Learning Intermittent Time Series with the Partial Autocorrelation Function Integral Transform (PACFIT)

Justin M. Baker[1], Tyler Headley[2], Narayanan Kannan[1], Anand Somayajula[1], Adrien Weihs[1], P. Jeffrey Brantingham[3], & Andrea L. Bertozzi[1,4]

[1] Department of Mathematics, University of California, 520 Portola Plaza, Los Angeles, CA 90095, USA
[2]Department of Mathematics, Harvey Mudd College, 320 E. Foothill Blvd., Claremont, CA 91711, USA
[3] Department of Anthropology, University of California, 375 Portola Plaza, Los Angeles, CA 951553, USA
[4]California NanoSystems Institute, 570 Westwood Plaza, Los Angeles, CA 90095, USA
justin@math.ucla.edu

Abstract

Training global forecasting models across multiple time series (TS) is significantly more challenging for intermittent time series (ITS) due to large variations in the sparsity between TS. Common data augmentation techniques are computationally prohibitive for large datasets. This paper introduces the partial autocorrelation function integral transform (PACFIT), a data-driven approach to handling ITS. The partial autocorrelation function (PACF) plays a fundamental role in uncovering temporal dependencies within time series. We motivate the use of a PACF kernel using the integral transform as a theoretical framework. We then integrate PACFIT into the DeepAR (Salinas et al. 2020) learning pipeline. DeepAR is a flexible architecture that offers probabilistic forecasting for both real-valued and non-negative integer-valued data. We empirically validate our approach on an intermittent time series indicating the occurrence of violent crimes. Compared to baseline data transformations, PACFIT outperforms on mean absolute scaled error metrics, delivering non-zero predictions that effectively capture the variance and patterns in the data. PACFIT paves the way for reliable decision-making in energy management, supply chain optimization, and public safety.

Introduction

Intermittent time series (ITS) are characterized by extended periods of zero-valued data interspersed with non-zero occurrences. This characteristic describes many real-world phenomena, from episodic river flows (Forghanparast and Mohammadi 2022), to supply chain demand (Ghobbar and Friend 2003), and violent crime occurrences (Wang et al. 2019). Due to its broad applicability, reliably forecasting ITS impacts decision-making in renewable energy (Boylan and Syntetos 2021), sustainable operations (Çerağ Pinçe, Turrini, and Meissner 2021), and humanitarian response strategy (Welsh, Zimmerman, and Zane 2018). Accurate ITS forecasting remains an active area of research, as is intermittent demand forecasting (IDF) (Croston 1972), a subset of ITS consisting of exclusively non-negative integer values. For example, in Denver, neighborhood-level violent-crime incidents (e.g., assaults or robberies) often show weeks of zeros punctuated by brief, high-magnitude spikes tied to localized events or seasonal patterns, making forecasting difficult.

ITS forecasting is challenging due to the data's sparse and irregular (i.e., intermittent) nature. Training cost is reduced on large datasets by training a global model across multiple TS. In this case, the variations in the sparsity levels between ITS compound the forecasting challenges. This necessitates the development of methods that can handle intermittency while extracting intricate temporal interdependencies across multiple ITS.

Croston's (Croston 1972) breakthrough statistical method for IDF separately predicts the non-zero quantities and the time between occurrences. Croston's method operates under the strong assumption that quantity and duration are independent. Autoregressive models (Mohammadipour 2009) struggle with variations in the duration, and bootstrapping methods (Hasni et al. 2019) struggle with variations in sparsity.

Machine learning (ML) architectures are exceptionally effective in TS forecasting (Zhang, Patuwo, and Hu 1998). ITS adapted ML methods include extreme learning machines (Lolli et al. 2017), nearest neighbors (Petropoulos et al. 2013), support vector machines (Christian et al. 2021), feed forward networks (Kourentzes 2013), recurrent neural networks (Babai, Tsadiras, and Papadopoulos 2020), long-short-term-memory (LSTM) (Gauch et al. 2021), and transformers (Zhang, Xia, and Xie 2024). The applicability of these models is limited by their lack of explainability.

Probabilistic forecasting aims to address this limitation by providing uncertainty quantification (Tyralis and Papacharalampous 2024; Abdar et al. 2021), including confidence intervals and risk assessment, which are crucial for decision-making in real-world applications. Gaussian processes (Duvenaud et al. 2013; Roberts et al. 2013), Bayesian neural networks (Seeger, Salinas, and Flunkert 2016), temporal point processes (Park et al. 2021), Gaussian copulas (Salinas et al. 2019), and likelihood models (Salinas et al. 2020) empirically struggle in forecasting ITS (Kourentzes 2013). The DeepAR architecture (Salinas et al. 2020) is a foundational global model for probabilistic forecasting that offers both Gaussian and negative binomial likelihood models for real- and integer-valued forecasting, respectively.

Data pre-processing, through composition or transformation, enhances ITS forecasting accuracy by reducing data sparsity. Data fusion(Hu et al. 2023), clustering (Jha et al.

2015), and aggregation (Gauch et al. 2021) reduce sparsity but are labor-intensive and incur additional computational expense. Integral transforms like the wavelet transform, Fourier transform and diurnal periodic integral mapping (DPIM) (Wang et al. 2019) are promising methods for alleviating data sparsity while capturing temporal dependencies. The wavelet transform (Percival and Walden 2000; Wang et al. 2018b) and Fourier transform (Zhang, Aggarwal, and Qi 2017) alter the representation of the data. Time-shifted integral transforms (e.g. convolutions, cross-correlations, and autocorrelations) are ideal for retaining interpretable data representations. The diurnal periodic integral mapping (DPIM) (Wang et al. 2019) is interpretable, univariate preserving, and sparsity alleviating. However, this method is based on a heuristic 24-hour window, and lacks the generalizability of a data-driven approach.

The autocorrelation function (ACF) and partial autocorrelation function (PACF) are data-driven methods for uncovering temporal dependencies within TS. The ACF measures the correlation between a TS and its time-lagged values. The partial autocorrelation function (PACF) controls for the influence of intermediate lags by computing the correlation between a TS and a regression of its lagged values. Leveraging these tools can enhance the understanding and forecasting of temporal dependencies in ITS, challenging the prevailing assumptions about data irregularity.

Our contribution

We propose a novel partial autocorrelation function integral transform (PACFIT) and incorporate it into the DeepAR learning pipeline. Using the integral transform as a theoretical framework, PACFIT is motivated as a data-driven and parameterized time-shift integral transform. PACFIT offers three main benefits: (1) The data-driven nature enables universal applicability. (2) The parameterization allows it to be incorporated into the learning pipeline. (3) As a time-shifted integral transform, it is interpretable with probabilistic forecasting. These advantages are demonstrated by integrating PACFIT into the DeepAR learning pipeline. We empirically validate this approach by comparing it against existing data transformations, where PACFIT outperforms in mean absolute scaled error (MASE) and ACFIT outperforms in quartile coverage. Furthermore, the approach is generalizable across multiple datasets with a variety of time scales, real- and integer-valued data, and sparsity. Our approach is an enhanced global model for accurate probabilistic forecasting of ITS.

Related work

PACF plays a role in many forecasting pipelines. The coefficients are used for model validation (Barlas 1990; Livieris et al. 2020) and parameter selection (Hyndman and Khandakar 2008; Mohammadipour 2009; Flores, Engel, and Pinto 2012; Nystrup et al. 2020). In (Hyndman and Khandakar 2008; Mohammadipour 2009) the order of autoregression for ARIMA and INARMA is chosen via the dominant PACF coefficients. This approach has also been applied to support vector machines (Christian et al. 2021). Data fusion techniques also use the PACF coefficients as inputs to the architecture (Flores, Engel, and Pinto 2012; Nystrup et al. 2020). While these approaches may alleviate the challenges of ITS, they do not directly address the issues with sparse data. The advantage of incorporating PACF into the learning pipeline is the data-driven approach. The PACF representation of temporal dependencies enhances a model's explainability and expressivity.

Background

A TS $\boldsymbol{X} = (\boldsymbol{x}[1], \boldsymbol{x}[2], \ldots, \boldsymbol{x}[k])$ is a sequence of data points discretely sampled over time $t = t_1, \ldots, t_k$, where $\boldsymbol{X} \in \mathbb{R}^{d \times k}$ is said to be multivariate if $d > 1$ and univariate if $d = 1$. We will assume time is sampled at regular intervals (i.e., $t_i = t_1 + i\Delta t$ for some Δt) and write $\boldsymbol{X} = (x[1], x[2], \ldots, x[k])$ for univariate data. Data preprocessing is a map $f : \boldsymbol{X} \to \boldsymbol{Y}$ that transforms $\boldsymbol{X} \in \mathbb{R}^{d_x \times k_x}$ into $\boldsymbol{Y} \in \mathbb{R}^{d_y \times k_y}$, and f is considered univariate preserving if $d_x = d_y = 1$. We will focus on the maps f that can be written as an integral transform.

The integral transform is a fundamental framework that underlies several foundational approaches, making it a powerful framework for comparison.

Definition 1 (Integral transform)**.** *Let X and Y be function spaces where $X(t)$ is a function defined on the interval $[t_1, t_k] \subset \mathbb{R}$ and $Y(\boldsymbol{s})$ is a function of $\boldsymbol{s}$ where $\boldsymbol{s} \in \mathbb{R}^d$. An* integral transform *is a mapping $f : X \to Y$ defined by the equation*

$$Y(\boldsymbol{s}) = \int_{t_1}^{t_k} X(t) K(\boldsymbol{s}, t) dt,$$

where $K(\boldsymbol{s}, t)$ is a given function called the kernel *of the transform.*

Our theoretical understanding of data transforms is based on the relation between $\boldsymbol{s}$ and t, and the kernel function.

Definition 2 (Convolution function)**.** *Let X and Y be function spaces where $X(t)$ and $Y(t)$ are defined on $\mathbb{R}$. The* convolution *of X and K is defined by the equation*

$$Y(\tau) = (X * K)(\tau) = \int_{-\infty}^{\infty} X(t) K(\tau - t) dt,$$

where $K(\tau - t)$ is a given function called the convolution kernel.

From Definition 1, $f : X \to Y$ is an integral transform with kernel $K(\tau, t) = K(\tau - t)$. Here the notation $K(\tau, t)$ is used in place of $K(\boldsymbol{s}, t)$ to emphasize that t and τ are both elements of $\mathbb{R}$. With this understanding, observe (1) $t, \tau \in \mathbb{R}$ leads to a univariate transform (2) $K(\tau - t)$ is a reversed time-shift kernel. The correlation function(Gubner and Safari 2006) is a time-shifted integral transform without time reversal.

Definition 3 (Correlation function)**.** *Let X and Y be function spaces where $X(t)$ and $Y(t)$ are defined on $\mathbb{R}$. The continuous cross-correlation integral is given by*

$$Y(\tau) = \int_{-\infty}^{\infty} X(t) K(t + \tau) dt.$$

When $K(t+\tau) = X(t+\tau)$, the correlation is taken with respect to (w.r.t.) itself (i.e., auto-correlation).

Definition 4 (Autocorrelation function (ACF))**.** *Let X and Y be function spaces where $X(t)$ and $Y(t)$ are defined on $\mathbb{R}$. The continuous autocorrelation integral is given by*

$$Y(\tau) = \int_{-\infty}^{\infty} X(t)X(t+\tau)dt.$$

Remark 1 (Notion of Time Lags)**.** *Without loss of generality, the ACF may be defined as*

$$Y(\tau) = \int_{-\infty}^{\infty} X(t)X(t-\tau)dt.$$

This is particularly useful when we want to consider τ as a variable that lags *time. This expression is distinct from the time reversal expression containing $-t$.*

Remark 2 (Assumptions for ITS)**.** *The autocorrelation at $\tau = 0$ is otherwise known as the energy of the system and is given by*

$$Y(0) = \int_{-\infty}^{\infty} X(t)^2 dt.$$

It will be particularly useful for normalizing the coefficients of the ACF. In addition, we assume that $Y(0)$ is finite, i.e.

$$\left(\int_{-\infty}^{\infty} |X(t)|^2 dt\right)^{\frac{1}{2}} < \infty.$$

This is known as the L^2 function space, and it is a reasonable assumption for time series analysis to assume $X(t) \in L^2$. Most importantly, L^2 assumption is compatible with piece-wise constant and piece-wise discontinuous functions that arise in ITS. The other assumption we make is that $||X(t)||_2^2 > 0$, i.e. there exist non-zero elements of $X(t)$.

Limitations of existing methods

The Fourier and wavelet transforms play a crucial role in understanding the frequency content and the temporal dependencies in the data. However, the wavelet transform is not univariate preserving and the Fourier transform represents the data in terms of its frequencies. This means that both transforms change the representation of the data in a manner that is not readily interpretable by probabilistic forecasting methods.

The Fourier transform The Fourier transform is closely related to the autocorrelation function (Papoulis 1962), but decomposes the data into frequency components given by s. For $t, s \subset \mathbb{R}$, the Fourier transform of $X(t)$ is defined by

$$Y(s) = \int_{-\infty}^{\infty} X(t)e^{-ist}dt.$$

From Definition 1, we see that the Fourier transform is an integral transform with kernel $K(s,t) = e^{-ist}$. While $s \subset \mathbb{R}$ and $t \subset \mathbb{R}$, the Fourier transform is not a convolution or correlation due to the multiplication of s and t in the kernel. As a data transformation, this is useful for understanding the frequency content of the data, but not for generating easily interpretable probabilistic forecasts.

The wavelet transform is another powerful tool for understanding the frequency content of a signal. The wavelet transform of a signal $X(t)$ is given by

$$Y(a,b) = \int_{-\infty}^{\infty} X(t)\psi_{a,b}(t)dt,$$

where $\psi_{a,b}(t)$ is the wavelet function

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \frac{\sin(2\pi\frac{t-b}{a}) - \sin(\pi\frac{t-b}{a})}{\pi\frac{t-b}{a}},$$

and $\boldsymbol{s} = (a,b) \subset \mathbb{R}^2$. Unlike the Fourier transform, which decomposes the signal based solely on frequency, the wavelet transform provides localized frequency information by analyzing the signal at various scales and positions. However, this also means that the wavelet transform is not univariate preserving, making it less interpretable for probabilistic forecasting.

Diurnal periodic integral mapping (DPIM) DPIM (Wang et al. 2018a) is a successful time-shift transform for ITS forecasting. The kernel is defined as

$$K(\tau,t) = \begin{cases} 1 & 0 < t-\tau < (t \bmod(24)) \\ 0 & \text{otherwise} \end{cases}$$

where $\bmod(24)$ is chosen to represent the 24-hour (i.e. diurnal) cycle of the data. Notice that this function is very nearly a correlation function, were it not for the enforced periodicity.

A parameterized integral mapping (PIM) The heuristic cyclic nature is parameterizable for $\theta \in \mathbb{R}$. The parameterized kernel $K_\theta(\tau,t)$ is defined as

$$K_\theta(\tau,t) = \begin{cases} 1 & 0 < t-\tau < t - \theta\left\lfloor\frac{t}{\theta}\right\rfloor \\ 0 & \text{otherwise} \end{cases}.$$

As a static transformation, this method works well if θ can be chosen heuristically or selected by hyperparameter optimization techniques. However, incorporating PIM into a learning pipeline is not viable because the floor function is discontinuous activation function w.r.t. θ.

Proposition 1 (Discontinuity of PIM)**.** *Let $Y_\theta(\tau)$ be the parameterized integral map defined by*

$$Y_\theta(\tau) = \int_{-\infty}^{\infty} X(t)K_\theta(\tau,t)dt$$

$$K_\theta(\tau,t) = \begin{cases} 1 & 0 < t-\tau < t - \theta\left\lfloor\frac{t}{\theta}\right\rfloor \\ 0 & \textit{otherwise} \end{cases}$$

for $X \in L^2$. Then $Y_\theta(\tau)$ is non-differentiable w.r.t. θ.

This implies that PIM cannot be integrated into a learning pipeline to learn θ. This leaves the value of θ as a hyperparameter, limiting PIM as data-driven technique.

Partial Autocorrelation Integral Transform

The core of our approach is the adaptation of the autocorrelation function into a data-driven and parameterized transform. The key idea is to use the ACF as the kernel,

$$\begin{aligned} Y(s) &= \int_{t_1}^{s} X(\tau)K(\tau)d\tau \\ K(\tau) &= \int_{-\infty}^{\infty} X(t)X(t-\tau)dt. \end{aligned}$$

It is evident that the kernel is data-driven, but we would also like it to be parameterizable.

Definition 5 (Continuous-time ACFIT)**.**

$$\begin{aligned} Y_\theta(s) =& \int_{t_1}^{s} X(\tau)K_\theta(\tau)d\tau \\ K_\theta(\tau) =& \rho(\tau)\cdot\sigma\left(\frac{|\rho(\tau)| - \theta\sqrt{\mathrm{Var}(\rho(\tau))}}{\delta}\right) \\ \rho(\tau) =& \frac{\int_{-\infty}^{\infty} X(t)X(t-\tau)dt}{\int_{-\infty}^{\infty} X(t)^2 dt} \\ \mathrm{Var}(\rho(\tau)) =& \frac{1}{E}\int_{-\infty}^{\infty} S^2(f)\cos^2(2\pi f\tau)df \\ & - \left(\frac{1}{E}\int_{-\infty}^{\infty} S(f)\cos(2\pi f\tau)df\right)^2 \\ S(f) =& \int_{-\infty}^{\infty} \rho(\tau)\cos(2\pi f\tau)d\tau \end{aligned}$$

where θ is a learnable parameter, $\rho(\tau)$ are the autocorrelation coefficients, $S(f)$ is the spectral power density of $\rho(\tau)$, and $\sigma(\cdot)$ is the sigmoid function.

To normalize the ACF kernel, the autcorrelation coefficients $\rho(\tau)$ are used. These values are also soft thresholded based on the confidence interval (CI) of the null-hypothesis. The confidence interval around the null hypothesis is given by $\theta\sqrt{\mathrm{Var}(\rho(\tau))}$ and tests whether ρ is significantly different from zero. If the correlation falls outside the interval, it indicates statistically significant autocorrelation at that lag. To threshold for this value, we use the sigmoid function, σ, applied to the θ parameterized CI, with threshold intensity δ.

Parameterizability

To show that this parameterization is viable for learning, we show that ACFIT is differentiable w.r.t. θ. First, notice that the normalization technique bounds the magnitude of the coefficients ρ.

Proposition 2 (Boundedness of the autocorrelation coefficients)**.** *For $X(t) \in L^2$, the autocorrelation coefficients*

$$\rho(\tau) = \frac{\int_{-\infty}^{\infty} X(t)X(t-\tau)dt}{\int_{-\infty}^{\infty} X(t)^2 dt}$$

are bounded in magnitude for all τ.

Proof. For $X(t) \in L^2$, there exists $M \in \mathbb{R}$, such $|\rho(\tau)| \leq M$ for all s. This follows directly from the Cauchy-Schwarz inequality

$$\begin{aligned} |\rho(\tau)| &= \left|\frac{\int_{-\infty}^{\infty} X(t)X(t-\tau)dt}{\int_{-\infty}^{\infty} X(t)^2 dt}\right| \\ &\leq \frac{\left(\int_{-\infty}^{\infty} |X(t)|^2 dt\right)^{\frac{1}{2}}\left(\int_{-\infty}^{\infty} |X(t-\tau)|^2 dt\right)^{\frac{1}{2}}}{||X(t)||_2^2} \\ &= \frac{||X(t)||_2^2}{||X(t)||_2^2} \leq 1. \end{aligned}$$

□

Proposition 3 (Boundedness of the variance)**.** *For $X(t) \in L^2$ and $||X(t)||_2^2 > 0$, the variance operator*

$$\begin{aligned} \mathrm{Var}(\rho(\tau)) =& \frac{1}{E}\int_{-\infty}^{\infty} S^2(f)\cos^2(2\pi f\tau)df \\ & - \left(\frac{1}{E}\int_{-\infty}^{\infty} S(f)\cos(2\pi f\tau)df\right)^2 \end{aligned}$$

are bounded for all τ.

Proof. Parseval's identity shows that $S(f)$ is integrable for $X \in L^2$, and given by

$$\int_{-\infty}^{\infty} S^2(f)df = \int_{-\infty}^{\infty} \rho(\tau)^2 = E < \infty.$$

Considering $\cos(2\pi f\tau) \in [-1, 1]$, it is clear that

$$0 \leq \frac{1}{E}\int_{-\infty}^{\infty} S^2(f)\cos^2(2\pi f\tau)df \leq \frac{E}{E} = 1.$$

The Wiener-Khinchin theorem states that

$$\rho(\tau) = \int_{-\infty}^{\infty} S(f)\cos(2\pi f\tau)df,$$

which from Proposition 2 implies

$$0 \leq \left(\int_{-\infty}^{\infty} S(f)\cos(2\pi f\tau)df\right)^2 = \rho(\tau)^2 \leq 1.$$

Therefore $0 \leq \mathrm{Var}(\rho(\tau)) \leq 1$ for all τ. □

This agrees with our intuition about the properties of ρ and Var, that $|\rho(\tau)| < 1$ and $0 \leq \mathrm{Var}(\rho(\tau)) \leq 1$.

Proposition 4 (Differentiability of ACFIT)**.** *For $X(t) \in L^2$, the ACFIT*

$$\begin{aligned} Y_\theta(s) &= \int_{t_1}^{s} X(\tau)K_\theta(\tau)d\tau \\ K_\theta(\tau) &= \rho(\tau)\cdot\sigma\left(\frac{|\rho(\tau)| - \theta\sqrt{\mathrm{Var}(\rho(\tau))}}{\delta}\right) \end{aligned}$$

is differentiable w.r.t. θ.

Proof. We will show that the integral,

$$\frac{\partial Y_\theta(s)}{\partial\theta} = \int_{t_1}^{s} X(\tau)\frac{\partial K_\theta(\tau)}{\partial\theta}d\tau$$

is well-defined for all θ. From Proposition 3, $\mathrm{Var}(\rho(\tau))$ is well-defined so that $\frac{\partial K}{\partial \theta}$ exists and is given by

$$\frac{\partial K_\theta(\tau)}{\partial \theta} = -\frac{\sqrt{\mathrm{Var}(\rho(\tau))}}{\delta} \cdot \rho(\tau) \cdot \sigma' \left(\frac{|\rho(\tau)| - \theta\sqrt{\mathrm{Var}(\rho(\tau))}}{\delta} \right).$$

Therefore, we consider whether the expression $\frac{\partial Y(\theta,s)}{\partial \theta}$ is integrable. First note that σ is bounded, and therefore $\sigma'(x) = \sigma(x)(1-\sigma(x))$ is bounded. The boundedness from Propositions 2 and 3, and integrability of $X \in L^2$ imply that $\frac{\partial Y}{\partial \theta}$ is a well-defined integral on $[t_1, t_k]$. □

The partial autocorrelation function integral transform (PACFIT) aims to control for the influence of intermediate values of ρ by regression.

Definition 6 (Continuous-time PACFIT)**.**

$$\begin{aligned}
Y_\theta(s) &= \int_{t_1}^{s} X(\tau) K_\theta(\tau) d\tau \\
K_\theta(\tau) &= \phi(\tau) \cdot \sigma \left(\frac{|\phi(\tau)| - \theta\sqrt{\mathrm{Var}(\phi(\tau))}}{\delta} \right) \\
\phi(\tau) &= \frac{1}{4E} \int_{-\infty}^{\infty} [X(t) - \hat{X}(\tau,t)][X(t-\tau) - \hat{X}(\tau,t)] dt \\
\hat{X}(\tau,t) &= \int_{t-\tau}^{t} \hat{\beta}(\xi) X(\xi) d\xi \\
\hat{\beta} &= \operatorname*{arg\,min}_{\beta \in L^2} \left\| X(t) - \int_{t-\tau}^{t} \beta(\xi) X(\xi) d\xi \right\|_2^2
\end{aligned}$$

Here the kernel $K_\theta(\tau)$ uses $\phi(\tau)$, the partial autocorrelation coefficients. These values control for the influence of intermediate lags by removing the influence of past data captured in $\hat{X}(t,\theta)$. In particular, $\hat{X}(t,\theta)$ reflects the influence of past data through a smoothed estimate of the time series with weights $\hat{\beta}$. These are the weights that minimize the L^2 reconstruction error between $X(t)$ and $\hat{X}(t)$.

PACFIT's differentiability is ensured from the boundedness of ϕ and $\mathrm{Var}(\phi)$. Again we observe that $|\phi(\tau)| \leq 1$ and $0 \leq \mathrm{Var}(\phi(\tau) \leq 1$ which follow identically to Propositions 3 and 4.

Discrete PACFIT

On regularly sampled discrete time intervals PACFIT is defined by

$$\begin{aligned}
y_\theta[n] &= \sum_{k=1}^{n} x[k] K_\theta[k] \\
K_\theta[k] &= \phi[n,k] \cdot \sigma \left(\frac{|\phi[n,k]| - \theta\sqrt{\mathrm{Var}(\phi[n,k])}}{\delta} \right) \\
\phi[n,k] &= \frac{\rho[k] - \sum_{j=1}^{k-1} \phi[n-1,j]\rho[k-j]}{1 - \sum_{j=1}^{k-1} \phi[n-1,j]\rho[j]} \\
\rho[k] &= \frac{\sum_{n=0}^{N} x[n]x[n-k]}{\sum_{n=0}^{N} x[n]^2} \\
\mathrm{Var}(\phi[n,k]) &= \frac{1}{n} \left(1 + 2\sum_{j=1}^{k-1} \phi^2[n,j] \right).
\end{aligned}$$

The Durbin-Levinson algorithm is used for computing the PACF coefficients $\phi_n[k]$ and Bartlett's formula for computing the variance.

The scalar multiplication of $\sqrt{\mathrm{Var}(\phi[n,k])}$ by θ poses challenges in learning, particularly as θ varies with the learning rate. To address this, we observe that the CIs for the null-hypothesis are centered around the line $y = 0$, and the maximum boundary of the CI can be directly learned as a threshold between 0 and 1. Furthermore, we separately parameterize the positive and negative thresholds with parameters α and β, respectively.

Definition 7 (PACFIT Neural Network)**.** *The PACFIT neural network is defined as*

$$\begin{aligned}
y_{\alpha,\beta}[n] &= \sum_{k=1}^{n} x[k] K_{\alpha,\beta}[k] \\
K_{\alpha,\beta}[k] &= \begin{cases} \phi[n,k] \cdot \sigma\left(\frac{\phi[n,k]-\alpha}{\delta}\right) & \phi[n,k] > 0 \\ \phi[n,k] \cdot \sigma\left(\frac{\beta-\phi[n,k]}{\delta}\right) & \phi[n,k] < 0 \\ 0 & \textit{otherwise} \end{cases}
\end{aligned}$$

for learnable parameters α, β *where* $0 \leq \alpha, \beta \leq 1$

The constraints on α and β are enforced using the scaled $\sigma(\frac{\cdot}{\delta})$ function, where we apply a scaling factor of $\delta = 0.01$ for α, β and the kernel threshold. This approach helps to non-dimensionalize the problem, mitigating the effects of scaling factors on the learning rate. This approach simplifies the CI to a maximum threshold, but works well in practice.

Region of Undefined ACFIT/PACFIT. We remark that ACF and PACF can be undefined when the lag-0 autocovariance is zero, in which case their respective normalization techniques result in an undefined value. This occurs when the series is constant which is typical when zero-intervals that constitute the intermittency are longer than the selected lag time. To overcome this issue, we define the ACFIT values as 0 when this occurs.

Computational Complexity The computation of ACFIT and PACFIT is determined by the computational complexity of ACF and PACF which depend on the length of the lag. Using the statsmodels tools in Python, we observe an upper bound on the time complexity of $\mathcal{O}(n \cdot k^2)$.

Methods

We proceed by incorporating the PACFIT neural network into the DeepAR learning pipeline. To rigorously test this approach we consider several metrics for ITS forecasting and a variety of challenging datasets.

DeepAR Model

DeepAR (Salinas et al. 2020) is well suited for learning large scale data, as it is an efficient global model trained on univariate data. It generates accurate predictions for TS datasets varying in scale and velocity. And, it is a popular benchmark model that requires little to no hyperparameter tuning

DeepAR is a multilayer, autoregressive NN built with LSTM cells for k-step ahead, probabilistic forecasting. It learns by minimizing the negative log-likelihood

$$\mathcal{L} = \sum_{i=1}^{M}\sum_{k=1}^{N} -\log \ell(x_i[k] \mid f_\theta(\boldsymbol{h}_i[k])),$$

over a set of M univariate TS $\{\boldsymbol{X}_i\}_{i=0,\dots,M}$ where $\boldsymbol{X}_i = \{x_i[k]\}_{k=0,\dots,N}$, ℓ is the likelihood function, f_θ is a parameterized NN, and $\boldsymbol{h}_i[k]$ is a deterministic function of the input.

In addition, DeepAR offers Gaussian and negative-binomial likelihood functions for forecasting ITS and IDF, respectively. The Gaussian likelihood is defined by

$$\begin{aligned} \ell_G(x|\mu,\sigma) &= (2\pi\sigma^2)^{-\frac{1}{2}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \\ \mu(\boldsymbol{h}_i[k]) &= \boldsymbol{w}_\mu^T \boldsymbol{h}_i[k] + b_\mu \\ \sigma(\boldsymbol{h}_i[k]) &= \log(1 + e^{\boldsymbol{w}_\sigma^T \boldsymbol{h}_i[k] + b_\sigma}) \end{aligned}$$

and the negative-binomial likelihood is defined by

$$\begin{aligned} \ell_{NB}(x|\mu,\alpha) &= \frac{\Gamma(x+\frac{1}{\alpha})}{\Gamma(x+1)\Gamma(\frac{1}{\alpha})} \left(\frac{1}{1+\alpha\mu}\right)^{\frac{1}{\alpha}} \left(\frac{\alpha\mu}{1+\alpha\mu}\right)^{x} \\ \mu(\boldsymbol{h}_i[k]) &= \log(1 + e^{\boldsymbol{w}_\mu^T \boldsymbol{h}_i[k] + b_\mu}) \\ \alpha(\boldsymbol{h}_i[k]) &= \log(1 + e^{\boldsymbol{w}_\alpha^T \boldsymbol{h}_i[k] + b_\alpha}). \end{aligned}$$

Datasets

The Denver crime (Denver Police Department 2025) dataset.

Denver crime data The Denver crime dataset is a novel dataset based on crime statistics published by the city of Denver (Denver Police Department 2025), from 2012 to August 2024. There are 988,799 crimes, each with the associated time of occurrence, report time, geological position in longitude and latitude, neighborhood, type of crime, and number of victims. The data is resampled on a daily scale, and aggregated by neighborhood into 78 TS.

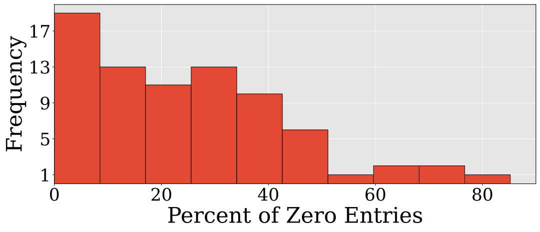
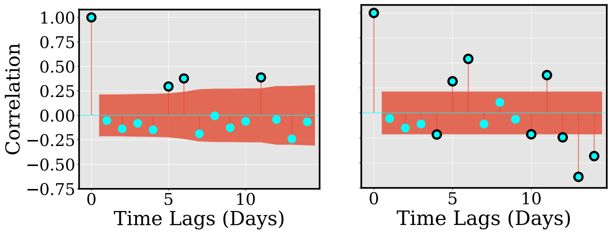


Figure 1: Sparsity of occurrences among 78 neighborhoods.

Figure 1 illustrates the sparsity among the neighborhoods in the dataset. The mean level of zero entries is 25%. Several neighborhoods have less than 50% of non-zero entries.

Figure 2 illustrates the autocorrelation correlogram for the Denver crime data. The coefficients for ACF and PACF for 14 lag values on a single neighborhood, with the confidence interval(CI) of the null hypothesis is shaded in red. The values that are statistically significant from the 75% CI, are outlined in black. We observe the least regularity on this dataset. Nevertheless, both the ACF and PACF yield several significant coefficients.



(a) ACF

(b) PACF

Figure 2: Correlation coefficients (blue) for 14 daily time lags on one TS of the Denver crime dataset. Using the 75% CI around the null hypothesis (shaded in red) as a threshold, the statistically significant coefficients are outlined in black.

Learning Pipeline

The learning pipeline is constructed using GluonTS (Alexandrov et al. 2020). We incorporate the parameterizable transforms directly into the DeepAR architecture, and perform slight modifications to the scale handling. We also utilize the negative-binomial likelihood function for non-negative integer valued predictions.

The parameterized transforms are included into the DeepAR architecture via $\boldsymbol{h}_i[k]$, the deterministic function of the input. We consider four transforms Identity, DPIM, ACFIT and PACFIT that can be used to construct $\boldsymbol{h}_i[k]$ from $x_i[k]$. In particular, we let $\boldsymbol{h}_i[k] = \mathrm{T}_\theta(x_i[k])$ where T_θ is the applied transformation, possibly the identity. The application of a, possibly parameterized, deterministic function applied to the input does not alter the training of DeepAR.

Two weighting techniques are used by DeepAR for scale handling. First, it uses mean scaling to normalize the values of the input and the output data. Because sparsity of ITS data causes the mean value to be very close to zero, we scale the data based on the mean of the non-zero elements in the data. In addition, DeepAR uses a weighted sampling to select training batches, where the weights are given by the mean value of the data. We adopt the mean of the non-zero data as weights in the sampling technique.

The DeepAR architecture optionally concatenates lagged data to form a multivariate input. To isolate for the effects of the transform and to preserve the univariate nature of the data, we remove the lag concatenation.

We use the default hyperparameters for DeepAR as implemented in GluonTS. DeepAR is a robust technique that requires little to no hyperparameter selection, making it an ideal candidate for benchmarking the data transforms. Training is performed by minimizing the NLL loss via the Adam optimizer. The learning rate is 10^{-3}, and the model is trained for 1000 epochs. Training and testing are performed on a single A100 GPU provided by GoogleColab (Google 2024).

Evaluation Metrics

Let $y_i[k]$ be the labels of a single TS at time step t_k. The value $\hat{y}_i[k]$ is the prediction of $y_i[k]$ made by $f_\theta(\boldsymbol{h}_i[k])$. Evaluating ITS forecasts is difficult because common metrics may not accurately capture the fidelity between $y_i[k]$

and $\hat{y}_i[k]$ for all time in all TS (Mohammadipour 2009; Hyndman and Koehler 2006).

The mean absolute percentage error (MAPE) is undefined when the observed demand is zero. Absolute accuracy measures like mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE) are unsuited for global models because they are not scale invariant.

We use mean absolute scaled error (MASE) and quantile coverage as evaluation metrics. Coverage alone does not capture the sharpness of the prediction intervals, as very wide intervals can achieve high coverage. We use coverage primarily to assess calibration and MASE to evaluate point-forecast accuracy.

MASE The mean absolute scaled error (MASE) defined by

$$\mathcal{L}_{\text{MASE}}(y, \hat{y}) = \frac{1}{T} \sum_{k=1}^{T} \left| \frac{y[k] - \hat{y}[k]}{\frac{1}{T-1} \sum_{i=2}^{T} |y[i] - y[i-1]|} \right|$$

is nicely interpretable for ITS. The naive forecast threshold for MASE is 1, i.e., if $\text{MASE} < 1$ a model achieves better than naive forecasting accuracy.

Quantile coverage. Let $\hat{y}_i^{(\rho)}[k]$ denote the predicted ρ-quantile of the forecast distribution for series i at time step k (over the evaluation horizon $k = 1, \ldots, T$). We define empirical ρ-coverage (Salinas et al. 2020) as

$$\text{Cov}_\rho(Y, \hat{Y}) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{k=1}^{T} \mathbf{1}\Big\{ y_i[k] \leq \hat{y}_i^{(\rho)}[k] \Big\}.$$

The coverage error is

$$\text{CE}_\rho = \left| \text{Cov}_\rho(Y, \hat{Y}) - \rho \right|.$$

Experimental Results

We find that PACFIT outperforms ACFIT, DPIM, and Identity in mean absolute scaled error, while also making non-zero predictions that are effective in capturing the variance and patterns in the data. Furthermore, we observe that PACFIT is generalizable to a wide range of datasets including intermittent demand forecasting. Finally, we demonstrate that PACFIT is robust to the choice of hyperparameters, and is capable of making accurate forecasts at any temporal resolution.

Denver Crime

In this task, we extend the PACFIT and ACFIT transforms to intermittent time series forecasting on the Denver Crime dataset. We provide a direct comparison between DPIM, ACFIT and PACFIT on weekly time lags. This assesses the benefits of ACFIT and PACFIT over DPIM on the DPIM designed task.

Table 1 reports the MASE and quantile coverage of each method on the Denver crime dataset. We observe that the PACFIT obtains the best MASE values. ACFIT outperforms on quantile coverage and achieves the second strongest MASE. Compared to the baseline without transforms, DPIM only outperforms on quantile coverage for fifty and ninety percent quantiles.

	MASE	**Q=0.1**	**Q=0.5**	**Q=0.9**
Identity	0.804	0.309	0.619	0.903
DPIM	0.805	0.292	0.636	0.935
ACFIT (Ours)	0.787	**0.310**	**0.668**	**0.953**
PACFIT (Ours)	**0.758**	0.293	0.616	0.922

Table 1: A comparison of methods trained on the Denver Crime dataset. We observe that ACFIT and PACFIT outperform DPIM and the Identity (no) transform.

Concluding Remarks

We proposed the PACFIT, a data-driven and parameterized time-shift integral transform for generalizable use in ITS forecasting. These integral transforms produce non-intermittent representations of intermittent data improving the accuracy of ITS forecasting. Our work is focused on demonstrating the theoretical motivation and benefit of ACFIT and PACFIT. It is limitted to utilizing only one global model and only one primary dataset. We leave the broader impacts of the ACFIT and PACFIT methods to future work.

Acknowledgments

This work is supported in part by NSF grant SCC-2125319.

References

Abdar, M.; Pourpanah, F.; Hussain, S.; Rezazadegan, D.; Liu, L.; Ghavamzadeh, M.; Fieguth, P.; Cao, X.; Khosravi, A.; Acharya, U. R.; Makarenkov, V.; and Nahavandi, S. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76: 243–297.

Alexandrov, A.; Benidis, K.; Bohlke-Schneider, M.; Flunkert, V.; Gasthaus, J.; Januschowski, T.; Maddix, D. C.; Rangapuram, S.; Salinas, D.; Schulz, J.; Stella, L.; Türkmen, A. C.; and Wang, Y. 2020. GluonTS: Probabilistic and Neural Time Series Modeling in Python. *Journal of Machine Learning Research*, 21(116): 1–6.

Babai, M. Z.; Tsadiras, A.; and Papadopoulos, C. 2020. On the empirical performance of some new neural network methods for forecasting intermittent demand. *IMA Journal of Management Mathematics*, 31(3): 281–305.

Barlas, Y. 1990. An autocorrelation function test for out put validation. *SIMULATION*, 55(1): 7–16.

Boylan, J.; and Syntetos, A. 2021. *Intermittent Demand Forecasting: Context, Methods and Applications*. ISBN 9781119976080.

Christian, K.; Roy, A. F. V.; Yudianto, D.; and Zhang, D. 2021. Application of optimized Support Vector Machine in monthly streamflow forecasting: using Autocorrelation Function for input variables estimation. *Sustainable Water Resources Management*, 7(3): 29.

Croston, J. D. 1972. Forecasting and Stock Control for Intermittent Demands. *Operational Research Quarterly (1970-1977)*, 23(3): 289–303.

Denver Police Department. 2025. Denver Open Data Catalog. https://opendata-geospatialdenver.hub.arcgis.com/. City and County of Denver. Accessed 2025-11-02. Open data portal.

Duvenaud, D.; Lloyd, J.; Grosse, R.; Tenenbaum, J.; and Zoubin, G. 2013. Structure Discovery in Nonparametric Regression through Compositional Kernel Search. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 1166–1174. Atlanta, Georgia, USA: PMLR.

Flores, J. H. F.; Engel, P. M.; and Pinto, R. C. 2012. Autocorrelation and partial autocorrelation functions to improve neural networks models on univariate time series forecasting. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Forghanparast, F.; and Mohammadi, G. 2022. Using Deep Learning Algorithms for Intermittent Streamflow Prediction in the Headwaters of the Colorado River, Texas. *Water*, 14(19).

Gauch, M.; Kratzert, F.; Klotz, D.; Nearing, G.; Lin, J.; and Hochreiter, S. 2021. Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network. *Hydrology and Earth System Sciences*, 25(4): 2045–2062.

Ghobbar, A. A.; and Friend, C. H. 2003. Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model. *Computers & operations research*, 30(14): 2097–2114.

Google. 2024. Google Colaboratory. https://colab.research.google.com/. Accessed: 2024-09-08.

Gubner, J.; and Safari, a. O. M. C. 2006. *Probability and Random Processes for Electrical and Computer Engineers*. EngineeringPro collection. Cambridge University Press.

Hasni, M.; Aguir, M.; Babai, M.; and Jemai, Z. 2019. On the performance of adjusted bootstrapping methods for intermittent demand forecasting. *International Journal of Production Economics*, 216: 145–153.

Hu, K.; Li, L.; Tao, X.; Velásquez, J. D.; and Delaney, P. 2023. Information fusion in crime event analysis: A decade survey on data, features and models. *Information Fusion*, 100: 101904.

Hyndman, R. J.; and Khandakar, Y. 2008. Automatic Time Series Forecasting: The forecast Package for R. *Journal of Statistical Software*, 27(3): 1–22.

Hyndman, R. J.; and Koehler, A. B. 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4): 679–688.

Jha, A.; Ray, S.; Seaman, B.; and Dhillon, I. S. 2015. Clustering to forecast sparse time-series data. In *2015 IEEE 31st International Conference on Data Engineering*, 1388–1399.

Kourentzes, N. 2013. Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, 143(1): 198–206.

Livieris, I. E.; Stavroyiannis, S.; Pintelas, E.; and Pintelas, P. 2020. A novel validation framework to enhance deep learning models in time-series forecasting. *Neural Computing and Applications*, 32(23): 17149–17167.

Lolli, F.; Gamberini, R.; Regattieri, A.; Balugani, E.; Gatos, T.; and Gucci, S. 2017. Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, 183: 116–128.

Mohammadipour, M. 2009. *Intermittent Demand Forecasting with Integer Autoregressive Moving Average Models*. Buckinghamshire New University, Brunel University.

Nystrup, P.; Lindström, E.; Pinson, P.; and Madsen, H. 2020. Temporal hierarchies with autocorrelation for load forecasting. *European Journal of Operational Research*, 280(3): 876–888.

Papoulis, A. 1962. *The Fourier Integral and Its Applications*. Classic Textbook Reissue Series. McGraw-Hill. ISBN 9780070484474.

Park, J.; Schoenberg, F. P.; Bertozzi, A. L.; and Brantingham, P. J. 2021. Investigating clustering and violence interruption in gang-related violent crime data using spatial–temporal point processes with covariates. *Journal of the American Statistical Association*, 116(536): 1674–1687.

Percival, D. B.; and Walden, A. T. 2000. *Wavelet methods for time series analysis*, volume 4. Cambridge university press.

Petropoulos, F.; Nikolopoulos, K.; Spithourakis, G. P.; and Assimakopoulos, V. 2013. Empirical heuristics for improving intermittent demand forecasting. *Industrial Management & Data Systems*, 113(5): 683–696.

Roberts, S.; Osborne, M.; Ebden, M.; Reece, S.; Gibson, N.; and Aigrain, S. 2013. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984): 20110550.

Salinas, D.; Bohlke-Schneider, M.; Callot, L.; Medico, R.; and Gasthaus, J. 2019. High-dimensional multivariate forecasting with low-rank Gaussian Copula Processes. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Salinas, D.; Flunkert, V.; Gasthaus, J.; and Januschowski, T. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, 36(3): 1181–1191.

Seeger, M. W.; Salinas, D.; and Flunkert, V. 2016. Bayesian Intermittent Demand Forecasting for Large Inventories. In Lee, D.; Sugiyama, M.; Luxburg, U.; Guyon, I.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Tyralis, H.; and Papacharalampous, G. 2024. A review of predictive uncertainty estimation with machine learning. *Artificial Intelligence Review*, 57(4): 94.

Wang, B.; Luo, X.; Zhang, F.; Yuan, B.; Bertozzi, A. L.; and Brantingham, P. J. 2018a. Graph-based deep modeling and real time forecasting of sparse spatio-temporal data. *arXiv preprint arXiv:1804.00684*. 4th Workshop on Mining and Learning from Time Series (MileTS), at KDD London, August 2018.

Wang, B.; Yin, P.; Bertozzi, A. L.; Brantingham, P. J.; Osher, S. J.; and Xin, J. 2019. Deep learning for real-time crime forecasting and its ternarization. *Chinese Annals of Mathematics, Series B*, 40(6): 949–966.

Wang, J.; Wang, Z.; Li, J.; and Wu, J. 2018b. Multilevel Wavelet Decomposition Network for Interpretable Time Series Analysis. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2437–2446. New York, NY, USA: Association for Computing Machinery. ISBN 9781450355520.

Welsh, B. C.; Zimmerman, G. M.; and Zane, S. N. 2018. The centrality of theory in modern day crime prevention: Developments, challenges, and opportunities. *Justice Quarterly*, 35(1): 139–161.

Zhang, G.; Patuwo, B. E.; and Hu, M. Y. 1998. Forecasting with Artificial Neural Networks: The State of the Art. *International Journal of Forecasting*, 14(1): 35–62.

Zhang, G. P.; Xia, Y.; and Xie, M. 2024. Intermittent demand forecasting with transformer neural networks. *Annals of Operations Research*, 339(1): 1051–1072.

Zhang, L.; Aggarwal, C.; and Qi, G.-J. 2017. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, 2141–2149. New York, NY, USA: Association for Computing Machinery.

Çerağ Pinçe; Turrini, L.; and Meissner, J. 2021. Intermittent demand forecasting for spare parts: A Critical review. *Omega*, 105: 102513.