Proceedings of Machine Learning Research **318** (2026)
The 39th Canadian Conference on Artificial Intelligence

Canadian AI 2026
25-29 May 2026,Vancouver,BC

Learning Adaptive Wiener Processes for Stochastic Financial Datasets with Physics-Informed Kolmogorov-Arnold Encoder-Decoder Networks

Joy Dip Das*, Ruppa K. Thulasiram
dasj@myumanitoba.ca
tulsi.thulasiram@umanitoba.ca
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

Abstract

Financial time series are non-stationary, heavy-tailed, and regime dependent, which complicates price forecasting and undermines the robustness of standard machine learning and deep learning models across assets. This work proposes a physics-informed framework, *Adaptive Wiener KAN-RNN*, that learns stochastic dynamics of the underlying financial dataset by operating directly in stochastic differential equation (SDE) parameter space: a Kolmogorov–Arnold Network (KAN) encoder transforms 120-day price windows into spline-based functional features tailored to drift (μ_t) and log-volatility ($\log \sigma_t$), and a long short-term memory (LSTM) or gated recurrent unit (GRU) decoder models their temporal evolution as latent state processes. A Wiener-process-based loss enforces consistency with geometric Brownian motion (GBM) by aligning the distribution of simulated and realized price paths, ensuring that the learned parameters remain stochastically coherent. Experiments on technology equities, including Apple (AAPL) and Microsoft (MSFT), show that this architecture delivers systematically lower error metrics and near-perfect explanatory power compared with dense KAN and conventional LSTM/GRU baselines, while yielding interpretable time-varying estimates of market drift and volatility.

Keywords: Physics-Informed Neural Networks, Encoder-Decoder Networks, LSTM, GRU, Stochastic Differential Equations, Financial Time Series Forecasting, stocks, Kolmogorov-Arnold Networks, Geometric Brownian Motion, Deep Learning.

1. Introduction

Financial equity markets exhibit nonlinear, unstable, and highly volatile behavior, making this apparent "noise" central to risk measurement and valuation rather than mere randomness [1]. Traditional techniques such as moving averages, exponential smoothing, and autoregressive integrated moving average (ARIMA) remain popular for their simplicity. However, their linear structure struggles with regime changes and complex dependencies in turbulent markets, motivating the adoption of machine learning and hybrid models that better capture nonlinear patterns and volatility clustering [2, 3].

Machine learning (ML) models have become central to financial price prediction because they offer greater flexibility than traditional statistical techniques, handling noisy, high-dimensional data and nonlinear dependencies using algorithms such as SVM, RF, gradient boosting, and k-nearest neighbors [4, 5]. Deep learning (DL) architectures, including recurrent neural networks (RNNs), long-short term memory (LSTM), gated recurrent unit (GRU), and transformers, further enhance performance by automatically learning hierarchical temporal representations. Hybrid ML/DL frameworks that combine methods like support vector regression (SVR) [6], random forest (RF), RNN, and transformers exploit

---

[*] Author for correspondence

complementary strengths in feature extraction, noise reduction, and temporal modeling to improve predictive accuracy, generalization, and risk management in complex markets [7–9].

Kolmogorov–Arnold Networks (KANs) [10] are a neural architecture based on the Kolmogorov -Arnold representation theorem, expressing complex multivariate functions as sums of simpler univariate functions. Unlike traditional DL models, KANs use learnable spline-based functions instead of fixed weights, capturing non-linear relationships efficiently while remaining interpretable. Their combination of accuracy, efficiency, and interpretability makes them promising for financial prediction, particularly for modeling asset prices and volatility [11]. Although effective in general time series [12], research on KANs in finance is still limited, emphasizing the need for further empirical validation.

Achieving robust predictions across heterogeneous assets is challenging, as different markets exhibit nonstationary, asset-specific volatility regimes [13]. Physics-inspired models based on stochastic processes and agent dynamics offer structured views of volatility and interactions [14], but are rarely coupled tightly with modern ML/DL to handle regime shifts and multi-scale risk. In this work, we introduce a physics-informed adaptive Wiener framework that uses a KAN–RNN architecture to infer time-varying drift (μ) and volatility (σ) from financial time series, asking whether such a model can reliably learn the latent stochastic differential equation (SDE) parameters governing current price evolution rather than directly forecasting prices; by combining RNN temporal modeling with KAN nonlinear approximation, the architecture yields interpretable, SDE-consistent parameters and is benchmarked against standalone LSTM and GRU baselines on two technology asset datasets.

The paper is organized as follows: Section 2 presents our proposed methodology of integrating the adaptive Wiener process framework and KAN-RNN architecture, Section 3 reports experimental comparisons with KAN, LSTM, and GRU, and Section 4 concludes with summarized insights from the study.

2. Methodology

The physics-informed KAN-RNN hybrid model predicts adaptive market dynamics through three linked phases: non-linear encoding, temporal decoding, and Wiener process-based optimization. A KAN encoder transforms normalized inputs into latent features using learnable spline functions, while an RNN decoder captures temporal structure. Two configurations are evaluated, KAN-LSTM ($32 \rightarrow 128$ units, ELU) and KAN-GRU ($16 \rightarrow 64$ units, ReLU), to estimate SDE parameters for drift (μ) and log volatility ($\log \hat{\sigma}$) under a custom geometric Brownian motion-inspired Wiener loss.

2.1. Wiener Processes for Financial Modeling

A Wiener process $\{W_t\}_{t\geq 0}$ is a fundamental continuous-time stochastic process with $W_0 = 0$, independent Gaussian increments $W_t - W_s \sim \mathcal{N}(0, t-s)$ for $0 \leq s < t$, and continuous sample paths. Infinitesimally, $dW_t \sim \mathcal{N}(0, dt)$, providing the principal source of randomness in stochastic differential equations used in finance.

Asset evolution is governed by the Itô SDE $dX_t = \mu(X_t, t)\, dt + \sigma(X_t, t)\, dW_t$, with drift μ and volatility σ defining expected return and instantaneous risk. Since the quadratic variation satisfies $[W]_t = t$, variance scales linearly with time, enabling volatility estimation. For discrete daily intervals $\Delta t = 1$, the dynamics simplify to $X_{t+1} = X_t \exp\Big((\mu_t - \frac{1}{2}\sigma_t^2)\Delta t + \sigma_t\sqrt{\Delta t}\, Z\Big)$, where $Z \sim \mathcal{N}(0,1)$. The proposed KAN-RNN framework adaptively learns μ_t and σ_t from historical data, modeling an efficient Wiener-driven evolution of market states.

2.2. Kolmogorov-Arnold Networks (KAN)

Kolmogorov–Arnold Networks (KAN) are neural models based on the Kolmogorov–Arnold representation theorem, which states that any continuous function $f : \mathbb{R}^d \to \mathbb{R}$, where $\mathbb{R}$ is the set of real numbers, can be expressed as compositions of univariate functions. A KAN layer computes $y = \sum_{i=1}^{m} g_i\big(\sum_{j=1}^{d} \phi_{ij}(x_j)\big)$, where ϕ_{ij} and g_i are learnable univariate mappings and x_j is the input asset price. Unlike standard layers $y = \sigma\big(W\,x\ (\text{weight} \cdot \text{input}) + b\ (\text{bias})\big)$, KAN replaces linear projections with sums of nonlinear univariate functions, enabling flexible approximation of local structure. Each ϕ_{ij} can be represented via basis expansions, e.g., $\phi_{ij}(x) = \sum_{k=1}^{K} \alpha_{ijk} B_k(x)$, where α_{ijk} are the learnable coefficients and $B_k(x)$ are the basis functions, improving interpretability and reducing parameter redundancy while retaining universal approximation capability.

2.2.1. Dense KAN Architecture

This study employs a dense KAN architecture, where a KAN layer maps $x \in \mathbb{R}^d$ to $h_i = \sum_{j=1}^{d} \sum_{k=1}^{K} \alpha_{ijk} B_k(x_j)$, followed by $z_i = \sigma(h_i)$. Stacking layers with widths $256, 128, 64, 32$ yields $z_i^{(\ell)} = \sigma\big(\sum_j \sum_k \alpha_{ijk}^{(\ell)} B_k(z_j^{(\ell-1)})\big)$ for $\ell = 1, \ldots, 4$, whose final output $z^{(4)}$ is flattened, passed through a linear layer $\hat{y} = w^\top \operatorname{vec}(z^{(4)}) + b$, and trained using an MSE loss with the Adam optimizer.

2.3. KAN-RNN Encoder–Decoder Architecture

The proposed KAN-RNN encoder–decoder maps a historical price sequence $\mathbf{X} \in \mathbb{R}^{T\times 1}$ ($T = 120$) to the GBM parameters $\mathbf{y}_{\text{pred}} = [\mu, \log(\sigma)]$ by first applying a KAN encoder that transforms each input x_t into a D-dimensional feature vector $z_t \in \mathbb{R}^D$ via learned spline-based univariate functions and a pointwise nonlinearity, yielding a feature sequence $\mathbf{Z}$, and then passing $\mathbf{Z}$ through either an LSTM or GRU decoder that updates its hidden state using standard gated recurrences to produce a final context vector $\mathbf{h}_T$, which is regularized with dropout and fed to a 2-unit dense layer to output $\hat{\mu}$ and $\log(\hat{\sigma})$, with $\hat{\sigma} = e^{\log(\hat{\sigma})}$ ensuring positive volatility and all parameters optimized under a custom Wiener/GBM loss. After grid search and manual refinement, the final setup used a 120-day look-back window, a stacked KAN with four layers (with 256, 128, 64, 32 units), KAN-LSTM and KAN-GRU hybrids with 32/16-unit KAN encoders and 128/64-unit LSTM/GRU decoders, respectively, ELU (LSTM decoder) and ReLU (GRU decoder) activations, Adam (with a reduced 10^{-4} learning rate and gradient clipping at 1.0 in one KAN-LSTM run), dropout $= 0.1$ after recurrent layers, and training controlled by validation-loss-based early stopping and model checkpointing to retain the best weights. The baseline LSTM and GRU models were extensively optimized through hyperparameter tuning to provide a fair comparison with the proposed approach, as well as with the multi-layer KAN architecture (see Section 2.2.1) that was similarly fine-tuned.

2.4. Custom Wiener Loss Function

To align training with Wiener-driven price dynamics, the network outputs a drift estimate $\widehat{\mu}$ and log-volatility $\log(\widehat{\sigma})$, from which $\widehat{\sigma} = \exp(\log(\widehat{\sigma})) > 0$ is recovered, and a one-step GBM-style forecast $\widehat{X}_{t+1} = X_t \exp(\widehat{\mu} - \frac{1}{2}\widehat{\sigma}^2 + \widehat{\sigma} Z_t)$ with $Z_t \sim \mathcal{N}(0, 1)$ is constructed; the loss is then the mini-batch mean squared error between $\widehat{X}_{t+1}$ and the observed X_{t+1}^{true}, providing a stochastic supervision signal that directly penalizes mismatch in the simulated next-step price under the learned GBM parameters.

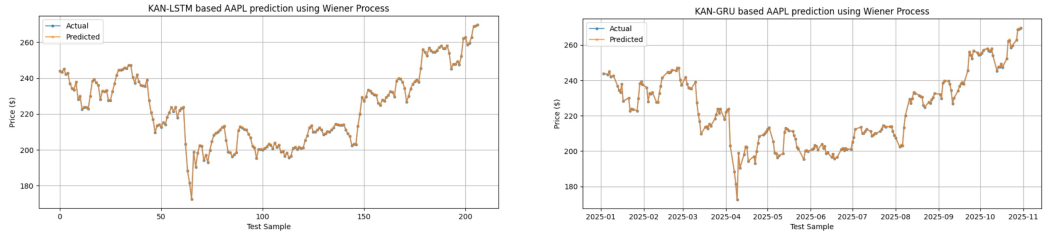
3. Results and Discussions

3.1. Data

The dataset includes two major technology assets to capture diverse market behaviors: Apple (AAPL) and Microsoft (MSFT). Daily prices from 2000 to 2025-10-31 provide a long historical window (almost 26 years), with data up to December 2024 for training and January 2025 onward for testing.

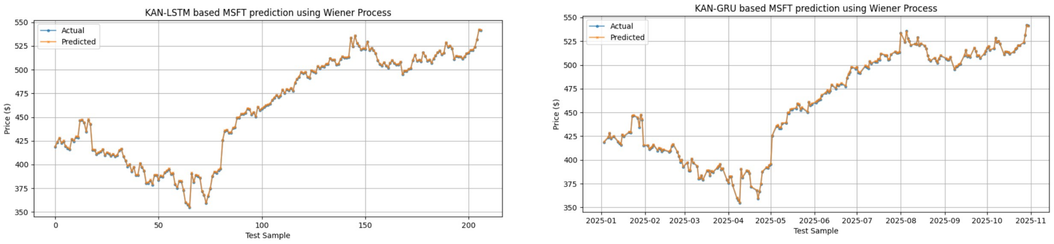
3.2. Predictive Analysis

For the technology equities AAPL and MSFT (Table 1), the standalone dense KAN produces substantially larger errors (MSE of 55.551 and 416.424, respectively) than any competing architecture, revealing that a feedforward KAN alone lacks the inductive bias required to model temporal dependencies in price trajectories. The conventional LSTM and GRU baselines reduce these errors considerably but exhibit asset-specific inconsistency: while LSTM attains a reasonable fit on AAPL (MSE = 0.592, $R^2 = 0.999$), its performance degrades markedly on MSFT (MSE = 65.484, $R^2 = 0.977$), and the GRU similarly shows wider variation across assets, suggesting that standard recurrent models struggle to generalize across equities with heterogeneous stochastic regimes.



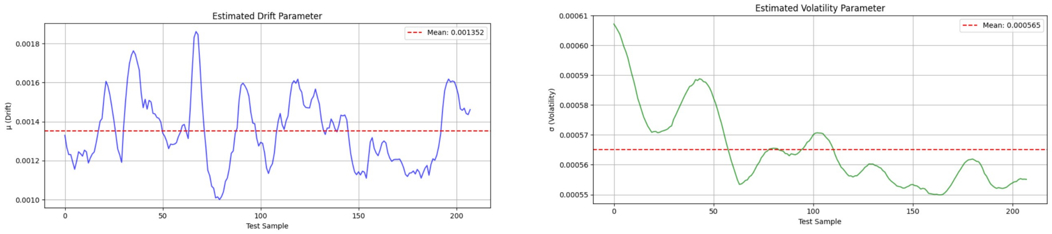
(a) AAPL: KAN-LSTM

(b) AAPL: KAN-GRU



(c) MSFT: KAN-LSTM

(d) MSFT: KAN-GRU



(e) Drift (μ)

(f) Volatility (σ)

Figure 1. Forecasting performance of KAN-RNN-Wiener models for AAPL and MSFT, together with the learned adaptive drift (μ) and volatility (σ) profiles.

By contrast, the Wiener-based KAN-LSTM and KAN-GRU architectures reduce the MSE by one to two orders of magnitude (For KAN-GRU, AAPL = 0.020, MSFT = 0.369) relative to all baselines while achieving R^2 values of 0.999 or higher for both assets, indicating near-perfect explanatory power for the daily prediction. As illustrated in Figures 1 (a)–(d), the predicted trajectories from both hybrids closely overlay the observed price paths for AAPL and MSFT, capturing trend reversals, volatility bursts, and gradual drift transitions without the overshooting or lag artifacts characteristic of the baseline recurrent models.

Sector / Model	**MSE**	**RMSE**	**MAE**	**R^2**	**MAPE**
Technology (AAPL / MSFT)					
KAN-LSTM-Wiener	0.023 / 0.342	0.152 / 0.585	0.148 / 0.577	0.999 / 0.999	0.103 / 0.136
KAN-GRU-Wiener	0.020 / 0.369	0.140 / 0.608	0.110 / 0.580	1.000 / 0.999	0.103 / 0.136
LSTM	0.592 / 65.484	0.769 / 8.092	0.640 / 7.982	0.999 / 0.977	0.280 / 1.730
GRU	0.141 / 5.111	0.376 / 2.261	0.195 / 2.084	0.999 / 0.998	0.190 / 0.450
DENSE-KAN	55.551 / 416.424	7.453 / 20.407	6.524 / 17.736	0.861 / 0.854	2.980 / 3.670

Table 1. Comparison of model performance for AAPL and MSFT (values rounded to three decimals).

Figures 1 (e) and (f) further reveal that the proposed framework successfully recovers economically meaningful time-varying drift and volatility estimates, with the learned parameters adapting coherently to structural shifts in the underlying price dynamics. Together, these results demonstrate that operating in SDE parameter space through KAN–RNN hybridization provides a more stable, interpretable, and cross-asset-consistent representation of equity price evolution than either standalone dense KAN or conventional LSTM/GRU models.

4. **Conclusion**

The proposed architectures are physics-informed, function-level hybrids that learn an adaptive Wiener process directly in SDE parameter space, targeting the latent drift and volatility that govern price evolution rather than performing unconstrained next-step forecasting. Using a KAN encoder, 120-day price windows for AAPL and MSFT are transformed into spline-based univariate compositions that build a feature space explicitly aligned with (μ_t, σ_t), while an LSTM/GRU decoder models the temporal evolution of these latent features instead of raw prices, effectively treating drift and volatility as state-dependent processes. A GBM-consistent Wiener loss then constrains the network to produce SDE parameters whose induced trajectories are statistically consistent with observed prices, embedding Brownian-increment structure into the optimization and providing a clear stochastic interpretation of the learned outputs. On the technology stocks AAPL and MSFT, this adaptive Wiener KAN–RNN framework delivers very low MSE, RMSE, MAE, R^2, and MAPE, whereas standalone dense KAN and conventional LSTM/GRU baselines exhibit error magnitudes that are one to two orders higher, underscoring that learning in SDE-parameter space via KAN–RNN hybridization yields a more robust, interpretable, and cross-asset-consistent representation of equity price dynamics.

Acknowledgements

The first author acknowledges the University of Manitoba Graduate Fellowship and the Graduate Enhancement of Tri-Agency Stipends from the University of Manitoba. The second author acknowledges funding from the Natural Sciences and Engineering Research Council of Canada through the Discovery Grant program and the research funding from the University of Manitoba.

This work used ChatGPT solely for polishing the manuscript. It was not used in the writing of the manuscript, research design, analysis, or any other components of this study.

References

[1] I. Behera, P. Nanda, S. Mitra, and S. Kumari. “Machine Learning Approaches for Forecasting Financial Market Volatility”. In: *Machine Learning Approaches in Financial Analytics* (2024), pp. 431–451.

[2] C. Cappello, A. Congedi, S. De Iaco, and L. Mariella. “Traditional prediction techniques and machine learning approaches for financial time series analysis”. In: *Mathematics* 13.3 (2025), p. 537.

[3] K. Gallagher. *The international financial architecture and sustainable prosperity.* 14/25. WIDER Working Paper, 2025.

[4] B. S. Alanazi. “A Comparative Study of Traditional Statistical Methods and Machine Learning Techniques for Improved Predictive Models”. In: *International Journal of Analysis and Applications* 23 (2025), pp. 18–18.

[5] M. Patsiarikas, G. Papageorgiou, and C. Tjortjis. “Using Machine Learning on Macroeconomic, Technical, and Sentiment Indicators for Stock Market Forecasting”. In: *Information* 16.7 (2025), p. 584.

[6] J. D. Das, A. Saumyamala, S. Bowala, R. K. Thulasiram, and A. Thavaneswaran. “Adapting Hybridization of Deep Learning Algorithms for High-Frequency Datasets”. In: *2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE. 2025, pp. 640–647.

[7] O. Shobayo, S. Adeyemi-Longe, O. Popoola, and O. Okoyeigbo. “A Comparative Analysis of Machine Learning and Deep Learning Techniques for Accurate Market Price Forecasting”. In: *Analytics* 4.1 (2025), p. 5.

[8] M. R. Kabir, D. Bhadra, M. Ridoy, and M. Milanova. “LSTM–transformer-based robust hybrid deep learning model for financial time series forecasting”. In: *Sci* 7.1 (2025), p. 7.

[9] J. D. Das, A. M. Gedara, S. Bowala, R. K. Thulasiram, and A. Thavaneswaran. “Non-Linear Data Representation with Machine Learning for Dynamic Covariance Based Financial Portfolio Optimization”. In: *2025 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CiFer)*. IEEE. 2025, pp. 1–9.

[10] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljačić, T. Y. Hou, and M. Tegmark. “KAN: Kolmogorov–Arnold Networks”. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2025. URL: https://proceedings.iclr.cc/paper_files/paper/2025/file/afaed89642ea100935e39d39a4da602c-Paper-Conference.pdf.

[11] R. Handal, K. Matoya, Y. Wang, and M. Hirano. “KANOP: A Data-Efficient Option Pricing Model using Kolmogorov-Arnold Networks”. In: *2025 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CiFer)*. IEEE. 2025, pp. 1–9.

[12] P. T. Yamak, Y. Li, T. Zhang, and M. S. Pathan. “Kolmogorov-Arnold networks for time series forecasting: a comprehensive review”. In: *Cluster Computing* 28.14 (2025), p. 929.

[13] Y. Zhu, S. I. Taasim, and A. Daud. “Volatility modeling and tail risk estimation of financial assets: Evidence from gold, oil, Bitcoin, and stocks for selected markets”. In: *Risks* 13.7 (2025), p. 138.

[14] C. Guo, S. Fang, Y. He, and Y. Zhang. “Some empirical studies for the applications of fractional G-Brownian motion in finance”. In: *Quantitative Finance and Economics* 9.1 (2025), pp. 1–39.