Sig}(X_{(n-1)k+1:(n-1)k+\ell})$. By the definition of the empirical signature transform, we have

$$Z_n = \exp\big(X_{(n-1)k+2} - X_{(n-1)k+1}\big) \otimes \cdots \otimes \exp\big(X_{(n-1)k+\ell} - X_{(n-1)k+\ell-1}\big).$$

This can be written as $Z_n = f(Y_n)$, where $f : \mathbb{R}^\ell \to \Pi_{n\geq 0}\mathbb{R}^{\otimes n}$ is the continuous map

$$f(x) = \exp\big(x^2 - x^1\big) \otimes \cdots \otimes \exp\big(x^\ell - x^{\ell-1}\big).$$

The continuity (and hence measurability) of f follows from the fact that each component of f is a polynomial in the differences $x^{i+1} - x^i$, and polynomials are continuous. Since measurable maps preserve stationarity, $\{Z_n\}$ is stationary.

Finally, consider the truncated signature $W_n := \texttt{Sig}^N(Y_n) = \pi_N(Z_n)$, where $\pi_N : \Pi_{n\geq 0}\mathbb{R}^{\otimes n} \to \Pi_{n=0}^N \mathbb{R}^{\otimes n}$ is the projection map. Because π_N is continuous (and hence measurable), $\{W_n\}$ is also stationary. This completes the proof. □

Remark 10. *In contrast to sliding windows, expanding windows do not generally induce a stationary sequence of signatures. To see this, consider a one-dimensional stationary sequence $\{X_n\}$ and define the sequence $Y_n := \texttt{Sig}(X_{1:n})$. By Example 1, the signature of a one-dimensional path is given by:*

$$Y_n = \exp(X_n - X_1).$$

In general, Y_2 and Y_3 do not have the same distribution, implying that $\{Y_n\}$ is not stationary.

To provide a concrete example, let $\{X_n\}$ be a two-state Markov chain with states $\{0, 1\}$ and transition matrix:

$$P = \begin{pmatrix} 0.1 & 0.9 \\ 0.9 & 0.1 \end{pmatrix}.$$

If the initial distribution of X_1 is uniform, the Markov chain is stationary. Now, consider the first-order terms of Y_2 and Y_3

$$Y_2^1 = X_2 - X_1, \quad Y_3^1 = X_3 - X_1.$$

Next, we observe that

$$P(X_2 - X_1 = 1) = P(X_2 = 1, X_1 = 0) = P(X_2 = 1 \mid X_1 = 0)P(X_1 = 0) = 0.9 \cdot 0.5 = 0.45,$$

while

$$\begin{aligned} P(X_3 - X_1 = 1) = P(X_3 = 1, X_1 = 0) &= \sum_{i\in\{0,1\}} P(X_3 = 1 \mid X_2 = i)P(X_2 = i \mid X_1 = 0)P(X_1 = 0) \\ &= 0.9 \cdot 0.1 \cdot 0.5 + 0.1 \cdot 0.9 \cdot 0.5 = 0.09. \end{aligned}$$

Since $P(X_3 - X_1 = 1) \neq P(X_2 - X_1 = 1)$, it follows that $Y_2 \not\sim Y_3$. Therefore, the sequence of signatures induced by expanding windows is not stationary.

4.3 Stationarity and Path Augmentations

In this section, we analyze whether common path augmentations preserve the stationarity of a time series. Since these transformations are applied before computing the signature, it is crucial to understand their impact on stationarity. Let $\{X_n\}$ be a stationary sequence. We examine the stationarity-preserving properties of the following augmentations:

Cumulative sum The cumulative sum transformation does not, in general, preserve stationarity. This is because we *do not* even have that $S_1 = X_1 \sim X_1 + X_2 = S_2$. A simple counterexample is provided by the two-state Markov chain from Remark 10, where $S_1 \not\sim S_2$.

Base-Point Augmentation Base-point augmentation adds an initial basepoint (typically zero) to the sequence, resulting in $\{0, X_1, X_2, \dots\}$. Strictly speaking, this transformation does not preserve stationarity because $0 \not\sim X_n$. However, it morally preserves stationarity in the sense that the resulting sequence is stationary after the first term.

Time Augmentation Time augmentation adds a deterministic time component to the sequence, transforming $\{X_n\}$ into $\{(t_n, X_n)\}$, where t_n is the time at which X_n was sampled. This augmentation preserves stationarity because the added time component is deterministic and does not introduce any stochastic dependencies.

Lead-Lag Transformation The lead-lag transformation does not, in general, preserve stationarity. This is because, in general, we *do not* even have that $(X_1^{\text{lead}}, X_1^{\text{lag}}) = (X_1, X_1) \sim (X_2, X_1) = (X_2^{\text{lead}}, X_2^{\text{lag}})$. The two-state Markov chain from Remark 10 serves as a counterexample. However, the sliding signature of a lead-lag transformed stationary sequence retains some stationarity properties. What we mean by that is the following. Recall from Remark 2 that the signature of $(X_n^{\text{lead}}, X_n^{\text{lag}})$ contains within it the signature of X_n^{lag}. Thus the sliding signature of a lead-lag transformed signal contains within it the sliding signature of the lag component. In turn, the sliding signature of X_n^{lag} has stationary subsequences corresponding to sliding signatures of X_n but with different step-sizes and window lengths. For example, if the sliding signature with window-length 3 and step size 1 is applied to X_n^{lag} it outputs exactly the same sequence as the sliding signature with window-length 2 and step size 1 applied to X_n, and hence is stationary; this is a consequence of reparametrization invariance of signatures.

4.4 Statistical Tests for Stationarity

In our analysis, we assume that the time series under consideration is stationary, meaning its statistical properties remain constant over time. We also make the (unjustified) assumption that our time series is ergodic. While stationarity is often enough for many statistical methods, ergodicity ensures that empirical averages truly represent expected values. If ergodicity fails, then long-term behavior of observations may not generalize. The definition of ergodicity requires checking whether every invariant event has probability 0 or 1. In practice, this is difficult to verify. There are heuristic methods that suggest that a process is ergodic, however, we will ignore such approaches and the issue of ergodicity entirely and focus on testing stationarity (which is also subtle). To validate our stationarity assumption, we employ two widely used statistical tests:

Augmented Dickey-Fuller (ADF) test: This test evaluates the null hypothesis that the time series contains a unit root, implying non-stationarity. Rejecting the null hypothesis suggest stationarity. However, absence of a unit root does not imply stationarity, for example if it contains a deterministic trend. Failure to reject the null suggests the presence of a unit root and thus non-stationarity.

Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: This test assesses the null hypothesis that the series is trend-stationary, meaning that it is stationary after removing a deterministic trend. Rejecting the null hypothesis suggests that the series is not trend-stationary and hence not stationary.

If the ADF rejects the null and KPSS does not, there is strong evidence for trend stationarity which, in our experiments, we will take for evidence of stationarity. A subtle but important distinction exists between stationarity and trend stationarity. A trend-stationary process exhibits a deterministic trend but remains stationary after detrending. The tests we employ assess weaker forms of stationarity (i.e., trend stationarity or absence of a unit root), whereas our analysis assumes stationarity. This presents a potential gap, as a trend-stationary process may not satisfy stationarity conditions.

Additionally, these tests are applied separately to each component of a multidimensional time series, evaluating weak stationarity (i.e., stationarity in mean and variance) rather than full joint stationarity. Assessing the stationarity of an entire multidimensional process is a challenging problem, and in our experiments, we focus on testing individual components.

Despite this limitation, we hypothesize that learning from even from trend-stationary processes is significantly more tractable than from non-stationary ones. Therefore, while our statistical tests focus on trend-stationarity, they still provide meaningful insights into the underlying structure of the data.

5 Experiments

In our experiments, we utilized top-of-book, tick-by-tick market data with fractional pip spreads at millisecond resolution, freely available (upon account registration) from TrueFX [Tru]. Specifically, we analyzed data from three forex pairs—EUR/USD, EUR/GBP, and GBP/USD—collected in July 2024. Our focus was on predicting the *Buy* price of each pair. After aligning timestamps across all three pairs, the dataset comprised 264,803 data points. From the raw price data, we constructed two distinct datasets:

1. The raw prices p_i at each timestamp t_i.

2. The log price increments, defined as $\log(p_i) - \log(p_{i-1})$ at each timestamp t_i.

We refer to these as *price data* and *log data*, respectively. Both datasets represent three-dimensional time series, as they include data from the three forex pairs. For our experiments, we divided each dataset into smaller sequences, which we refer to as *price sequences* and *log sequences*.

5.1 Stationarity Testing

In the first experiment, we extracted 128 non-overlapping sequences of length 2048 from both price and log increment datasets and assessed their stationarity using the ADF and KPSS tests. A sequence was considered stationary if all its individual components had an ADF p-value < 0.05 and a KPSS p-value > 0.05.

As summarized in Table 1, none of the multidimensional price sequences were stationary, while 91.4% of the multidimensional log increment sequences met the stationarity criteria. This aligns with the expectation from the Black-Scholes model, which suggests that log returns are more likely to exhibit stationarity.

	Stationary	Nonstationary	Percentage Stationary
Prices	0 (3)	128 (381)	0% (0.8%)
Log Increments	117 (372)	11 (12)	91.4% (96.9%)
Signature of Log Increments	60 (1306)	57 (98)	51.3% (93.0%)

Table 1: The table shows the number of stationary 3-dimensional sequences for price, log increment, and 12-dimensional sliding signature of log increment data. The numbers in parentheses indicate the count of stationary/nonstationary 1-dimensional sequences, representing individual components of the 3D/12D sequences. Signatures were only applied to log increment sequences that were tested to be stationary. Note that we deem a multi-dimensional sequence stationary if its components test to be stationary.

After confirming the stationarity of log sequences, in our second experiment, we investigated whether the sliding signature transform preserves stationarity. Using stationary log increment sequences (117 sequences) from the first experiment, we applied a sliding window of length 16, step size 16 and computed depth-2 signatures, producing 12-dimensional sequences of length 128.

Results (Table 1, third row) indicate that 51.3% of multidimensional signature sequences remained stationary, although 93.0% of individual components remained stationary. This suggests that the sliding signature transform largely preserves stationarity, with only a few components failing the test. This empirically confirms that the signature transform is effective in maintaining stationarity.

5.2 Predictive Performance Evaluation

	Prices	Log Increments	Signature of Prices	Signature of Log Increments
Linear Regression	(2.56e-09, 1.00)	(2.55e-09, 1.00)	(5.27e-06, -0.58)	**(2.25e-09, 1.00)**
Random Forest	(4.21e-07, 0.83)	**(2.59e-09, 1.00)**	(1.80e-05, -3.45)	(2.68e-09, 1.00)
SVR	(8.85e-06, -1.34)	(1.01e-07, 0.97)	(8.95e-06, -1.24)	**(1.00e-07, 0.97)**
XGBoost	(2.97e-07, 0.92)	(2.31e-09, 1.00)	(1.76e-05, -3.40)	**(2.29e-09, 1.00)**
Neural Network	(7.08e-2, -2.27e4)	**(2.62e-5, -8.28)**	(3.66e-02, -1.16e4)	(5.55e-04, -1.80e2)

Table 2: Mean Squared Error (MSE) and R^2 scores for different machine learning models predicting prices using Signature of Prices, Signature of Log Increments, Prices, and Log Increments as their features. The first value in each cell represents the MSE, while the second represents the R^2 score. The best result in each row is highlighted in bold.

	Prices	Log Increments	Signature of Prices	Signature of Log Increments
Linear Regression	**(7.75e-06, 3.99e-05)**	(7.63e-06, 3.99e-05)	(4.43e-07, 4.07e-05)	(5.49e-06, 4.02e-05)
Random Forest	(8.51e-07, 4.07e-05)	(-2.95e-07, 4.08e-05)	(6.63e-07, 4.07e-05)	**(2.91e-06, 4.06e-05)**
SVR	**(3.53e-07, 4.07e-05)**	(-4.84e-07, 4.07e-05)	(3.53e-07, 4.07e-05)	(-4.84e-07, 4.07e-05)
XGBoost	(4.93e-07, 4.07e-05)	(-4.84e-07, 4.07e-05)	**(1.75e-06, 4.07e-05)**	(-4.84e-07, 4.07e-05)
Neural Network	(3.47e-08, 4.08e-05)	**(9.81e-07, 4.07e-05)**	(-2.33e-07, 4.08e-05)	(8.41e-07, 4.07e-05)

Table 3: Average profit per transaction and standard deviation for different machine learning models predicting prices using different feature types: raw prices, log increments, signature of prices, and signature of log increments. The best result in each row is highlighted in bold.

In the third experiment, we compared the predictive performance of models trained on log increment, non-stationary prices, signatures of log increments and signatures of prices.

We divided the log and price data into 8,024 non-overlapping sequences, each of length 33, with 75% of sequences used for training and the remaining 25% reserved for testing. Each model aimed to predict the final value of a sequence based on the preceding 32 values. For log increment sequences, we exponentiated the predicted increment and multiplied it by the penultimate price to allow direct comparison with raw price predictions.

We evaluated five machine learning models: Linear Regression, Random Forest, Support Vector Regression (SVR), eXtreme Gradient Boosting (XGBoost), and Neural Network[3]. The results, shown in Table 2, demonstrate that the signature of log increments generally yields superior predictive accuracy compared to the signature of raw prices, although the performance differences between models were often very small and may not always represent significant improvements. Specifically, Linear Regression, Random Forest, and XGBoost showed slightly better performance using log increment signatures, each achieving an MSE around 10^{-9} and $R^2 \approx 1.00$. SVR and Neural Network also performed well with log increments, achieving an MSE of 1.00×10^{-7} and an R^2 score of approximately 0.97. Despite the small differences, the results consistently highlight the utility of stationary sequences and the potential advantage of using signatures derived from them.

5.3 Practical Profitability Evaluation

In an additional trading experiment summarized in Table 3, we measured the practical profitability of predictions made by the same models using various feature sets. Profits were computed by initiating trades based on model predictions; long positions were opened if an increase in price was predicted, short positions were considered for predicted decreases. Each trade involved one unit of the GBP/USD forex pair. The average profits per transaction across all models and features were extremely small, typically in the order of 10^{-6}, accompanied by standard deviations around 4×10^{-5}. Notably, the standard deviation substantially exceeds the average profit, indicating a high variability in trading outcomes, which suggests considerable uncertainty and risk in practical applications. Although differences in profitability between models and features were minimal the best models were Linear Regression with raw prices, log increments and signature of log increment.

5.4 Augmented Signature Testing

In the final experiment, we tested whether augmented signature transforms improve predictive accuracy and profitability. We compared two augmented variants:

1. Sig(T, LL, B, D): Time augmentation, lead-lag, base-point augmentation, and dyadic windowing.

2. Sig(T, D): Only time augmentation and dyadic windowing.

Tables 4 and 5 show minimal but consistent improvements when using simpler augmented signatures (Sig(T,D)) in Random Forest and SVR models, achieving exceptionally low MSE, perfect R^2 scores and higher profits. The full augmentation variant (Sig(T,LL,B,D)) did not provide clear improvements, suggesting that simpler transformations might suffice for optimal predictive performance. Linear Regression using plain log increment signatures outperformed all augmented signatures model in terms of MSE and

[3]A three-layer feed-forward neural network ($32 \to 16 \to$output) with ReLU activations, batch normalization, and dropout regularization ($p = 0.2$), trained using Adam optimizer and early stopping.

R^2 but not in profitability. In all cases, profitability remained marginal and its relationship with low MSE remained unclear, consistent with earlier results.

Figure 4 highlights that Linear Regression with raw prices and Random Forest with Sig(T, D) produced the highest cumulative profits, albeit very small (approx. 0.01$ for 4 hours of trading). Practical considerations, including transaction costs and trade delays, would further reduce (or completely erase) profitability, highlighting that statistically accurate predictions do not necessarily translate into meaningful practical profits.

	Sig(T, LL, B, D)	**Sig(T, D)**	**Signature of Log Increments**
Linear Regression	(2.36e-09, 1.00)	(2.35e-09, 1.00)	**(2.25e-09, 1.00)**
Random Forest	(2.59e-09, 1.00)	**(2.38e-09, 1.00)**	(2.68e-09, 1.00)
SVR	(1.01e-07, 0.97)	**(1.00e-07, 0.98)**	(1.00e-07, 0.97)
XGBoost	**(2.26e-09, 1.00)**	(2.31e-09, 1.00)	(2.29e-07, 1.00)
Neural Network	**(4.60e-04, -1.11e2)**	(4.29e-1, -1.88e5)	(5.55e-04, -1.80e2)

Table 4: Mean Squared Error (MSE) and R^2 scores for different machine learning models predicting prices using different features: signatures of log increments and two different augmented signature transformations. T, LL, B, and D stand, respectively, for time-augmentation, lead-lag transformation, base-point augmentation, and dyadic windowing. The first value in each cell represents the MSE, while the second represents the R^2 score. The best result in each row is highlighted in bold.

	Sig(T, LL, B, D)	**Sig(T, D)**	**Signature of Log Increments**
Linear Regression	(9.88e-07, 4.07e-05)	(-3.20e-07, 4.04e-05)	**(5.49e-06, 4.02e-05)**
Random Forest	(5.62e-06, 4.04e-05)	**(7.39e-06, 4.01e-05)**	(2.91e-06, 4.06e-05)
SVR	(-4.82e-07, 4.07e-05)	**(-4.80e-07, 4.07e-05)**	(-4.84e-07, 4.07e-05)
XGBoost	**(-4.81e-07, 4.07e-05)**	(-4.83e-07, 4.07e-05)	(-4.84e-07, 4.07e-05)
Neural Network	(-1.75e-06, 4.07e-05)	**(2.45e-07, 4.07e-05)**	(8.41e-07, 4.07e-05)

Table 5: Average profit per transaction and standard deviation for different machine learning models predicting prices using different features: signatures of log increments and two different augmented signature transformations. T, LL, B, and D stand, respectively, for time-augmentation, lead-lag transformation, base-point augmentation, and dyadic windowing. The best result in each row is highlighted in bold.

6 Conclusions

In this dissertation, we have investigated the theoretical properties and practical utility of the signature transform, a powerful mathematical tool rooted in rough path theory and increasingly prominent in machine learning applications. Our primary contribution includes establishing and empirically validating a novel theoretical result: the sliding signature transform preserves the stationarity of underlying processes. This finding fills a critical theoretical gap and motivates the practical exploration of signatures derived from stationary time series, particularly in financial contexts.

Empirically, we demonstrated the stationarity of log increments derived from forex price data, aligning with classical financial models such as the Black–Scholes framework. Consistent with our theoretical prediction, sliding signatures largely maintained this

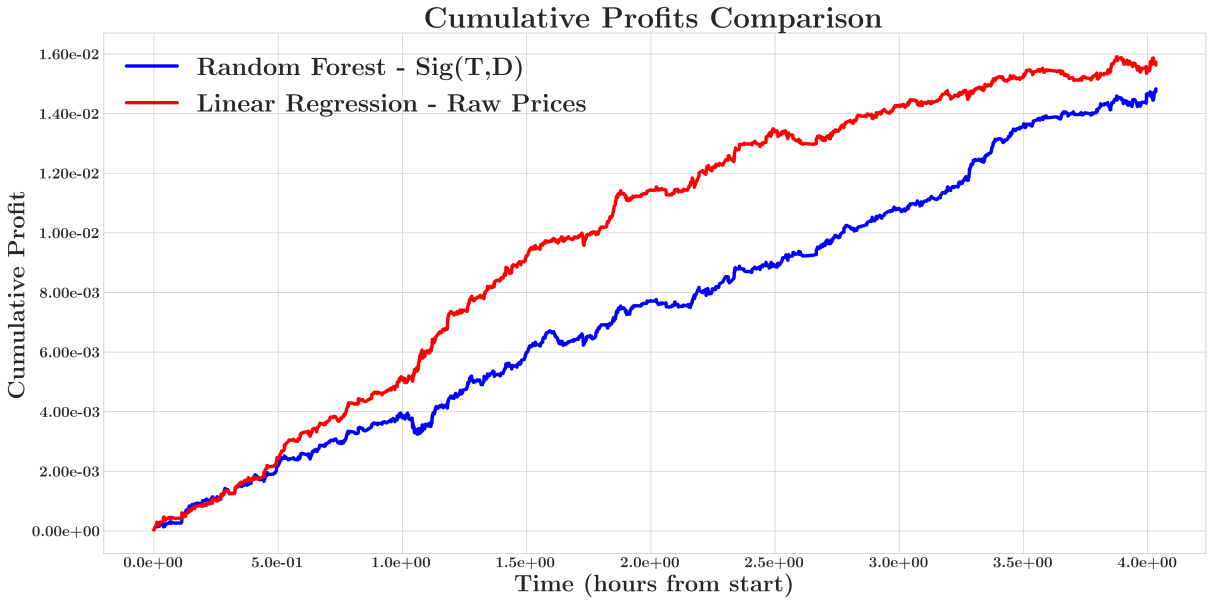

Figure 4: Comparison of cumulative profits between two trading strategies over time. The blue line represents a Random Forest model trained on augmented signature features, while the red line shows a Linear Regression model using raw prices. Both models were evaluated on the same test period, with profits calculated from simulated long/short positions based on the models' predictions. Time is measured in hours from the start of the testing period.

stationarity, with 51.3% of multidimensional sequences and 93.0% of individual components remaining stationary.

Our predictive experiments confirmed that models utilizing sliding signatures of stationary log increments achieve consistently superior statistical performance, achieving exceptionally low mean squared errors and near-perfect predictive accuracy across multiple machine learning algorithms. However, translating these statistical improvements into meaningful practical outcomes remains challenging. The simulated trading strategy we employed exhibited negligible average returns with high variability, underscoring the complexities involved in real-world financial applications. However, we only tested a relatively crude and simplistic trading strategy.

The exploration of path augmentations highlighted that simpler augmentation methods, such as combining time augmentation with dyadic windowing, can deliver competitive predictive performance compared to more elaborate approaches. This suggests potential efficiency gains without sacrificing predictive accuracy.

In conclusion, our study emphasizes both the significant theoretical advantages of the signature transform and the practical limitations encountered in translating these advantages into tangible financial returns. Future research should focus on bridging this gap, possibly through exploring advanced predictive models, incorporating transaction costs, examining multi-asset scenarios, and further refining the integration of signature methods into practical decision-making frameworks. Additionally, investigating computational efficiency and real-time applicability, particularly when combined with more sophisticated trading algorithms, presents a promising avenue for future research.

References

TrueFX Downloads – TrueFX. URL https://www.truefx.com/truefx-historical-downloads/.

Thomas Cass and Cristopher Salvi. Lecture notes on rough paths and applications to machine learning, 2024. URL https://arxiv.org/abs/2404.06583.

Kuo-Tsai Chen. Iterated integrals and exponential homomorphisms. *Proceedings of the London Mathematical Society*, s3-4(1):502–512, 1954. doi: https://doi.org/10.1112/plms/s3-4.1.502. URL https://londmathsoc.onlinelibrary.wiley.com/doi/abs/10.1112/plms/s3-4.1.502.

Kuo-Tsai Chen. Integration of paths, geometric invariants and a generalized baker-hausdorff formula. *Annals of Mathematics*, 65(1):163–178, 1957. ISSN 0003486X, 19398980. URL http://www.jstor.org/stable/1969671.

Kuo-Tsai Chen. Formal differential equations. *Annals of Mathematics*, 73(1):110–133, 1961. ISSN 0003486X, 19398980. URL http://www.jstor.org/stable/1970284.

Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning, 2025. URL https://arxiv.org/abs/1603.03788.

Thomas Cochrane, Peter Foster, Varun Chhabra, Maud Lemercier, Cristopher Salvi, and Terry Lyons. Sk-tree: a systematic malware detection algorithm on streaming trees via the signature kernel. *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*, pages 35–40, 2021. URL https://api.semanticscholar.org/CorpusID:231933900.

Adeline Fermanian. Embedding and learning with signatures. *Computational Statistics Data Analysis*, 157:107148, 2021. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2020.107148. URL https://www.sciencedirect.com/science/article/pii/S0167947320302395.

Benjamin Graham. Sparse arrays of signatures for online character recognition, 2013. URL https://arxiv.org/abs/1308.0371.

Lajos Gergely Gyurkó, Terry Lyons, Mark Kontkowski, and Jonathan Field. Extracting information from the signature of a financial data stream, 2014. URL https://arxiv.org/abs/1307.7244.

Olav Kallenberg. *Foundations of modern probability*. Springer, 3 edition, 2021.

Patrick Kidger and Terry Lyons. Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU. In *International Conference on Learning Representations*, 2021. https://github.com/patrick-kidger/signatory.

Patrick Kidger, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/d2cdf047a6674cef251d56544a3cf029-Paper.pdf.

Franz J. Kiraly and Harald Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20(31):1–45, 2019. URL http://jmlr.org/papers/v20/16-314.html.

Maud Lemercier, Cristopher Salvi, Theodoros Damoulas, Edwin Bonilla, and Terry Lyons. Distribution regression for sequential data. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3754–3762. PMLR, 13–15 Apr 2021. URL https://proceedings.mlr.press/v130/lemercier21a.html.

Terry Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310, 1998. URL http://eudml.org/doc/39555.

Terry Lyons and Zhongmin Qian. *System Control and Rough Paths*. Oxford University Press, 12 2002. ISBN 9780198506485. doi: 10.1093/acprof:oso/9780198506485.001.0001. URL https://doi.org/10.1093/acprof:oso/9780198506485.001.0001.

Terry Lyons and Nadia Sidorova. Sound compression: a rough path approach. In *Proceedings of the 4th International Symposium on Information and Communication Technologies*, WISICT '05, page 223–228. Trinity College Dublin, 2005. ISBN 1595931694.

Terry Lyons, Michael Caruana, and Thierry Lévy. *Differential Equations Driven by Rough Paths: Ecole d'Eté de Probabilités de Saint-Flour XXXIV-2004*. Lecture Notes in Mathematics. Springer Berlin Heidelberg, 2007. ISBN 9783540712855. URL https://books.google.co.uk/books?id=hOm5BQAAQBAJ.

Terry Lyons, Hao Ni, and Harald Oberhauser. A feature set for streams and an application to high-frequency financial tick data. In *Proceedings of the 2014 International Conference on Big Data Science and Computing*, BigDataScience '14, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450328913. doi: 10.1145/2640087.2644157. URL https://doi.org/10.1145/2640087.2644157.

Terry Lyons, Sina Nejad, and Imanol Perez Arribas. Numerical method for model-free pricing of exotic derivatives in discrete time using rough path signatures. *Applied Mathematical Finance*, 26(6):583–597, 2019. doi: 10.1080/1350486X.2020.1726784. URL https://doi.org/10.1080/1350486X.2020.1726784.

Paul Moore, Terry Lyons, and John Gallacher. Using path signatures to predict a diagnosis of alzheimer's disease. *PLOS ONE*, 14(9):e0222212, September 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0222212. URL http://dx.doi.org/10.1371/journal.pone.0222212.

James Morrill, Adeline Fermanian, Patrick Kidger, and Terry Lyons. A generalised signature method for time series. *arXiv preprint arXiv:2006.00873*, 2020a.

James Morrill, Andrey Kormilitzin, Alejo Nevado-Holgado, Sumanth Swaminathan, Samuel Howison, and Terry Lyons. Utilisation of the signature method to identify the early onset of sepsis from multivariate physiological time series in critical care monitoring. *Critical Care Medicine*, 48(10):e976–e981, 2020b.

James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pages 7829–7838. PMLR, 2021.

Imanol Perez Arribas, Guy M. Goodwin, John R. Geddes, Terry Lyons, and Kate E. A. Saunders. A signature-based machine learning model for distinguishing bipolar disorder and borderline personality disorder. *Translational Psychiatry*, 8(1), December 2018. ISSN 2158-3188. doi: 10.1038/s41398-018-0334-0. URL http://dx.doi.org/10.1038/s41398-018-0334-0.

Daniel Wilson-Nunn, Terry Lyons, Anastasia Papavasiliou, and Hao Ni. A path signature approach to online arabic handwriting recognition. In *2018 IEEE 2nd International Workshop on Arabic and Derived Script Analysis and Recognition (ASAR)*, pages 135–139, 2018. doi: 10.1109/ASAR.2018.8480300.

Weixin Yang, Terry Lyons, Hao Ni, Cordelia Schmid, and Lianwen Jin. *Developing the Path Signature Methodology and Its Application to Landmark- Based Human Action Recognition*, pages 431–464. Springer International Publishing, Cham, 2022. ISBN 978-3-030-98519-6. doi: 10.1007/978-3-030-98519-6_18. URL https://doi.org/10.1007/978-3-030-98519-6_18.