



Article

Fractional Time-Varying Autoregressive Modeling: Parallel GAM and PINN Approaches to Dynamic Volatility Forecasting

Zhixuan Jia ¹ and Nan Rao ^{2,*} ¹ School of Information Management, Wuhan University, Wuhan 430072, China; 00034737@whu.edu.cn² Center for Financial Engineering, Soochow University, Suzhou 215006, China

* Correspondence: nan.rao@suda.edu.cn

Abstract

Time-series modeling plays a crucial role in analyzing and forecasting financial volatility. Classical approaches, such as the Autoregressive (AR) and Fractional Autoregressive (FAR) models, capture short-term linear dependencies and long-range correlations, respectively, but their reliance on fixed structures and stationarity assumptions limits their adaptability to evolving market dynamics. To overcome these limitations, this study introduces a Fractional Time-Varying Autoregressive (FTVAR) framework that allows model parameters to evolve smoothly over time, integrating long-memory effects with nonstationary behavior. The FTVAR process is examined through two complementary methods: a Generalized Additive Model (GAM) for interpretable estimation of time-varying coefficients, and a Physics-Informed Neural Network (PINN) that embeds system dynamics to enhance forecasting under complex conditions. Simulation studies demonstrate that the FTVAR model consistently outperforms conventional AR approaches, offering superior forecasting accuracy, robustness to nonstationarity, and a more comprehensive representation of evolving volatility structures. Empirical analyses on the S&P 500 and VIX indices further confirm the effectiveness and practical relevance of the proposed framework.

Keywords: financial volatility analysis; fractional time-varying autoregressive (FTVAR); physics-informed neural network (PINN); generalized additive model (GAM)



Academic Editor: Norbert Herencsar

Received: 20 October 2025

Revised: 13 November 2025

Accepted: 24 November 2025

Published: 26 November 2025

Citation: Jia, Z.; Rao, N. Fractional Time-Varying Autoregressive Modeling: Parallel GAM and PINN Approaches to Dynamic Volatility Forecasting. *Fractal Fract.* **2025**, *9*, 772. <https://doi.org/10.3390/fractalfract9120772>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Time-series analysis has become a cornerstone of modern data science, providing essential tools for understanding, modeling, and forecasting data that evolves over time. Unlike cross-sectional data, which assume independent observations, a time series inherently preserves temporal ordering, making it crucial for capturing dependencies, trends, cycles, and persistence. This temporal structure is particularly important in finance, where accurately modeling and forecasting volatility, returns, and risk directly inform portfolio management, derivative pricing, and risk assessment strategies [1,2]. Capturing volatility dynamics is challenging due to clustering, long memory, and abrupt market shifts, which traditional autoregressive (AR) models struggle to represent, as they are limited to short-term dependencies with rapidly decaying autocorrelations, whereas many financial series exhibit slowly decaying correlations indicative of long-range dependence. Fractional time-series models, such as the fractional autoregressive (FAR) and fractional autoregressive moving average (FARMA) models, address this limitation by introducing fractional differencing, $(1 - L)^d$, where L is the lag operator and d is a fractional parameter. This

enables the process to exhibit intermediate behavior between stationarity and nonstationarity, effectively capturing long memory and making fractional approaches particularly valuable for improving volatility forecasting and supporting informed decision-making under uncertainty [3,4].

Unfortunately, like traditional AR models, FAR and FARMA models often face practical limitations because they assume a fixed underlying structure, which restricts their ability to capture evolving dynamics, time-varying dependencies, and abrupt shifts commonly observed in financial volatility. This rigidity can reduce forecasting accuracy and hinder their capacity to reflect market irregularities or regime changes. To address these shortcomings, we extend the classical FAR framework by incorporating time-varying coefficients, resulting in a novel fractional time-varying autoregressive (FTVAR) process, as introduced in Equation (1), which flexibly adapts to both persistent long-memory effects and dynamic structural changes in financial time series.

$$X_t = \sum_{k=1}^R \alpha_k(t) X_{t-k} + \sum_{s=R+1}^{\infty} \Pi(s) X_{t-s} + \epsilon_t, \quad (1)$$

where the fractional weights $\Pi(s)$ are defined as Equation (2).

$$\Pi(s) = \frac{\Gamma(s-d)}{\Gamma(-d)\Gamma(s+1)}, \quad (2)$$

and $\Gamma(\cdot)$ denotes the Gamma function. To model financial volatility using the FTVAR framework, we extend the traditional generalized additive model (GAM) to incorporate the FTVAR process, allowing it to capture both time-varying dynamics and long-range dependencies induced by fractional differencing. By representing time-varying coefficients through flexible, smooth basis functions, GAMs offer interpretable and structured estimation of nonlinear temporal patterns while simultaneously capturing gradual shifts and persistent effects in financial volatility, balancing flexibility with robustness and making them well-suited for processes exhibiting both short-term fluctuations and long-memory behaviors. Additionally, we incorporate the FTVAR process as a structural constraint within Physics-Informed Neural Networks (PINNs), enabling the model to leverage prior knowledge of fractional, time-varying dynamics while retaining the flexibility and expressive power of deep neural networks. This integration enhances generalization and robustness to noise by directly embedding governing constraints into the learning process, guiding the model to respect the underlying dynamics and making PINNs particularly well-suited compared to conventional neural networks. Moreover, the flexible neural network backbone allows PINNs to capture nonlinear, time-varying relationships that are challenging to model with purely data-driven approaches, including standard deep learning architectures. These two distinct methodologies provide complementary perspectives: GAM offers interpretability and smooth functional estimation, while PINNs provide flexibility and data-driven modeling capacity for complex financial volatility patterns. The primary contributions of this research are summarized as follows:

- Building on the classical FAR framework, we introduce time-varying parameters into the autoregressive component. This extension accommodates nonstationarity, enhancing flexibility in capturing the evolving dynamics of market volatility.
- For modeling the FTVAR process, we adapt the classical GAM framework to incorporate fractional components, and apply a similar extension within the PINNs setting. We evaluate and discuss the strengths and limitations of both approaches in the context of financial volatility modeling.

- We provide an open-source implementation of both GAM- and PINN-based frameworks, developed using TensorFlow 2.18 for neural network training and SciPy for numerical optimization, to facilitate reproducibility and future research.

The structure of the paper is organized as follows: Section 3 introduces the proposed FTVAR process along with the foundational concepts of the GAM- and PINNs-based approaches. Section 4 reports simulation results under two distinct scenarios, linear and periodic time-varying settings. Section 5 validates the effectiveness of the proposed methods using real-world financial time-series data from the S&P 500 and the VIX index.

2. Related Work

Currently, dynamic modeling, particularly via autoregressive frameworks, is a fundamental tool for analyzing and forecasting financial market volatility, as it systematically captures temporal dependencies and serial correlations in financial time series while providing a structured framework to reveal complex patterns and evolving dynamics over time. Traditional approaches include Structural Vector Autoregressions [5], which facilitate the identification of contemporaneous interactions among multiple financial variables; Autoregressive Conditional Heteroskedasticity models [6], which are effective in modeling time-varying volatility; as well as Nonlinear Autoregressive Distributed Lag [7,8] and Quantile Autoregressive Distributed Lag models [9], which capture asymmetric and distribution-specific dynamics in volatility.

Besides these classical approaches, advances in both theoretical and practical probabilistic modeling, including fractional Brownian motion [10,11] and probabilistic machine learning techniques such as autoencoding and diffusion models [12,13], have enhanced volatility analysis by enabling a more nuanced representation of uncertainty and stochastic dynamics. Moreover, the rapid advancement of artificial intelligence has driven the development of data-driven methods for volatility modeling. Machine learning techniques [14] and deep learning architectures, such as Long Short-Term Memory (LSTM) networks [15–17] and traditional feedforward neural networks [18], have been increasingly applied to uncover complex, nonlinear patterns in financial time series. However, while these models often achieve high predictive accuracy, they frequently suffer from a lack of interpretability and limited ability to incorporate domain-specific constraints, which reduces their explanatory power in financial applications.

To address this limitation, Physics-Informed Neural Networks (PINNs) were introduced as a hybrid framework that integrates physical or mathematical constraints with data-driven learning. This framework, first proposed in 2019 [19], provides a mesh-free and data-efficient methodology that integrates governing equations directly into the loss function through automatic differentiation. The design of PINNs enforces both data fidelity and physical consistency, making PINNs particularly effective for solving ordinary and partial differential equations (ODEs/PDEs), especially in scenarios with sparse or noisy observations. Since their introduction, PINNs have been successfully applied to diverse domains, including wave equations [20,21], the Schrödinger equation [22], and the Huxley equation [23]. More recently, beyond classical ODE/PDE problems, PINNs have also been extended to fields with well-defined mathematical models, such as cybersecurity (e.g., false data injection detection [24]), wireless communication (e.g., WiFi signal propagation [25]), and finance, including option pricing and volatility analysis [26–29].

Then, in the context of classical time-varying autoregressive (TVAR) processes, where no fractional component is present, estimation is typically carried out using either the Generalized Additive Model (GAM) framework or kernel-smoothing techniques. GAMs approximate time-varying coefficients via flexible basis functions, while kernel-smoothing methods employ localized weighting schemes to track gradual structural changes in the

process. Both approaches have been widely applied across disciplines, including the influence analysis and evaluation of COVID-19 [30–32], research in psychology and behavioral sciences [33,34], Electroencephalography signal processing [35–37], marine science [38], and the study of financial and commodity market volatility [39–41]. These applications highlight the adaptability of TVAR models in capturing dynamic structural changes and their relevance across theoretical and applied domains. Finally, beyond the time-varying framework, several extensions of the fractional autoregressive (FAR) process have also been explored, including the seasonal FAR [42], nonlinear FAR [43], periodic FAR [44], and FARMA models [45,46], reflecting the growing interest in fractional dynamics and long-memory structures in time-series analysis.

Despite these advances, a critical gap remains in extending time-varying autoregressive frameworks to fractional processes, which are essential for modeling long-memory and persistent behaviors commonly observed in financial volatility. Existing approaches struggle to balance data efficiency, interpretability, and flexibility when incorporating fractional dynamics, highlighting the need for novel methodologies to bridge this gap.

3. Proposed Methods for Fractional Time-Varying Process Modeling

Recall that the classical AR process is expressed as Equation (3).

$$X_t = \sum_{i=1}^R \alpha_i X_{t-i} + \epsilon_t, \quad (3)$$

where X_t denotes the observed one-dimensional time series at time t , α_i represents the fixed autoregressive coefficients that capture dynamic dependencies at lag i , and ϵ_t is a noise term, typically assumed to be white noise. Next, to incorporate long-memory effects, we follow the fractional differencing formulation proposed by [47], assuming a zero-order moving average polynomial. This leads to the fractional autoregressive (FAR) model defined in Equation (4):

$$\Phi(L)(1-L)^d X_t = \epsilon_t, \quad (4)$$

where

$$\Phi(L) = 1 - \phi_1 L - \dots - \phi_R L^R,$$

$d \in (-0.5, 0.5)$ denotes the fractional differencing parameter, and L is the lag operator. The operator $(1-L)^d$, known as the fractional differencing operator, generalizes the conventional integer differencing used in ARMA and ARIMA models. This generalization allows the FAR model to capture long-range dependence that traditional short-memory processes cannot represent. The fractional differencing operator can be expanded using the generalized binomial theorem, as shown in Equation (5), providing a numerically tractable representation that retains the long-memory property.

$$(1-L)^d = \sum_{i=0}^{\infty} (-1)^i \binom{d}{i} L^i = \sum_{i=0}^{\infty} \Pi(i) L^i, \quad (5)$$

$$\Pi(i) = (-1)^i \binom{d}{i} = \frac{\Gamma(i-d)}{\Gamma(-d)\Gamma(i+1)},$$

where $\Gamma(\cdot)$ denote Gamma function. Substituting Equation (5) back into Equation (4) and rearranging X_t , the FAR model can be rewritten in infinite-sum form Equation (6):

$$(1 - \phi_1 L - \dots - \phi_R L^R) \sum_{k=0}^{\infty} \Pi(k) X_{t-k} = \epsilon_t. \quad (6)$$

Next, define the variable $Y_t := (1 - L)^d X_t$, so that Equation (4) can be rewritten as:

$$\Phi(L)Y_t = \epsilon_t. \quad (7)$$

By evaluating $\Phi(L)$ and solving for Y_t in Equation (7), we obtain:

$$Y_t = \sum_{i=1}^R \phi_i Y_{t-i} + \epsilon_t. \quad (8)$$

Substituting X_t back into Equation (8) and isolating X_t yields:

$$X_t = \sum_{i=1}^R \phi_i \sum_{k=0}^{\infty} \Pi(k) X_{t-i-k} - \sum_{k=1}^{\infty} \Pi(k) X_{t-k} + \epsilon_t. \quad (9)$$

Finally, by separating the infinite sum, the FAR process can be represented in a finite form with an appended tail:

$$X_t = \sum_{k=1}^R \alpha_k X_{t-k} + \sum_{s=R+1}^{\infty} \beta_s X_{t-s} + \epsilon_t, \quad (10)$$

where the first sum captures the short-term dynamics via fixed autoregressive coefficients, while the second summation represents the long-memory structure induced by fractional differencing. Additional theoretical properties and asymptotic behavior of this formulation are further discussed in [47–49]. In [47,50], they discussed the stationarity of fractional models. When $d < 0.5$, the series remains weakly stationary when all the roots of $\Phi(z) = 0$ lie outside the unit circle, whereas values of $0.5 \leq d < 1$ produce nonstationary but mean-reverting dynamics.

To further accommodate nonstationarities and evolving dynamics, the time-varying coefficients were introduced into Equation (10), which allows the autoregressive structure to evolve over time. This leads to the fractional time-varying autoregressive (FTVAR) process Equation (11).

$$X_t = \underbrace{\sum_{k=1}^R \alpha_k(t) X_{t-k}}_{\text{Time varying components}} + \underbrace{\sum_{s=R+1}^{\infty} \beta_s X_{t-s}}_{\text{Fractional components}} + \epsilon_t, \quad (11)$$

where $\alpha_k(t)$ are time-varying coefficients capturing evolving dependencies weights, and β_s are the fractional weights. Next, based on Equation (9), we can derive the explicit expression for β_s as:

$$\beta_s = \sum_{j=1}^R \phi_j \Pi(s+j) - \Pi(s).$$

From Equation (5), we know that:

$$\Pi(i+1) = \left(\frac{i-d}{i+1} \right) \Pi(i).$$

Define $I = (i-d)/(i+1)$, where $0 < I < 1$. Substituting this relation yields:

$$\beta_s = \left(\phi_1 I + \phi_2 I^2 + \cdots + \phi_R I^R - 1 \right) \Pi(s). \quad (12)$$

Then, we assume that the polynomial term $\phi_1 I + \phi_2 I^2 + \dots + \phi_R I^R - 1$ remains within the interval $(0, 1)$ for $0 < I < 1$. Under this assumption, the equation in Equation (11) can be approximated as:

$$X_t \approx \sum_{k=1}^R \alpha_k(t) X_{t-k} + \sum_{s=R+1}^{\infty} \Pi(s) X_{t-s} + \epsilon_t, \quad (13)$$

which serves as the working formulation in our modeling framework. Several representative FTVAR sample paths based on this approximation are shown in Figure 1.

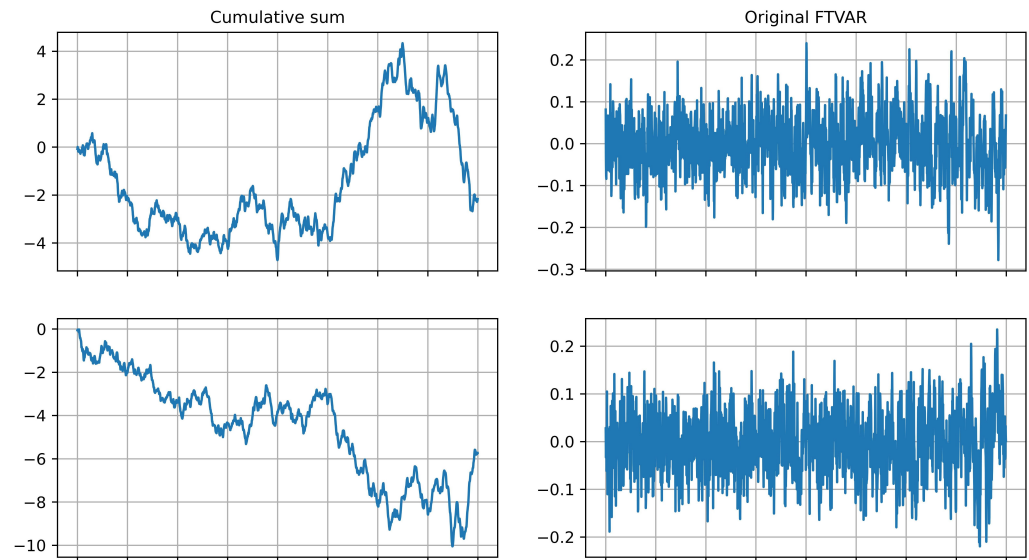


Figure 1. Two sample processes of FTVAR(1) where $d = 0.2$, $\epsilon_t \sim N(0, 0.05)$, and time-varying coefficient $\alpha(t) = 0.3 + t/T$.

Next, we examine the stationarity of the FTVAR model. Following the proof of Theorem 2 in [47], we first rewrite the process as:

$$X_t = \frac{(1 - L)^{-d}}{\Phi(L)} \epsilon_t,$$

where in our model, the first R coefficients are time-dependent by assumption. The fractional differencing operator $(1 - L)^{-d}$ admits a convergent power series for $|L| \leq 1$ when $d \leq 0.5$, ensuring it is well-defined. However, due to the time-varying nature of the time-varying coefficients, $\Phi(L)^{-1}$ cannot be expressed as a simple power series, and the operator cannot be inverted analytically, leading to global nonstationarity. But locally, where the time-varying terms can be approximated as constants, with the assumption that all the roots of the equation $\Phi(L) = 0$ lie outside the unit circle, $\Phi(L)^{-1}$ admits a convergent power series representation. Under these conditions, the series expansion of $(1 - L)^{-d}/\Phi(L)$ converges for all $|L| \leq 1$, implying that X_t is locally stationary when $d < 0.5$. For $d > 0.5$, the locally spectral density of X_t satisfies $s(\omega) \sim \omega^{-2d}$ [47], indicating locally nonstationarity.

In order to model and make predictions based on the FTVAR process, we consider two methodological approaches: First, we employ Generalized Additive Model (GAM)-based techniques, which have been widely used in the estimation of time-varying autoregressive processes and offer a flexible framework for modeling smooth temporal variations in the coefficient $\alpha_k(t)$. Second, we introduce a novel Physics-Informed Neural Network (PINN)-based approach, which embeds the governing equations of the FTVAR model directly into

the training process, thereby combining data-driven learning with physical constraints to more effectively capture both short-term dynamics and long-range dependencies.

3.1. Generalized Additive Based Approach

The Generalized Additive Model (GAM) is widely regarded as a flexible extension of classical linear models, designed to accommodate nonlinear patterns in data while retaining interpretability. Unlike linear models, which assume that predictors exert fixed linear effects on the response, GAMs replace these fixed coefficients with smooth, non-parametric functions known as smoothers. The key assumption of this approach is that each predictor's contribution to the response can be represented by a smooth function, thereby relaxing the restrictive linearity constraint and enabling the model to capture complex, nonlinear relationships without the need to prespecify a functional form. In practice, the smooth functions in a GAM are constructed using basis functions, such as splines, radial basis functions, or other suitable families, where the general structure of a GAM with n basis functions can be expressed as Equation (14):

$$g(t) = \sum_{i=1}^n \zeta_i f_i(t), \quad (14)$$

where f_i denotes a chosen smooth basis function, and ζ_i is the corresponding linear coefficient in the GAM framework. In this research, we estimate the time-varying coefficients of the FTV-AR process by assuming the existence of r nonzero smooth functions, $g_1(t), \dots, g_r(t)$, each of which captures the temporal evolution of a corresponding lagged effect. Within the GAM framework, these coefficients can be represented as a linear combination of smooth basis functions, expressed as Equation (15):

$$a_t = \sum_{i=1}^n \alpha_i f_i(t). \quad (15)$$

Next, we consider the estimation of the fractional differencing parameter; estimating all parameters in the FTV-AR model can be formulated as the following optimization problem Equation (16).

$$L_{GAM}(d, \alpha) = \underset{0 < d < 1, \alpha}{\operatorname{argmin}} : \left\| X_r - \sum_{i=1}^R \alpha_i X_{r-i} - \sum_{j=1}^M \Pi(j, d) X_{r-j} \right\|_2^2. \quad (16)$$

where d is the fractional differencing parameter, α_i are the autoregressive coefficients, and $\Pi(j, d)$ denotes the fractional weights. This formulation seeks the parameter values that minimize the error between the observed series and its reconstructed form.

A key consideration in implementing the GAM framework is the selection of basis functions, as overparameterization can facilitate benign overfitting that enhances model flexibility, whereas excessive flexibility may lead to unstable predictions. Common spline bases offer varying degrees of smoothness and adaptability but are sensitive to the number and placement of knots, potentially causing underfitting or overfitting, whereas alternative basis functions can be employed depending on the target function and the desired trade-offs between interpretability, smoothness, and computational efficiency. In this study, we adopt a combination of spline bases and other selected functions, including Gaussian and tanh, to capture both smooth and nonlinear patterns in the time-varying coefficients, with a subset of the basis functions illustrated in Figure 2.

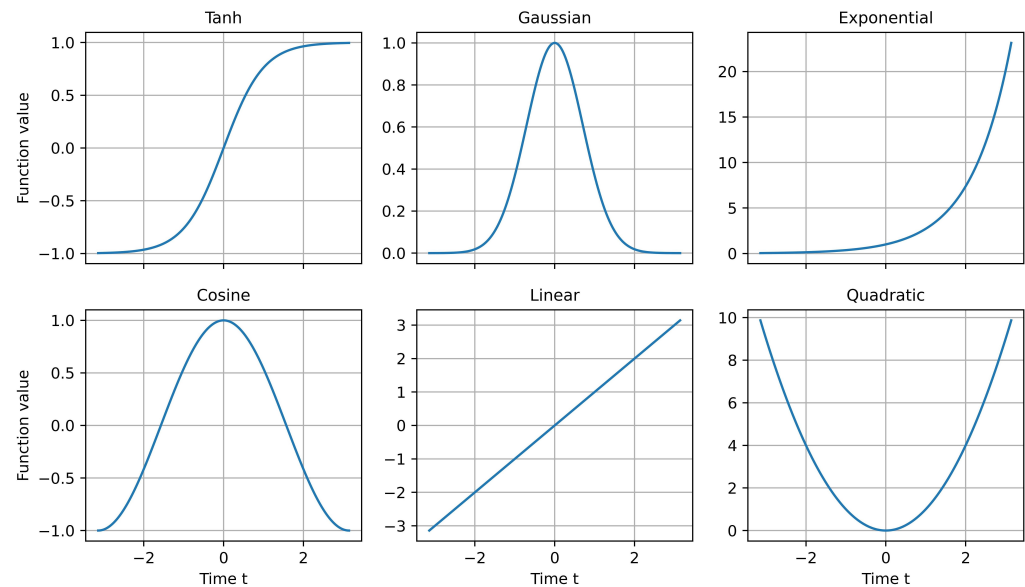


Figure 2. Six basis functions were used in the simulation over $t \in [-\pi, \pi]$.

Finally, after selecting the appropriate basis functions, several strategies are available to solve the optimization problem in Equation (16). Unfortunately, unlike classical GAMs used for TVAR simulations, this problem cannot be reformulated as a linear regression due to the presence of fractional weights dependent on d , making gradient-based optimization a more suitable approach. Currently, there are several methods available, including naive gradient descent, conjugate gradient, and the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, which can handle both constrained and unconstrained optimization. Hence, to address the constraints in Equation (16) while maintaining computational efficiency, we employ a Lagrangian gradient-based solver from the Python 3.9.19 SciPy package, specifically the trust-constr algorithm. The workflow of the GAM-based approach is presented in Figure 3.

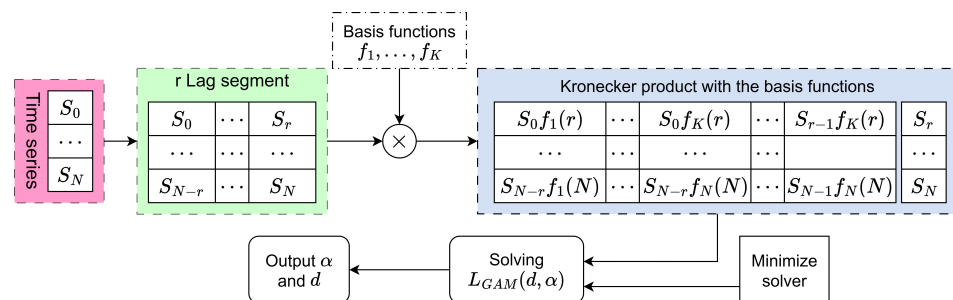


Figure 3. The working flow of GAM-based approach.

3.2. The Physics-Informed Neural Networks Based Approach

Physics-Informed Neural Networks (PINNs) extend conventional neural architectures by embedding domain-specific knowledge directly into the training procedure, thereby overcoming one of the central limitations of black-box learning: the lack of interpretability and adherence to known physical principles. Rather than relying solely on data-driven empirical fitting, PINNs augment the loss function with additional terms that enforce consistency with governing equations, such as partial differential equations (PDEs), conservation laws, or other structural invariants. In practice, the total loss function is constructed to reflect multiple sources of physical information, typically combining the residuals of the PDE, initial conditions, and boundary conditions, which are expressed as:

$$L_{total} = L_{PDSs} + L_{initial} + L_{boundary},$$

where each component quantifies the discrepancy between the neural network approximation and the corresponding physical constraint. Consequently, training a PINN can be formulated as a constrained optimization problem, in which the network parameters are iteratively updated to minimize the composite loss through gradient-based optimization methods. Upon convergence, the trained model provides a mesh-free and continuous surrogate of the underlying physical solution, enabling efficient evaluation at arbitrary points within the domain by directly supplying the independent variables.

In the case of the FTVAR model, the structural constraints cannot be directly formulated as physical laws, which limits their compatibility with the standard PINN framework. This challenge arises from the time-varying nature of the model coefficients, which prevents the system from being represented by a single, time-invariant differential or algebraic equation. By contrast, traditional deep learning approaches to time-series analysis directly estimate or forecast the observed trajectories $X^{(t)}$, but despite their forecasting ability, they often fail to disentangle intrinsic temporal dynamics from noise or exogenous effects and tend to overlook the autoregressive structure that underpins interpretable and stable representations of dynamical systems. Hence, for time-varying processes, a more principled modeling strategy is to focus on learning the evolution of time-dependent coefficients rather than directly fitting the observed trajectories $X^{(t)}$, in line with prior studies such as [51], which emphasize classical PINNs-based TVAR modeling. From this perspective, under the assumption that d is constant while the coefficients α_i evolve over time, the total loss functions are formulated as presented in Equation (17).

$$L_{FTVAR} = \left\| X_r - \sum_{i=1}^R \alpha_i X_{r-i} - \sum_{j=1}^M \Pi(j, \bar{d}) X_{r-j} \right\|_2^2 + \frac{1}{N-1} \sum_{i=0}^N (d - \bar{d})^2, \quad (17)$$

where $\bar{d}$ denotes the mean estimate of d across all time steps t . Considering the long-range temporal dependencies inherent in the FTVAR process, the Long Short-Term Memory (LSTM) network represents a suitable modeling choice due to its strong capability to capture dynamic and time-dependent patterns. To further enhance the model's capacity for representing static and nonlinear relationships, a hybrid architecture integrating both a sequential neural network (NN) component and an LSTM module is proposed. This design allows the NN component to learn deterministic and nonlinear structures, while the LSTM focuses on capturing the temporal dynamics and stochastic variations. Accordingly, the proposed PINN-based framework is organized around two core components:

- A sequence of k layers, each equipped with an activation function $G = [g_0, \dots, g_k]$, trainable input weight matrices $W = [W_0, \dots, W_k]$, and intercept vectors $b = [b_0, \dots, b_k]$;
- Hidden-state value $H = [h_0, \dots, h_l]$ and weight matrices $U = [U_0, \dots, U_l]$ for capturing recurrent dynamics in the LSTM component.

Formally, the operation of the LSTM, which models temporal dependencies across lagged inputs $y^{(t-1)}, \dots, y^{(t-r)}$, is expressed as follows:

$$\begin{aligned} f_t &= \sigma(Wx_t + Uh_{t-1} + b); \\ i_t &= \sigma(Wx_t + Uh_{t-1} + b); \\ o_t &= \sigma(Wx_t + Uh_{t-1} + b); \\ c_t &= f_t \times c_{t-1} + i_t \times \tanh(Wx_t + Uh_{t-1} + b); \\ h_t &= o_t \times \tanh(c_t). \end{aligned}$$

In the output layer, the output value was separated into two parts: time-varying coefficients α_i and fractional parameter d , where an extra layer was added into the fractional parameter due to its constraint from 0 to 1, as illustrated in Equations (18) and (19).

$$\hat{\alpha}_0, \dots, \hat{\alpha}_r, d_0 = W_k h_k + b_k; \quad (18)$$

$$\hat{d} = \frac{1}{1 + e^{d_0}}; \quad (19)$$

where $W_k \in \mathbb{R}^{d_{k-1} \times (r+2)}$, and $b_k \in \mathbb{R}^{(r+2) \times 1}$. Accordingly, the proposed LSTM-based TVAR-PINNs can be interpreted as a solution to the optimization problem defined in Equation (20).

$$W, U, b = \underset{W, U, b}{\operatorname{argmin}} : L_{FTVAR}(y, W, U, b), \quad (20)$$

where $L_{FTVAR}(y, W, U, b)$ represents the loss function that integrates the time-varying autoregressive components into the neural network framework. The overall architecture of this approach is illustrated in Figure 4.

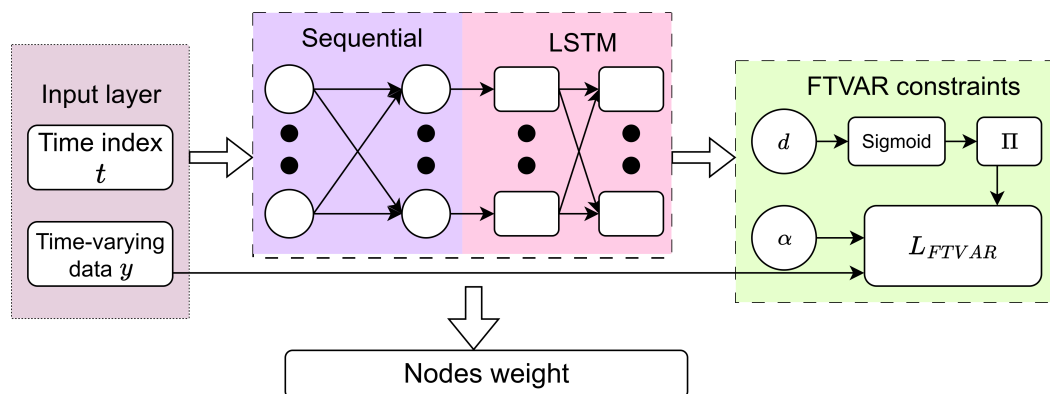


Figure 4. The main structure for the PINNs-based fractional time-varying autoregressive modeling approach.

Next, to solve the optimization problem defined in Equation (20), various gradient-based algorithms can be employed, including the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, the conjugate gradient method, and Adaptive Moment Estimation (Adam). In this study, we adopt the Adam optimizer due to its complementary strengths in handling large-scale optimization tasks. Adam combines adaptive learning rates with momentum-based updates, enabling efficient and stable convergence even in high-dimensional parameter spaces, which makes it particularly well-suited for models with complex, smooth, and differentiable loss landscapes. Additionally, when paired with methods that support bound constraints, Adam facilitates accurate and reliable estimation of time-varying coefficients, while maintaining computational efficiency and reducing memory overhead—features that are critical for training PINN-based FTVAR models on large datasets.

Finally, the prediction mechanism in our proposed FTVAR framework fundamentally differs from conventional deep learning approaches. Instead of directly forecasting future time-series values, the model estimates the underlying time-varying parameters: the autoregressive coefficients $\alpha(t)$, and the fractional differencing parameter d . Predicting the value at time $T + h$ in an FTVAR model with r lags involves the following steps. First, construct the lagged input vector from observed data spanning $T - r$ to $T - 1$ to generate the TVAR component of the prediction. Second, use the estimated fractional parameter $\hat{d}$ and the observed data from $T - M$ to $T - r$ to compute the fractional differenced component.

Third, combine these two components to obtain the final prediction $\hat{X}_t$. This procedure is then iteratively repeated until reaching the forecast horizon $T + h$.

3.3. Performance Evaluation

To evaluate the performance of the proposed methods, three complementary measurement criteria were employed to capture both prediction accuracy and the ability to reconstruct the underlying temporal dependence structure of the data.

- **Mean absolute error (MAE):** To assess the accuracy of the predictions, we consider the Mean Absolute Error (MAE). MAE provides a straightforward and interpretable metric by averaging the absolute deviations between the predicted values $\hat{y}_i$ and the true values y_i . The MAE is computed over a forecast horizon of length H as follows Equation (21)

$$MAE(\hat{y}, y) = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|. \quad (21)$$

- **Autocorrelation Function (ACF):** While MAE captures pointwise predictive accuracy, it does not measure how well the model preserves the temporal correlation structure of the data. To evaluate reconstruction performance, we employ the ACF, which measures the correlation of a time series with its own lagged values and serves as an essential tool for characterizing the dynamics of time-dependent processes. The ACF at lag K is defined as Equation (22).

$$\rho(K) = \frac{Cov(X_t, X_{t-K})}{\sqrt{VAR(X_t)VAR(X_{t-K})}}. \quad (22)$$

In our evaluation, the ACF is first computed separately for both the original and reconstructed time series across a range of lags, and the MAE between the resulting ACF sequences is subsequently calculated. A lower ACF-based error indicates that the reconstructed series retains more of the temporal dependency present in the original data.

- **Shapiro–Wilk test:** A statistical test used to evaluate whether a sample comes from a normally distributed population, with the null hypothesis H_0 stating that the data is normally distributed. The test statistic W is defined as Equation (23):

$$W = \frac{(\sum_{i=1}^n a_i x_i)^2}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad (23)$$

where x_i are the ordered sample values from smallest to largest, and a_i are constants derived from the expected values of order statistics of a standard normal distribution. Values of W close to 1 indicate approximate normality, while smaller values suggest deviations, with the associated p value formally testing H_0 , such that $p > 0.05$ implies that H_0 is not rejected and data are consistent with a normal distribution.

Together, MAE, ACF, and the Shapiro–Wilk test form a comprehensive evaluation framework: MAE measures forecast accuracy in the time domain, while ACF and the Shapiro–Wilk test analyze residuals, with ACF assessing temporal dependencies and structural patterns and the Shapiro–Wilk test evaluating normality.

4. Simulation

In practical implementations, the infinite summation of fractional components must be truncated to a finite lag $M > R$, with M selected to be sufficiently large to preserve the long-memory characteristics while maintaining computational feasibility. Accordingly, the

process in Equation (13) can be approximated by the following finite-lag representation Equation (24).

$$X_t \approx \sum_{k=1}^R \alpha_k(t) X_{t-k} + \sum_{s=R+1}^M \Pi_s X_{t-s} + \epsilon_t, \quad (24)$$

while $\alpha_k(t) \in \mathbb{R}$ denotes the time-varying coefficient for lag k at time t , and noise term $\epsilon \sim N(0, \sigma)$ is assumed to follow a Gaussian distribution with zero mean and variance $\sigma \in \mathbb{R}$. Here, the truncation at lag M introduces an approximation error due to the neglected tail of the infinite fractional summation. To quantify this error, we define the truncation error term as Equation (25).

$$\Delta_t := X_t - X_t^{(M)} = \sum_{j=M+1}^{\infty} \Pi_s X_{t-s}. \quad (25)$$

Assume that there exists a finite constant B such that $|X_t| \leq B$ for all t . Substituting this bound into Equation (25) yields

$$|\Delta_t| \leq B \sum_{s=M+1}^{\infty} |\Pi_s|.$$

From the generalized binomial expansion of the fractional differencing operator and the asymptotic approximation of the ratio of Gamma functions, for large j , the coefficients $\Pi(j)$ satisfy

$$|\Pi(s)| = \left| \frac{\Gamma(s-d)}{\Gamma(-d)\Gamma(s+1)} \right| \sim \frac{s^{-(1+d)}}{\Gamma(-d)}. \quad (26)$$

Putting Equation (26) into Equation (25) gives the asymptotic upper bound of the truncation error:

$$|\Delta_t| \leq B \sum_{s=M+1}^{\infty} |\Pi(s)| \sim O(M^{-(1+d)}), \text{ as } M \rightarrow \infty. \quad (27)$$

Therefore, the truncation error decays polynomially with order $1+d$, implying that larger M values improve approximation accuracy at a sub-exponential rate. Finally, to determine an appropriate truncation lag M for a target tolerance k , which requires $|\Delta_t| \leq k$. Using the asymptotic form of $\Pi(s)$, we have

$$\begin{aligned} |\Delta_t| &\leq B \sum_{s=M+1}^{\infty} |\Pi(s)| \\ &\leq B \sum_{s=M+1}^{\infty} \frac{s^{-(1+d)}}{|\Gamma(-d)|} \\ &\leq \frac{B}{|\Gamma(-d)|} \int_M^{\infty} s^{-(1+d)} dj = \frac{BM^{-d}}{|\Gamma(-d)|d}. \end{aligned}$$

Hence, to ensure that the truncation error remains below the specified tolerance k , the M should satisfy:

$$M \geq \left[\frac{B}{|\Gamma(-d)|dk} \right]^{1/d}. \quad (28)$$

From Equation (28), we observe a critical dependence of the truncation lag M on the fractional differencing parameter d . As d approaches 1, the fractional coefficients decay faster, meaning the influence of distant past observations becomes negligible; hence, a smaller truncation lag is sufficient to achieve a desired approximation accuracy. In contrast, as d is close to 0^+ , the term d^{-d} increases sharply, implying that M should be

significantly larger to achieve the same approximation accuracy. Specifically, when d becomes sufficiently close to 0, the fractional component effectively vanishes, and the process reduces to a standard TVAR process, while the asymptotic bound in Equation (28) diverges, reflecting the progressively slower decay of fractional weights.

In summary, the GAM framework models time-varying coefficients via basis functions from historical data, whereas the PINNs approach learns them directly from temporal inputs for multi-step forecasting over horizon H . All analyses were implemented in Python 3.9.19 and executed on a workstation with an NVIDIA RTX 3050 Ti GPU and AMD Ryzen 7 5800H CPU, using GPU acceleration to enhance training efficiency.

4.1. Simulation Preparation

In the simulation study, we examine multiple configurations defined by the fractional differencing parameter $d \in \{0.2, 0.8\}$, series length $N \in \{750, 1000, 1500\}$ with normalized time $t \in [0, 1]$, Gaussian noise with zero mean and variance $\sigma = 0.1$, and truncation lags of $M = 500$ for $d = 0.8$ and $M = 10,000$ for $d = 0.2$. Each setting is characterized by the following time-varying coefficient functions:

- Scenario 1: A univariate time series with lag order $r = 1$ and an initial value $y_0 = 0$. The coefficient is initialized by a constant scalar $a_0 \in [0, 1]$, $c \in [-1, 1]$ and evolves linearly:

$$f(t) = a_0 + ct.$$

- Scenario 2: The same setting for lag order r and initial value y_0 , but with a periodic coefficient. The coefficient is initialized by a constant scalar $a_0 \in [0.3, 0.8]$, $c \in [0, 1]$ and evolves as:

$$f(t) = a_0 + c \cos(2\pi t).$$

- Scenario 3: The same settings are used for the lag order r and initial value y_0 , but with a discrete coefficient consisting of two constants, a_1 and a_2 , defined for a given time m as follows:

$$f(t) = \begin{cases} a_1, & 0 < t \leq m; \\ a_2, & m < t \leq T. \end{cases}$$

For the GMA-based methods, optimization was performed using the trust-constr algorithm from SciPy, which implements a constrained optimization framework based on Lagrangian gradient descent, while the PINNs-based approach employed the Adam optimizer. Prediction performance was evaluated using the mean absolute error (MAE) for one-step and five-step forecasts, consistent with established practice in financial time-series analysis, whereas reconstruction performance was assessed using the ACF and the KS test.

4.2. Simulation Results

In this study, flexibility in the GAM-based methods was achieved by employing a diverse set of basis functions, including linear and quadratic terms for polynomial trends, hyperbolic tangent and exponential functions for saturation effects and exponential growth, Gaussian functions for localized smooth approximations, and cosine functions for periodic or oscillatory patterns. This combination enhances the expressive capacity of the models, enabling them to approximate a broad range of functional forms in the estimation of time-varying coefficients. For the PINNs-based approach, we employed a hybrid architecture that integrates a dense sequential network with LSTM components. The model consists of three hidden layers with 16, 32, and 16 neurons, respectively, uses the tanh and sigmoid activation functions, and is trained for up to 1200 iterations.

For each scenario, one-step-ahead predictions were made using 50 randomly generated time series of length 1000 with time t normalized from 0 to 1, and model performance was

evaluated across these steps using MAE, ACF, and the Shapiro–Wilk test. To provide a comprehensive comparison, three benchmark models were included: (1) a traditional AR model; (2) a hybrid ARFIMA-GARCH model (denoted as “ARF-GAR”), where ARFIMA estimates the conditional mean and GARCH models the stochastic volatility; and (3) a pure LSTM network composed of three LSTM layers with 32 nodes each and a tanh activation function.

All models were estimated with lag orders $r = 1, 5$, and the results for Scenarios 1–2 with $d = 0.2$ and $d = 0.8$ are presented in Tables 1–6. To compute the ACF and Shapiro–Wilk test, we first obtain the one-step predictions ($\hat{X}_t$), and calculate the residuals ($R_t = X_t - \hat{X}_t$), then apply the ACF and the Shapiro–Wilk test to the residuals, assessing their conformity with standard Gaussian noise, while also reporting the MAE of the ACF as a quantitative measure of goodness-of-fit. After that, a sample of representative one-step prediction results for all methods with lag order $r = 5$ is shown in Figure 5. Finally, to demonstrate the convergence behavior of the proposed PINNs-based approach, the error convergence rates for three scenarios with $d = 0.8$ and lag order $r = 5$ are presented in Figure 6.

Table 1. The summary statistics of absolute errors across 50 random samples for Scenario 1, where $a_0 = 1$, $c = -1$, and $d = 0.2$.

Method	MAE	ACF	Shapiro
PINN(1)	4.11×10^{-2}	3.88×10^{-2}	2.99×10^{-1}
PINN(5)	3.86×10^{-2}	3.18×10^{-2}	1.52×10^{-1}
GAM(1)	3.97×10^{-2}	3.59×10^{-2}	3.31×10^{-1}
GAM(5)	3.88×10^{-2}	3.21×10^{-2}	1.09×10^{-1}
AR(1)	4.28×10^{-2}	3.68×10^{-2}	2.36×10^{-1}
AR(5)	4.15×10^{-2}	3.78×10^{-2}	1.22×10^{-1}
ARF-GAR(1)	5.85×10^{-2}	3.75×10^{-2}	3.04×10^{-1}
ARF-GAR(5)	5.21×10^{-2}	3.54×10^{-2}	4.34×10^{-1}
LSTM	4.18×10^{-2}	3.66×10^{-2}	3.38×10^{-1}

Table 2. The summary statistics of absolute errors across 50 random samples for Scenario 1, where $a_0 = 1$, $c = -1$, and $d = 0.8$.

Method	MAE	ACF	Shapiro
PINN(1)	4.62×10^{-2}	4.44×10^{-2}	4.77×10^{-1}
PINN(5)	4.07×10^{-2}	3.66×10^{-2}	3.55×10^{-1}
GAM(1)	4.51×10^{-2}	4.25×10^{-2}	4.21×10^{-1}
GAM(5)	4.08×10^{-2}	3.72×10^{-2}	3.87×10^{-1}
AR(1)	4.75×10^{-2}	4.67×10^{-2}	4.55×10^{-1}
AR(5)	4.35×10^{-2}	3.91×10^{-2}	2.43×10^{-1}
ARF-GAR(1)	5.94×10^{-2}	3.62×10^{-2}	2.09×10^{-1}
ARF-GAR(5)	5.85×10^{-2}	3.58×10^{-2}	4.95×10^{-1}
LSTM	5.02×10^{-2}	4.76×10^{-2}	1.36×10^{-1}

Table 3. The summary statistics of absolute errors across 50 random samples for Scenario 2, where $a_0 = 0.5$, $c = 0.3$, and $d = 0.2$.

Method	MAE	ACF	Shapiro
PINN(1)	4.21×10^{-2}	4.34×10^{-2}	4.85×10^{-1}
PINN(5)	4.00×10^{-2}	3.51×10^{-2}	5.11×10^{-1}
GAM(1)	3.95×10^{-2}	3.85×10^{-2}	5.04×10^{-1}
GAM(5)	3.85×10^{-2}	3.22×10^{-2}	4.77×10^{-1}
AR(1)	4.55×10^{-2}	4.77×10^{-2}	4.66×10^{-1}
AR(5)	4.21×10^{-2}	3.88×10^{-2}	5.06×10^{-1}
ARF-GAR(1)	6.31×10^{-2}	3.66×10^{-2}	6.44×10^{-1}
ARF-GAR(5)	6.17×10^{-2}	3.42×10^{-2}	7.22×10^{-1}
LSTM	5.69×10^{-2}	6.51×10^{-2}	2.14×10^{-1}

Table 4. The summary statistics of absolute errors across 50 random samples for Scenario 2, where $a_0 = 0.5$, $c = 0.3$, and $d = 0.8$.

Method	MAE	ACF	Shapiro
PINN(1)	5.07×10^{-2}	4.36×10^{-2}	5.23×10^{-1}
PINN(5)	4.21×10^{-2}	4.01×10^{-2}	3.99×10^{-1}
GAM(1)	4.88×10^{-2}	3.64×10^{-2}	6.77×10^{-1}
GAM(5)	4.05×10^{-2}	3.82×10^{-2}	3.78×10^{-1}
AR(1)	5.44×10^{-2}	4.01×10^{-2}	4.91×10^{-1}
AR(5)	4.78×10^{-2}	4.55×10^{-2}	3.66×10^{-1}
ARF-GAR(1)	6.70×10^{-2}	3.66×10^{-2}	6.44×10^{-1}
ARF-GAR(5)	6.12×10^{-2}	3.43×10^{-2}	7.31×10^{-1}
LSTM	7.05×10^{-2}	7.64×10^{-2}	1.01×10^{-1}

Table 5. The summary statistics of absolute errors across 50 random samples for Scenario 3, where $a_1 = 0.2$, $a_2 = 0.8$, and $d = 0.2$.

Method	MAE	ACF	Shapiro
PINN(1)	4.15×10^{-2}	3.89×10^{-2}	3.54×10^{-1}
PINN(5)	4.05×10^{-2}	3.60×10^{-2}	3.87×10^{-1}
GAM(1)	4.00×10^{-2}	3.80×10^{-2}	3.49×10^{-1}
GAM(5)	3.95×10^{-2}	3.52×10^{-2}	3.80×10^{-1}
AR(1)	4.21×10^{-2}	3.68×10^{-2}	3.83×10^{-1}
AR(5)	4.35×10^{-2}	3.47×10^{-2}	3.62×10^{-1}
ARF-GAR(1)	5.76×10^{-2}	3.58×10^{-2}	4.13×10^{-1}
ARF-GAR(5)	5.85×10^{-2}	3.46×10^{-2}	6.83×10^{-1}
LSTM	5.17×10^{-2}	5.59×10^{-2}	4.69×10^{-1}

Table 6. The summary statistics of absolute errors across 50 random samples for Scenario 3, where $a_1 = 0.2$, $a_2 = 0.8$, and $d = 0.8$.

Method	MAE	ACF	Shapiro
PINN(1)	5.24×10^{-2}	4.71×10^{-2}	4.95×10^{-1}
PINN(5)	4.11×10^{-2}	4.15×10^{-2}	5.57×10^{-1}
GAM(1)	4.99×10^{-2}	4.78×10^{-2}	5.54×10^{-1}
GAM(5)	4.02×10^{-2}	4.08×10^{-2}	5.60×10^{-1}
AR(1)	5.20×10^{-2}	4.62×10^{-2}	6.69×10^{-1}
AR(5)	4.39×10^{-2}	4.19×10^{-2}	4.64×10^{-1}
ARF-GAR(1)	6.14×10^{-2}	3.86×10^{-2}	5.88×10^{-1}
ARF-GAR(5)	5.87×10^{-2}	3.59×10^{-2}	5.90×10^{-1}
LSTM	5.17×10^{-2}	6.29×10^{-2}	2.44×10^{-1}

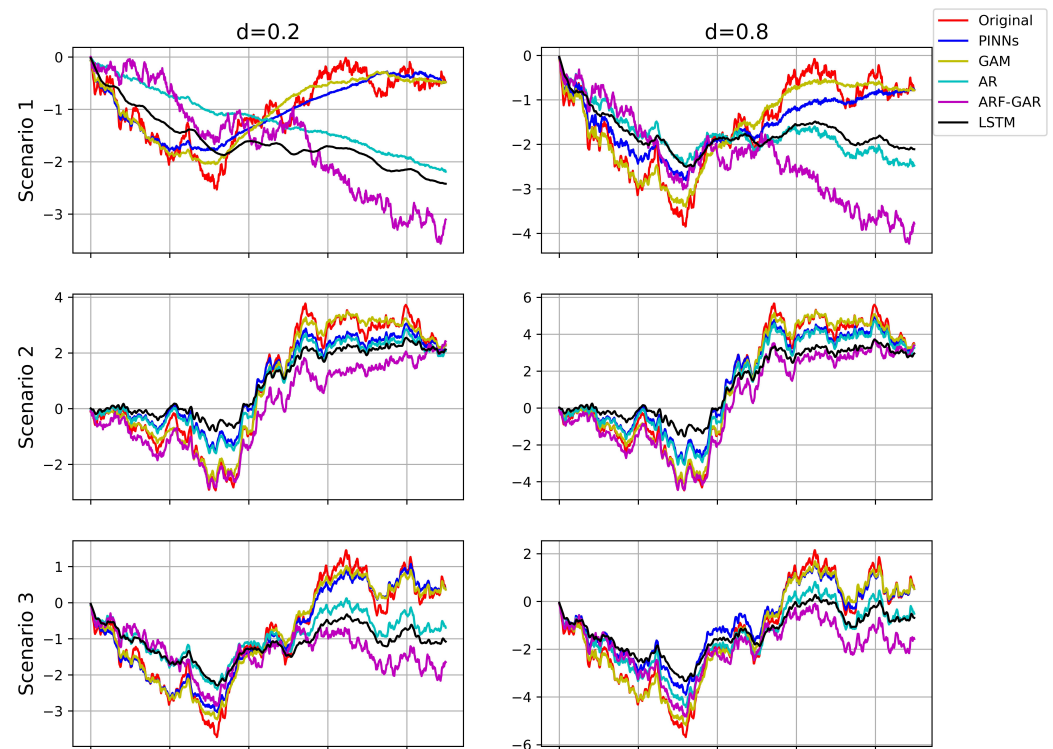


Figure 5. The Sample cumulative sum of FTVAR within different scenarios.

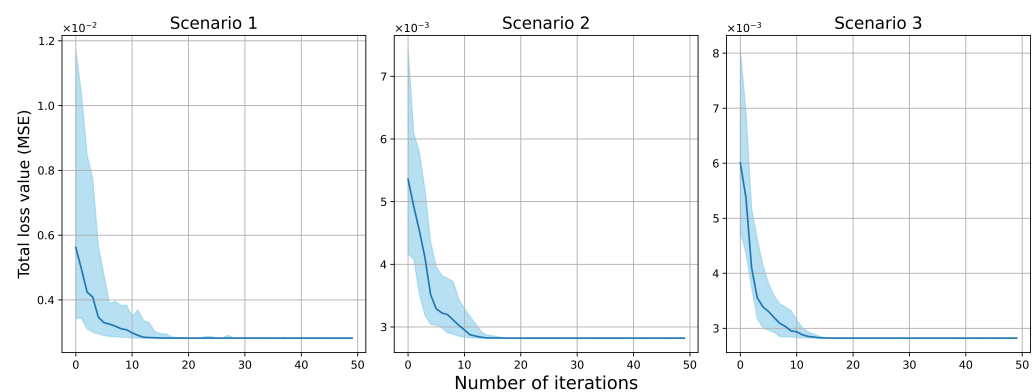


Figure 6. MSE convergence during training, with the median loss shown by the line and the shaded band representing 10–90% of the total error range.

4.3. Discussion

Initially, when examining the sensitivity of the proposed models to the finite lag length M , we observed that when d is close to zero, both models exhibit minimal sensitivity as M increases. This outcome aligns with theoretical expectations, since when d approaches zero, the process behaves as a short-memory series, and additional long lags contribute little to predictive performance. For larger values of d , model performance improves slightly with increasing M , although the effect remains marginal. This behavior may arise partly because the series length is finite, larger M values reduce the effective training data, and partly due to the strength of modern open-source optimizers, which can efficiently capture dependencies even with relatively small lag structures. Hence, to balance this trade-off, we recommend setting M to approximately 5% of the total series length, which provides sufficient temporal depth for FTVAR modeling while retaining enough data for effective training. Next, we discuss the overall strengths and weaknesses of all considered methods in light of the results of simulations.

Stationary vs. non-stationary: Based on the MAE of one-step prediction, the FTVAR model generally outperforms the traditional AR model, though our experiments reveal that this advantage is not guaranteed, as in certain scenarios the AR model can perform better. A primary reason for this reversal lies in overfitting: both the GAM and the PINNs-based implementations of FTVAR introduce substantially more parameters than the comparatively simple AR model. When the data are dominated by high levels of noise, or when the underlying dynamics exhibit only limited variability, these additional degrees of freedom can lead to overfitting to noise. In such cases, the expected advantage of non-stationary modeling may be diminished or even eliminated, allowing the more parsimonious stationary AR model to achieve superior predictive accuracy. Moreover, although the AR model typically yields competitive MAE with residuals that are statistically acceptable, residual ACF values below 0.05, and relatively high mean p-values in the Shapiro–Wilk test, its one-step predictions are often smoother than those produced by GAM- or PINNs-based FTVAR models. This effect is especially evident in Scenario 1, which aligns well with the characteristics of the AR model, as its predictions can be interpreted as a form of conditional mean estimation. Next, compared with the ARFIMA-GARCH model, both the GAM- and PINNs-based approaches exhibit consistently superior performance across MAE and ACF. This improvement highlights their enhanced capacity to capture nonlinear and non-stationary behaviors inherent in the FTVAR process. In contrast, the ARFIMA-GARCH model, which assumes a relatively stable mean and volatility structure, exhibits greater randomness but limited adaptability to time-varying dynamics, resulting in reduced accuracy when modeling such complex processes.

PINNs-based vs. GAM-based: Our simulations suggest that neither PINNs nor GAM-based models can be regarded as universally superior. PINNs offer greater flexibility by bypassing predefined basis functions, but this comes at the cost of a substantially larger parameter space that makes them more susceptible to noise and highly sensitive to the choice of activation functions, introducing additional variability in performance. Conversely, the performance of GAM models depends heavily on the choice of basis functions, and inadequate specification can lead to underfitting and reduced adaptability in complex or rapidly changing environments. As a result, the trade-offs between flexibility and robustness make direct comparisons between PINNs- and GAM-based approaches nontrivial, with their relative effectiveness ultimately depending on the characteristics of the data and the modeling objectives.

Length and lag order: Our simulations indicate that increasing the order generally improves performance: as the lag length grows, both the MAE and ACF decrease, suggesting that higher-order specifications allow the PINNs- and GAM-based approaches to better capture the underlying temporal structure of the process. Regarding sample length, although we did not explicitly report it in the main results, additional experiments revealed that our proposed methods are largely insensitive to the length of the simulated series. Specifically, when evaluating the same process with lengths of 750, 1000, and 2000 observations, the resulting performance metrics remained consistent, indicating that the approaches maintain robustness across different data sizes. This robustness highlights the adaptability of both frameworks to varying empirical settings without requiring strict tuning of sample length.

Convergence and stability of the PINNs-based approach: As illustrated in Figure 6, the training loss exhibits a consistent downward trend with minor oscillations, suggesting that the proposed PINNs loss formulation is well aligned with the optimization dynamics of the open-source ADAM minimizer. The adopted learning configuration effectively maintains a balance between numerical stability and fitting accuracy, as evidenced by the steady reduction in loss that reflects the network's progressive capture of the underlying system dynamics without signs of divergence. Overall, these results confirm the reliability of the

proposed framework in achieving stable and convergent training performance, even in the presence of non-stationary and time-varying components.

4.4. Limitations

The main limitation of both the GAM- and PINNs-based approaches lies in their large number of parameters. For instance, in the GAM-based framework, assuming the use of 6 basis functions with lag order (r), the total number of parameters to be estimated is $6r + 1$ without an intercept and $6(r + 1) + 1$ with an intercept. Similarly, in the PINNs-based approach, if the network consists of n sequential layers with dense sizes $d_1, \dots, d_n$, the total number of parameters to be estimated is $\sum_{i=1}^{n-1} d_i d_{i+1}$. Specifically, from a computational standpoint, the proposed framework exhibits moderate complexity arising from its large parameter space and the iterative nature of the ADAM and BFGS optimization algorithms. While these optimizers ensure stable and reliable convergence, the overall training cost increases substantially with the number of hidden nodes, revealing a trade-off between robustness and efficiency. This limitation highlights the need for future research to systematically analyze the computational characteristics of different network architectures and optimization schemes to achieve a more balanced compromise among robustness, accuracy, and computational efficiency. Furthermore, to clearly illustrate these computational costs, particularly for the PINN-based framework relative to the GAM-based approach, the detailed time consumption of each method is summarized in Table 7.

Table 7. Mean computational time for the PINN- and GAM-based methods across Scenario 1 and 2.

Methods	Lags	Second	Lags	Second
GAM	1	5	5	38
PINN	1	15	5	40

Another key limitation concerns the risk of overfitting. Owing to the large number of parameters, both GAM- and PINNs-based approaches exhibit heightened sensitivity to noise in the data. This sensitivity increases the likelihood that the models capture spurious patterns rather than genuine temporal dependencies, leading to overfitting. As a result, the supposed advantage of the flexible FTVAR framework over traditional AR models may be reduced, or even eliminated, when the fitted models adapt excessively to noise. This issue highlights the importance of careful model regularization, hyperparameter tuning, and validation strategies to ensure that the enhanced flexibility of GAM- and PINNs-based FTVAR models in predictive performance rather than artifacts of overfitting.

5. Application to Financial Volatility Series

In this real-world application, we model financial time series using key market indicators that capture both historical dynamics and forward-looking investor expectations. Specifically, we consider the S&P 500 3-month volatility index, representing overall U.S. equity market performance, and the VIX index, which reflects market volatility and anticipates near-term market risk. The S&P 500 3-month volatility index effectively capturing the patterns of historical volatility clustering and persistence. In contrast, the VIX provides a forward-looking measure of implied volatility, reflecting market participants' expectations of future risk. By integrating these complementary measures, our analysis accounts for both observed and expected volatility, providing a comprehensive framework for modeling and understanding financial market dynamics.

All financial data used in this study were obtained from publicly available sources, including the Federal Reserve Economic Data (FRED, <https://fred.stlouisfed.org/>) and the Chicago Board Options Exchange (CBOE, <https://www.cboe.com/>), as shown in

Figure 7. These indices are widely recognized in both academic research and industry practice for their ability to capture market behavior, investor sentiment, and macro-financial conditions. Their properties, including persistence, volatility clustering, and long-memory effects, make them particularly suitable for FTVAR modeling, allowing us to evaluate how these frameworks capture both the historical and forward-looking characteristics of financial volatility.

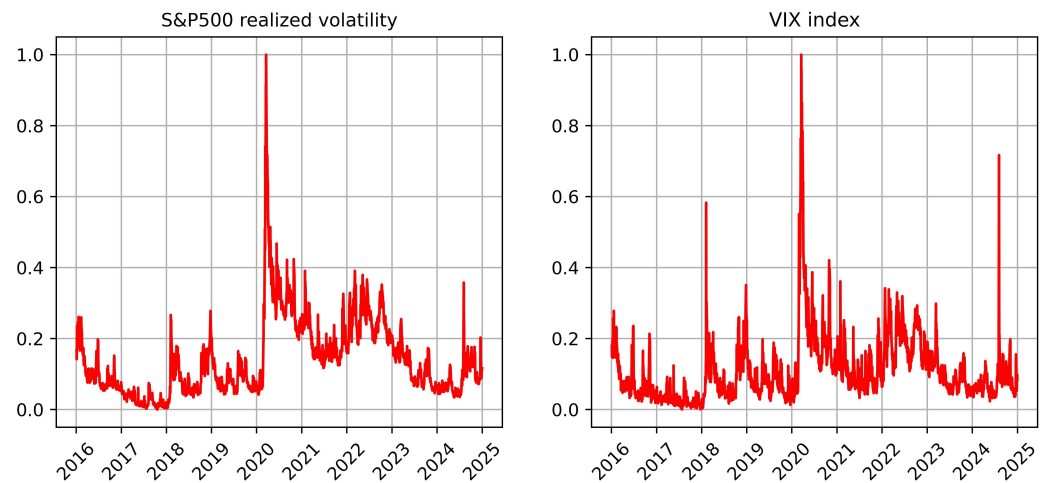


Figure 7. Normalized value of S&P 500 3-month volatility and VIX.

To evaluate the performance of our methods on real-world financial data, we use the dataset spanning 4 January 2016 to 31 December 2024 to train the TV-VAR model under both the GAM- and PINNs-based frameworks with lag orders of $r = 1$ and $r = 5$, performing one-step-ahead forecasting and comparing the predicted values to the actual observations. In addition to MAE, we assess temporal dependence to evaluate model adequacy. Unlike simulations, where ACF can be applied to stationary residuals, real-world series are often non-stationary, making residuals less meaningful. Since both the VIX and the S&P 500 3-month volatility can be considered approximately stationary, we apply the ACF directly to the predicted sequence to preserve the temporal ordering and long-range dependence of the original data, allowing us to assess whether the model captures the underlying stationary or quasi-stationary structure of financial volatility. The complete results are presented in Figures 8 and 9, and Table 8. Next, to enable a deeper analysis during the COVID-19 pandemic, a specific result is presented in Table 9.

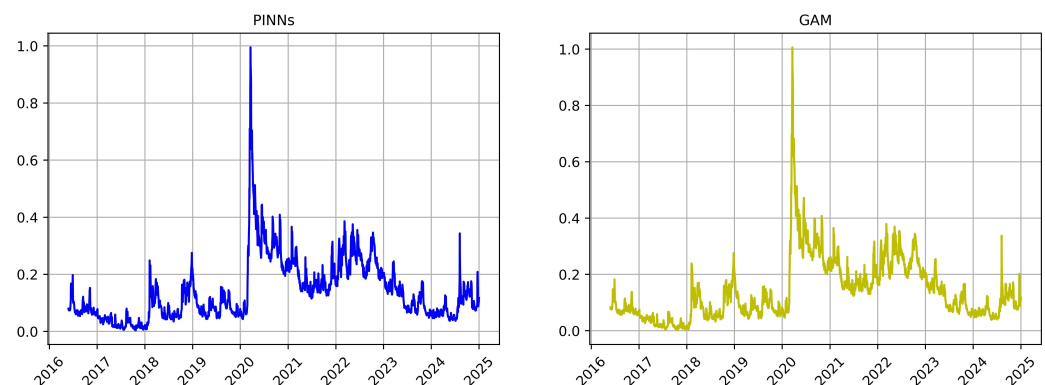


Figure 8. The prediction results of PINNs- and GAM-based approach with lag $r = 5$ for S&P 500 3-month volatility.

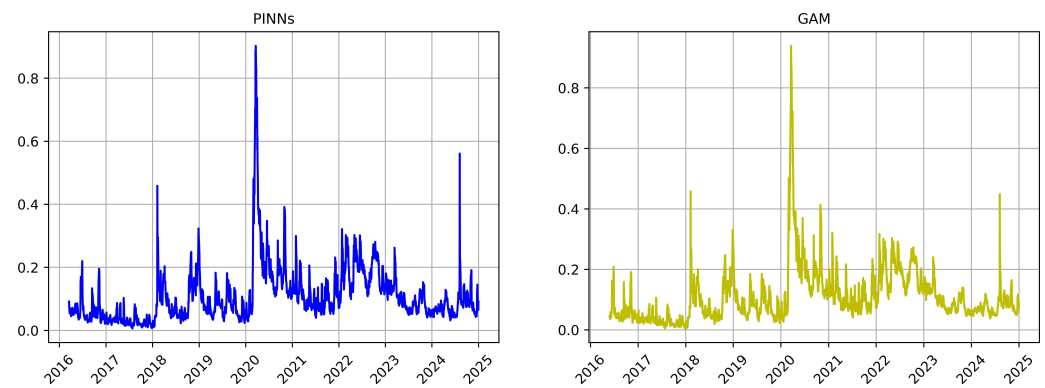


Figure 9. The prediction results of PINNs- and GAM-based approach with lag $r = 5$ for VIX index.

Table 8. MAE and ACF comparison of the PINNs- and GAM-based methods against ARFIMA-GARCH and LSTM benchmarks on the S&P 500 3-month volatility and VIX index.

Methods	MAE (S&P500)	ACF (S&P500)	MAE (VIX)	ACF (VIX)
PINN(1)	1.67×10^{-2}	4.62×10^{-2}	2.27×10^{-2}	8.64×10^{-2}
PINN(5)	1.25×10^{-2}	1.09×10^{-2}	1.76×10^{-2}	8.78×10^{-3}
GAM(1)	1.21×10^{-2}	7.07×10^{-3}	1.82×10^{-2}	2.10×10^{-2}
GAM(5)	1.20×10^{-2}	7.17×10^{-3}	1.77×10^{-2}	2.60×10^{-2}
ARF-GAR(1)	1.97×10^{-2}	1.92×10^{-2}	2.86×10^{-2}	4.82×10^{-2}
ARF-GAR(5)	1.92×10^{-2}	3.23×10^{-2}	2.79×10^{-2}	4.97×10^{-2}
LSTM	2.46×10^{-2}	5.26×10^{-2}	3.26×10^{-2}	4.54×10^{-2}

Table 9. MAE and ACF comparison of the PINNs- and GAM-based methods against ARFIMA-GARCH and LSTM benchmarks during the COVID-19 pandemic (1 January 2020–31 December 2022).

Methods	MAE (S&P500)	ACF (S&P500)	MAE (VIX)	ACF (VIX)
PINN(1)	2.48×10^{-2}	1.07×10^{-2}	3.11×10^{-2}	1.15×10^{-2}
PINN(5)	1.88×10^{-2}	5.23×10^{-3}	2.45×10^{-2}	9.81×10^{-3}
GAM(1)	1.76×10^{-2}	2.24×10^{-3}	2.53×10^{-2}	3.74×10^{-3}
GAM(5)	1.75×10^{-2}	2.44×10^{-3}	2.44×10^{-2}	5.38×10^{-2}
ARF-GAR(1)	2.50×10^{-2}	2.17×10^{-2}	3.65×10^{-2}	4.39×10^{-2}
ARF-GAR(5)	2.74×10^{-2}	1.43×10^{-2}	3.60×10^{-2}	1.89×10^{-2}
LSTM	3.59×10^{-2}	7.34×10^{-2}	4.13×10^{-2}	4.20×10^{-2}

The findings show that both the PINNs- and GAM-based approaches achieve strong predictive performance for the VIX, indicating that the proposed FTVAR framework effectively captures the dynamics of implied volatility. In contrast, the S&P 500 3-month volatility presents a greater challenge: predictive accuracy declines sharply around 2020, when the COVID-19 shock triggered unprecedented fluctuations. Compared with the traditional ARFIMA-GARCH and LSTM models, our two approaches demonstrate notably higher performance, particularly when larger lag structures are considered. This highlights the limitation of both methods in handling abrupt structural breaks, while also demonstrating the flexibility of the FTVAR approach through the estimated baseline curve for S&P 500 volatility. Unlike the original straight-line structure, the model captures a non-linear increase-then-decrease trend, particularly evident in the GAM-based estimates, reflecting its ability to adapt to gradual regime shifts. For the VIX, both methods again perform well, though the PINNs results display more variability than the smoother GAM outputs, indicating a trade-off between flexibility and stability in model choice. Overall,

these results emphasize the strengths of our framework in capturing persistent volatility structures, while also underscoring the difficulty of modeling extreme market episodes.

Finally, as shown in Table 9, during the COVID-19 period, both the PINN- and GAM-based approaches achieve lower prediction errors for S&P 500 3-month volatility and the VIX. These methods also produce lower ACF values, indicating that the reconstructed series better preserves the temporal dependence present in the original data. Comparing Tables 8 and 9 shows that MAEs for both volatility measures increase during the stress period, reflecting heightened turbulence. Nevertheless, ACFs are lower in the COVID-19 period than over the entire sample. Taken together, these results indicate that, even under stress, our model outperforms the traditional benchmark in predictive accuracy and that the reconstructed series maintains stronger temporal correlation with the observed data during the crisis.

6. Conclusions

In conclusion, this research advances the modeling and prediction of financial volatility by introducing a fractional time-varying autoregressive (FTVAR) framework, operationalized through redesigned Generalized Additive Models (GAMs) and Physics-Informed Neural Networks (PINNs). Simulation experiments and empirical analyses on real financial datasets demonstrate the effectiveness of these approaches in improving predictive accuracy and capturing complex volatility dynamics. Nonetheless, practical challenges remain: the high dimensionality of parameters can lead to overfitting and can significantly increase computational demands, particularly in the case of PINNs, which require careful training and hyperparameter tuning. These limitations highlight the trade-off between model flexibility and efficiency, underscoring the need for further methodological advances to enhance scalability and ensure reliable application in real-time or resource-constrained environments.

Finally, future research in FTVAR modeling can be advanced along several promising directions. One avenue is the extension to high-dimensional settings, such as fractional vector autoregressive and vector time-varying autoregressive frameworks, which would enable the joint modeling of multivariate financial or economic systems with complex cross-dependencies and evolving structures. Additionally, while the proposed framework is primarily formulated for numerical or continuous problems in which variables evolve smoothly over time or space, extending it to discrete domains presents a promising direction for future research. With the incorporation of suitable discretization strategies, the continuous formulations used in our model could be effectively adapted to handle discrete data structures or categorical processes. Furthermore, the assumption on β_s introduced in Equation (12) may be relatively stringent; relaxing or generalizing this condition to encompass a broader class of processes constitutes another potential avenue for theoretical advancement. Another direction is the integration of moving-average components, leading to fractional time-varying autoregressive moving-average models, which would allow for richer noise structures and more accurate volatility representations. Moreover, while physics-informed neural networks (PINNs) offer a natural tool for embedding structural constraints, their reliance on explicit PDE/ODE formulations limits applicability in contexts lacking clear governing equations. Broadening PINNs to operate effectively in such data-driven environments represents another promising avenue, with the potential to enhance scalability while retaining interpretability.

Author Contributions: Conceptualization, Z.J.; Methodology, Z.J.; Formal analysis, N.R.; Investigation, Z.J.; Writing—original draft, Z.J.; Writing—review & editing, N.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by Basic Research Program of Jiangsu province under grant number BK20250808.

Data Availability Statement: The data and code used in this study are available on GitHub at https://github.com/zhixuan1994/Fractional_TVARE.git (accessed on 7 October 2025) for purposes of replication and further research. Additionally, they will be deposited in a publicly accessible repository upon publication to ensure transparency and facilitate broader access to the research community.

Conflicts of Interest: The authors declare no competing interests relevant to this research.

References

1. Malladi, R.K.; Dheeriy, P.L. Time series analysis of cryptocurrency returns and volatilities. *J. Econ. Financ.* **2021**, *45*, 75–94. [CrossRef]
2. Popovska, E.; Georgieva-Tsaneva, G. Fractal-Based Robotic Trading Strategies Using Detrended Fluctuation Analysis and Fractional Derivatives: A Case Study in the Energy Market. *Fractal Fract.* **2025**, *9*, 5. [CrossRef]
3. Benzid, L.; Saâdaoui, F. Forecasting volatility in the EUR/USD exchange rate utilizing fractional autoregressive models. *Fluct. Noise Lett.* **2024**, *23*, 2440032. [CrossRef]
4. Liu, F.; Umair, M.; Gao, J. Assessing oil price volatility co-movement with stock market volatility through quantile regression approach. *Resour. Policy* **2023**, *81*, 103375. [CrossRef]
5. Attilio, L.A.; Mollick, A.V. Assessing the baseline model of WTI oil and stock returns under financial volatility and spillover effects. *Energy Econ.* **2024**, *135*, 107643. [CrossRef]
6. Petrova, M.; Todorov, T. Empirical Testing of Models of Autoregressive Conditional Heteroscedasticity Used for Prediction of the Volatility of Bulgarian Investment Funds. *Risks* **2023**, *11*, 197. [CrossRef]
7. Tran, M.P.B.; Vo, D.H. Asia-Pacific stock market return and volatility in the uncertain world: Evidence from the nonlinear autoregressive distributed lag approach. *PLoS ONE* **2023**, *18*, e0285279. [CrossRef]
8. Armeanu, D.S.; Gherghina, S.C.; Andrei, J.V.; Joldes, C.C. Evidence from the nonlinear autoregressive distributed lag model on the asymmetric influence of the first wave of the COVID-19 pandemic on energy markets. *Energy Environ.* **2023**, *34*, 1433–1470. [CrossRef]
9. Olasehinde-Williams, G.; Omotosho, R.; Bekun, F.V. Interest rate volatility and economic growth in Nigeria: New insight from the quantile autoregressive distributed lag (QARDL) model. *J. Knowl. Econ.* **2024**, *15*, 20172–20195. [CrossRef]
10. Kirichenko, L.; Lavrynenko, R. Probabilistic Machine Learning Methods for Fractional Brownian Motion Time Series Forecasting. *Fractal Fract.* **2023**, *7*, 517. [CrossRef]
11. Lavrynenko, R.; Kirichenko, L.; Ryabova, N.; Lavrynenko, S. Impact of outliers on median-based and mean-based methods for anomalous diffusion exponent estimation. *J. Phys. Photonics* **2025**, *7*, 025018. [CrossRef]
12. Kirichenko, L.; Koval, Y.; Yakovlev, S.; Chumachenko, D. Anomaly Detection in Fractal Time Series with LSTM Autoencoders. *Mathematics* **2024**, *12*, 3079. [CrossRef]
13. Su, X. Predictive Modeling of Volatility Using Generative Time-Aware Diffusion Frameworks. *J. Comput. Technol. Softw.* **2025**, *4*.
14. Sadorsky, P. Using machine learning to predict clean energy stock prices: How important are market volatility and economic policy uncertainty? *J. Clim. Financ.* **2022**, *1*, 100002. [CrossRef]
15. Ersin, Ö.Ö.; Bildirici, M. Financial volatility modeling with the GARCH-MIDAS-LSTM approach: The effects of economic expectations, geopolitical risks and industrial production during COVID-19. *Mathematics* **2023**, *11*, 1785. [CrossRef]
16. Xu, K.; Wu, Y.; Jiang, M.; Sun, W.; Yang, Z. Hybrid LSTM-GARCH framework for financial market volatility risk prediction. *J. Comput. Sci. Softw. Appl.* **2024**, *4*, 22–29.
17. Lin, Y.; Lin, Z.; Liao, Y.; Li, Y.; Xu, J.; Yan, Y. Forecasting the realized volatility of stock price index: A hybrid model integrating CEEMDAN and LSTM. *Expert Syst. Appl.* **2022**, *206*, 117736. [CrossRef]
18. Kumar, S.; Rao, A.; Dhochak, M. Hybrid ML models for volatility prediction in financial risk management. *Int. Rev. Econ. Financ.* **2025**, *98*, 103915. [CrossRef]
19. Raissi, M.; Perdikaris, P.; Karniadakis, G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
20. Rasht-Behesht, M.; Huber, C.; Shukla, K.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions. *J. Geophys. Res. Solid Earth* **2022**, *127*, e2021JB023120. [CrossRef]

21. Liu, Y.; Zhang, X.; Dong, Q.; Chen, G.; Li, X. Phase-resolved wave prediction with linear wave theory and physics-informed neural networks. *Appl. Energy* **2024**, *355*, 121602. [[CrossRef](#)]
22. Pu, J.; Li, J.; Chen, Y. Solving localized wave solutions of the derivative nonlinear Schrödinger equation using an improved PINN method. *Nonlinear Dyn.* **2021**, *105*, 1723–1739. [[CrossRef](#)]
23. Bai, Y.; Chaolu, T.; Bilige, S. Solving Huxley equation using an improved PINN method. *Nonlinear Dyn.* **2021**, *105*, 3439–3450. [[CrossRef](#)]
24. Khaleghi, A.; Oshnoei, S.; Mirzajani, S. Federated Learning Detection of Cyberattacks on Virtual Synchronous Machines Under Grid-Forming Control Using Physics-Informed LSTM. *Fractal Fract.* **2025**, *9*, 569. [[CrossRef](#)]
25. Olivares, E.; Ye, H.; Herrero, A.; Nia, B.A.; Ren, Y.; Donovan, R.P.; Flaviis de, F. Applications of information channels to physics-informed neural networks for WiFi signal propagation simulation at the edge of the industrial internet of things. *Neurocomputing* **2021**, *454*, 405–416. [[CrossRef](#)]
26. Hainaut, D.; Casas, A. Option pricing in the Heston model with physics inspired neural networks. *Ann. Financ.* **2024**, *20*, 353–376. [[CrossRef](#)]
27. Wang, X.; Li, J.; Li, J. A Deep Learning Based Numerical PDE Method for Option Pricing. *Comput. Econ.* **2023**, *62*, 149–164. [[CrossRef](#)]
28. Tavakoli, K.; Neisy, A.; Zamanpour, A. European option pricing underlying two assets using PINN. *J. Math. Model. Financ.* **2024**, *4*, 17–31. [[CrossRef](#)]
29. Mohamed, A.; Rigo-Mariani, R.; Debusschere, V.; Pin, L. Operational Planning Strategies to Mitigate Price Uncertainty in Day-Ahead Market for a Battery Energy System. *IEEE Access* **2024**, *12*, 85388–85399. [[CrossRef](#)]
30. Giudici, P.; Tarantino, B.; Roy, A. Bayesian time-varying autoregressive models of COVID-19 epidemics. *Biom. J.* **2023**, *65*, 2200054. [[CrossRef](#)]
31. Roy, A.; Karmakar, S. Time-varying auto-regressive models for count time-series. *Electron. J. Stat.* **2021**, *15*, 2905–2938. [[CrossRef](#)]
32. Mohamad, A.; Inani, S.K. Price discovery in bitcoin spot or futures during the Covid-19 pandemic? Evidence from the time-varying parameter vector autoregressive model with stochastic volatility. *Appl. Econ. Lett.* **2023**, *30*, 2749–2757. [[CrossRef](#)]
33. Chen, M.; Chow, S.M.; Hammal, Z.; Messinger, D.S.; Cohn, J.F. A person-and time-varying vector autoregressive model to capture interactive infant-mother head movement dynamics. *Multivar. Behav. Res.* **2021**, *56*, 739–767. [[CrossRef](#)]
34. Haslbeck, J.M.; Bringmann, L.F.; Waldorp, L.J. A tutorial on estimating time-varying vector autoregressive models. *Multivar. Behav. Res.* **2021**, *56*, 120–149. [[CrossRef](#)] [[PubMed](#)]
35. Wang, Q.; Wei, H.L.; Wang, L.; Xu, S. A novel time-varying modeling and signal processing approach for epileptic seizure detection and classification. *Neural Comput. Appl.* **2021**, *33*, 5525–5541. [[CrossRef](#)]
36. Yi, C.; Qiu, Y.; Chen, W.; Chen, C.; Wang, Y.; Li, P.; Yang, L.; Zhang, X.; Jiang, L.; Yao, D.; et al. Constructing time-varying directed EEG network by multivariate nonparametric dynamical granger causality. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 1412–1421. [[CrossRef](#)]
37. Vakıtbilir, N.; Sainbhi, A.S.; Islam, A.; Gomez, A.; Stein, K.Y.; Froese, L.; Bergmann, T.; McClarty, D.; Raj, R.; Zeiler, F.A. Multivariate linear time-series modeling and prediction of cerebral physiologic signals: Review of statistical models and implications for human signal analytics. *Front. Netw. Physiol.* **2025**, *5*, 1551043. [[CrossRef](#)]
38. He, J.; Gao, Z.; Jiang, Y.; Li, M. Spatial heterogeneity of long-range dependence and self-similarity of global sea surface chlorophyll concentration with their environmental impact factors analysis. *Front. Phys.* **2024**, *12*, 1331660. [[CrossRef](#)]
39. Drachal, K. Forecasting crude oil real prices with averaging time-varying VAR models. *Resour. Policy* **2021**, *74*, 102244. [[CrossRef](#)]
40. Zhao, L.T.; Wang, S.G.; Zhang, Z.G. Oil price forecasting using a time-varying approach. *Energies* **2020**, *13*, 1403. [[CrossRef](#)]
41. Kim, S.; Yun, S.B.; Bae, H.O.; Lee, M.; Hong, Y. Physics-informed convolutional transformer for predicting volatility surface. *Quant. Financ.* **2024**, *24*, 203–220. [[CrossRef](#)]
42. Cai, C.; Shang, Y. Parameter estimation for fractional autoregressive process with seasonal structure. *Stat. Theory Relat. Fields* **2025**, *9*, 255–284. [[CrossRef](#)]
43. Malik, M.F.; Chang, C.L.; Chaudhary, N.I.; Khan, Z.A.; kausar Kiani, A.; Shu, C.M.; Raja, M.A.Z. Swarming intelligence heuristics for fractional nonlinear autoregressive exogenous noise systems. *Chaos Solitons Fractals* **2023**, *167*, 113085. [[CrossRef](#)]
44. Benaklef, N.; Belaide, K.; Seba, D. Minimum Hellinger distance estimates for a periodic fractional autoregressive model. *Commun. Stat.-Simul. Comput.* **2025**, 1–24. [[CrossRef](#)]
45. Xu, X.; Jin, X.; Xiao, D.; Ma, C.; Wong, S.C. A hybrid autoregressive fractionally integrated moving average and nonlinear autoregressive neural network model for short-term traffic flow prediction. *J. Intell. Transp. Syst.* **2023**, *27*, 1–18. [[CrossRef](#)]
46. Chang, F.; Huang, H.; Chan, A.H.; Man, S.S.; Gong, Y.; Zhou, H. Capturing long-memory properties in road fatality rate series by an autoregressive fractionally integrated moving average model with generalized autoregressive conditional heteroscedasticity: A case study of Florida, the United States, 1975–2018. *J. Saf. Res.* **2022**, *81*, 216–224. [[CrossRef](#)]
47. Hosking, J. Fractional differencing. *Biometrika* **1981**, *68*, 165–176. [[CrossRef](#)]

48. Montanari, A.; Rosso, R.; Taqqu, M.S. A seasonal fractional ARIMA Model applied to the Nile River monthly flows at Aswan. *Water Resour. Res.* **2000**, *36*, 1249–1259. [[CrossRef](#)]
49. Baillie, R.T. Long memory processes and fractional integration in econometrics. *J. Econom.* **1996**, *73*, 5–59. [[CrossRef](#)]
50. Sowell, F. Maximum likelihood estimation of stationary univariate fractionally integrated time series models. *J. Econom.* **1992**, *53*, 165–188. [[CrossRef](#)]
51. Jia, Z.; Zhang, C. Time-Varying Autoregressive Models: A Novel Approach Using Physics-Informed Neural Networks. *Entropy* **2025**, *27*, 934. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.