STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **14**, December 2025, pp 3282–3298.
Published online in International Academic Press (www.IAPress.org)

Stock Price Forecasting Using Ito-ADAM Optimized Long Short-Term Memory Neural Networks

Hamza Lahbabi [1], Zakaria Bouhanch [1],Karim El Moutaouakil [1,*], Vasile Palade [2], Alina-Mihaela Patriciu [3]

[1] *Laboratory of Mathematics and Data Science, Multidisciplinary Faculty of Taza, SidiMohamed Ben Abdellah University, Taza-35000, Morocco*
[2] *Centre for Computational Science and Mathematical Modelling, Coventry University, Priory Road, Coventry - CV1 5FB, UK*
[3] *Department of Mathematics and Computer Science, Faculty of Sciences and Environment, Dunărea de Jos University of Galaţi, Galaţi-800201 , Romania*

Abstract This study introduces a novel optimization algorithm, ItoAdam, which integrates stochastic differential calculus—specifically Itô's lemma—into the standard Adam optimizer. While Adam is widely adopted for deep learning, it often exhibits unreliable convergence in highly non-convex landscapes. ItoAdam addresses this limitation by injecting Brownian noise into the gradient updates, thereby enhancing probabilistic exploration of the loss surface and reducing the risk of trapping in poor local minima. The algorithm is applied to train Long Short-Term Memory (LSTM) neural networks for stock price forecasting using historical data from 13 major companies, including Google, Nvidia, Apple, Microsoft, and JPMorgan. Convergence of the method is theoretically established under mild assumptions, and a Differential Evolution (DE) framework is employed to optimize critical hyperparameters such as hidden size, number of layers, bidirectionality, and noise variance. Extensive experiments on the test dataset demonstrate that ItoAdam-LSTM consistently improves predictive accuracy compared to standard Adam-LSTM, as measured by RMSE, MAE, and R^2. Sensitivity analysis with respect to the Brownian noise parameter confirms that moderate noise levels yield the best generalization performance. In particular, Hotelling's T^2 test indicates that values around $\varepsilon = 10^{-8}$ achieve significantly better results than higher-noise regimes ($\varepsilon \geq 10^{-6}$), with improvements most evident in Google stock forecasting. These findings highlight the effectiveness of combining Itô-driven stochastic optimization with evolutionary hyperparameter tuning for robust financial time series prediction in noisy, nonstationary environments.

Keywords LSTM; Forecasting; Stock market; ADAM optimizer; Ito derivative

DOI: 10.19139/soic-2310-5070-2876

1. Introduction

Optimization in deep learning presents persistent challenges due to non-convex loss surfaces, the presence of numerous local minima, saddle points, and the risk of overfitting. Adaptive gradient methods such as Adam [1] have become widely adopted for their efficiency and stability. However, their deterministic update rules often limit exploratory behavior in highly irregular or noisy objective landscapes, leading to premature convergence and suboptimal generalization [2, 3]

A promising line of research seeks to address these limitations by incorporating stochastic perturbations inspired by stochastic differential equations (SDEs). In particular, Itô calculus—originally developed to handle the mathematical irregularities of Brownian motion—extends classical calculus to stochastic processes through

*Correspondence to: Karim El Moutaouakil (Email: karim.elmoutaouakil@usmba.ac.ma). Laboratory of Mathematics and Data Science, Multidisciplinary Faculty of Taza, SidiMohamed Ben Abdellah University, Taza-35000, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Itô's lemma [4, 5, 6]. This framework enables rigorous modeling of dynamic systems under uncertainty and has seen successful applications in domains such as finance, physics, biology, and control theory [7, 8].

In the context of machine learning, recent studies have explored the benefits of injecting gradient noise to emulate SDE dynamics, encouraging broader exploration of the loss landscape and potentially improving both convergence and generalization [9, 11, 18]. Building upon this foundation, this work introduces a novel stochastic optimization algorithm, referred to as **ItoAdam**. This method integrates Gaussian noise into the gradient updates, imitating the diffusion term from stochastic calculus and allowing the optimizer to escape sharp or narrow local minima more effectively.

The proposed approach is applied to train Long Short-Term Memory (LSTM) neural networks [26] for stock price forecasting, a task inherently affected by stochastic volatility and noise. To strengthen the empirical robustness and theoretical foundation, the study presents the following key contributions:

- A formal derivation of the **convergence properties** of ItoAdam under mild regularity conditions, ensuring algorithmic stability.
- Integration of ItoAdam into LSTM architectures for financial time series modeling, enabling probabilistic learning dynamics aligned with noisy data environments.
- A comprehensive comparison of numerical approximation methods for the Itô derivative—such as Euler, Milstein, Taylor–2, Weak approximation, Monte Carlo (MC), and Multi-Level Monte Carlo (MLMC)—adapted to deep learning contexts.
- Use of a **Differential Evolution (DE)** strategy to automatically optimize critical hyperparameters (e.g., hidden layer size, number of layers, bidirectionality, and noise variance), reducing manual tuning effort and enhancing performance consistency.
- Empirical evaluation on historical data from 14 major publicly traded companies (AAPL, MSFT, AMZN, GOOGL, META, NVDA, TSLA, JPM, V, UNH, MA, HD, BAC, DIS) sourced from Yahoo Finance [29, 30, 31], showing that **ItoAdam-LSTM** consistently outperforms standard Adam-based models.
- Comprehensive analysis using **quantitative metrics** (RMSE, MAE, R^2) and **visual diagnostics** to assess forecasting accuracy and interpretability.

By bridging the theoretical tools of stochastic calculus with the practical demands of modern deep learning, this research demonstrates that noise-aware optimization based on Itô dynamics can lead to statistically significant improvements in both convergence behavior and forecasting accuracy. The proposed ItoAdam algorithm offers a scalable and theoretically sound alternative to conventional optimizers, particularly in settings where model generalization and robustness under uncertainty are critical.

The remainder of the paper is organized as follows: Section 2 reviews key contributions from the literature relevant to optimization algorithms in deep learning and financial time series forecasting. Section 3 introduces the mathematical foundation of Itô calculus, emphasizing its potential in modeling stochastic behavior within learning dynamics. Section 4 presents the proposed Ito-Adam optimizer, detailing its derivation from Itô's lemma and its integration into LSTM training. Section 5 describes the experimental setup, including data sources, preprocessing, model architecture, and comparative evaluation against standard optimizers. Finally, Section 6 summarizes the main findings, highlights the contributions, and discusses potential directions for future research.

2. Related Work

Stochastic differential equations (SDEs) and Itô calculus constitute the mathematical foundation for modeling systems subject to random perturbations, particularly in domains such as mathematical finance and physics [4, 5, 6]. Classical numerical schemes, including the Euler–Maruyama and Milstein methods, have been widely adopted for approximating solutions to SDEs [13, 14]. Further developments, such as multilevel Monte Carlo methods, have enhanced the scalability and accuracy of stochastic simulations [15].

Concurrently, significant progress has been achieved in the development of optimization algorithms for deep learning. Foundational methods such as Stochastic Gradient Descent (SGD) and Momentum [40], and Nesterov Accelerated Gradient (NAG) [41] have given rise to adaptive techniques like Adagrad [37], Adadelta [38], RMSProp [39], and the widely adopted Adam optimizer [1]. Nonetheless, the convergence limitations of Adam in non-convex settings have led to enhanced variants, including AMSGrad, AdamW, AdaBelief, and RAdam [3].

More recent approaches have introduced stochastic process theory into optimization, offering new perspectives on learning dynamics. Algorithms such as Stochastic Gradient Langevin Dynamics (SGLD) and its variants incorporate Gaussian noise into gradient updates to better explore complex loss landscapes [16, 18, 9]. Optimizers framed within stochastic differential systems—including GNAdam [20], StochasticAdam [22], and AdaFair [19]—demonstrate improved generalization and robustness through noise-driven regularization.

In the field of time series modeling, Long Short-Term Memory (LSTM) networks [26] have become standard tools for capturing sequential dependencies, particularly under high-noise conditions like those encountered in financial forecasting. Empirical studies have confirmed the effectiveness of LSTM models for stock price prediction and market behavior modeling using historical data [27, 28, 29, 30, 31]. However, the performance of these models remains sensitive to the choice of optimization algorithm, especially in volatile financial environments.

Recent theoretical investigations have further clarified the connection between stochastic optimization methods and Itô diffusion processes [9, 11, 22], highlighting the advantages of controlled stochasticity in achieving flatter minima and enhanced generalization.

A recent contribution proposed a fractional-order gradient optimization approach for LSTM networks, exhibiting improved convergence and memory retention when applied to complex time series tasks [36]. This development underscores the potential of integrating stochastic calculus and fractional dynamics into next-generation learning algorithms.

Despite these advancements, the combination of Itô-based stochastic optimization techniques with LSTM architectures remains relatively unexplored, particularly in the context of large-scale financial time series forecasting. To date, few studies have systematically assessed this integration across diverse stock datasets under realistic market noise. The current work addresses this gap by introducing the *Ito-Adam* optimizer—rooted in Itô calculus—and employing it to train LSTM models on historical financial data. Experimental results demonstrate consistent improvements in RMSE, MAE, and R^2 metrics, supporting the effectiveness of incorporating stochastic dynamics into the optimization process.

3. Itô Derivatives for Stochastic Knowledge Processing

This section introduces the theoretical foundations of the Itô derivative, which generalizes differentiation to stochastic processes. The motivation comes from the fact that classical derivatives fail when applied to paths of Brownian motion, due to their irregularity and infinite variation [4, 5]. We then derive Itô's Lemma as the central tool for handling stochastic differentials, followed by practical methods for computing its terms, illustrative examples contrasting classical and stochastic derivatives, and the role of martingales. Finally, we highlight implications for machine learning, particularly in the context of stochastic optimization and financial forecasting.

3.1. Theoretical Foundation of the Itô Derivative

Let $X(t)$ be a stochastic process, and consider a smooth test function $f(t, X(t))$. In classical calculus, the chain rule gives

$$df = \frac{\partial f}{\partial t} dt + \frac{\partial f}{\partial x} dX(t).$$

This expansion relies on the assumption that higher-order terms such as $(dX(t))^2$, $dt\, dX(t)$, and $(dt)^2$ are negligible.

However, when $X(t)$ is driven by Brownian motion $W(t)$, the assumptions collapse:

- Brownian paths are continuous but nowhere differentiable [43].
- The increments scale as $dW(t) \sim \mathcal{N}(0, dt)$, so their variance is of order dt.
- Crucially, the quadratic variation of Brownian motion satisfies

$$(dW(t))^2 = dt, \qquad dW(t)\, dt = 0, \qquad (dt)^2 = 0.$$

This single identity—$(dW(t))^2 = dt$—is the cornerstone of Itô calculus. It forces us to amend the classical chain rule, yielding the corrected stochastic chain rule known as **Itô's Lemma**. The operator that incorporates this correction is called the *Itô derivative*.

Suppose $X(t)$ follows a stochastic differential equation (SDE):

$$dX(t) = \mu(t, X(t))dt + \sigma(t, X(t))dW(t),$$

with drift μ and diffusion σ. Then Itô's Lemma gives the differential of $f(t, X(t))$ as:

$$df(t, X(t)) = \left(\frac{\partial f}{\partial t} + \mu(t, X(t)) \frac{\partial f}{\partial x} + \frac{1}{2}\sigma^2(t, X(t)) \frac{\partial^2 f}{\partial x^2} \right) dt + \sigma(t, X(t)) \frac{\partial f}{\partial x}\, dW(t).$$

Compared to the classical chain rule, there is an additional correction term:

$$\frac{1}{2}\sigma^2 \frac{\partial^2 f}{\partial x^2} dt,$$

which arises exactly because $(dW(t))^2 = dt$. This term embodies the roughness of Brownian motion and is what makes the Itô derivative fundamentally different from the classical derivative [44].

3.2. Why the Itô Derivative Matters

The introduction of the Itô derivative has profound consequences for both theory and applications of stochastic systems:

1. **Mathematical rigor for random dynamics.** Classical derivatives fail for processes with infinite variation such as Brownian motion, whose paths are nowhere differentiable. The Itô derivative resolves this by defining differentiation in terms of stochastic integrals, making it possible to analyze random processes rigorously [4].
2. **Separation of drift and noise.** A defining feature of the Itô framework is the decomposition of dynamics into deterministic and stochastic components:

$$df = \underbrace{\Big(\cdots \Big) dt}_{\text{drift}} + \underbrace{\Big(\cdots \Big) dW(t)}_{\text{noise}}.$$

   This separation allows for clear interpretation of long-term trends (drift) versus short-term fluctuations (noise), and is central in both financial modeling and stochastic control [42].
3. **Foundation for applied modeling.** The Itô derivative underlies martingale theory, stochastic integration, and the Black–Scholes framework in finance. Beyond finance, it has become increasingly important in machine learning, where stochastic gradient methods can be interpreted as discretized Itô processes, with injected noise acting as a regularizer to improve generalization.
4. **Bridging disciplines.** By enabling rigorous treatment of uncertainty, the Itô derivative connects probability theory, optimization, and data-driven learning. This cross-disciplinary role explains its widespread adoption in modern quantitative fields ranging from mathematical finance to deep learning.

Examples: Classical vs. Itô Derivatives Let $dX(t) = \mu dt + \sigma dW(t)$. For selected functions $f(X(t))$, the classical and Itô derivatives differ as follows:

- For $f(X) = X^2$:

$$df_{Class} = 2X \cdot dX, \quad df_{Itô} = (2X\mu + \sigma^2)dt + 2X\sigma dW.$$

- For $f(X) = \sin(X)$:

$$df_{Class} = \cos(X)\, dX, \quad df_{Itô} = \left(\cos(X)\mu - \tfrac{1}{2}\sigma^2 \sin(X)\right) dt + \sigma \cos(X)\, dW.$$

- For $f(X) = e^X$:

$$df_{Class} = e^X dX, \quad df_{Itô} = \left(\mu e^X + \tfrac{1}{2}\sigma^2 e^X\right) dt + \sigma e^X dW.$$

3.3. Martingales and the Role of Itô's Lemma

A central concept in stochastic calculus is the *martingale*. Intuitively, a martingale represents a "fair game" process in which future expected values, conditioned on the present, equal the current value. Formally, a stochastic process $M(t)$ adapted to a filtration $\{\mathcal{F}_t\}$ is a martingale if

$$\mathbb{E}[M(t+h) \mid \mathcal{F}_t] = M(t), \quad \forall h \geq 0.$$

A canonical example is Brownian motion $W(t)$, which has independent and stationary increments. In particular, it satisfies

$$\mathbb{E}[W(t+h) - W(t) \mid \mathcal{F}_t] = 0,$$

thus ensuring that no predictable drift exists. This property is crucial for Itô calculus: the stochastic integral with respect to Brownian motion has zero expectation, a fact that underpins the interpretation of Itô's Lemma [45, 46].

Consequently, in Itô's Lemma, the stochastic term,

$$\sigma \frac{\partial f}{\partial x}\, dW(t),$$

has expectation zero, while the deterministic drift term contributes to the expected dynamics. This separation enables tractable modeling of random fluctuations while retaining analytical control over long-run behavior. In practical terms, martingale properties are widely exploited in finance (e.g., in risk-neutral pricing and derivative valuation), in stochastic control (e.g., for optimal stopping and filtering), and in machine learning (e.g., stochastic optimization with unbiased noise models). Recent studies emphasize how martingale-based methods remain central in financial mathematics and stochastic analysis [47].

In financial modeling, stock prices are often modeled by SDEs (e.g., the Black–Scholes model), where Itô calculus is essential [23]. In machine learning, embedding Itô corrections into optimizers (e.g., Ito-Adam) provides noise-aware gradient updates that align with the stochastic nature of high-variance data such as financial time series. This enhances robustness, prevents overfitting to noise, and improves forecasting accuracy in models such as LSTMs trained on stock data.

The Itô derivative extends classical differentiation to irregular stochastic processes, resolving the failure of the classical chain rule for Brownian motion. By introducing the correction term from quadratic variation, Itô's Lemma becomes the cornerstone of modern stochastic analysis, with fundamental applications across finance, physics, control theory, and deep learning.

4. Brownian Adam Optimizer to LSTM learning

We propose modifying the standard Adam optimizer by replacing the deterministic gradient $\nabla_\theta \mathcal{L}(\theta_t)$ with a stochastic Ito derivative $d\mathcal{L}(\theta_t)$.

4.1. Brownian Adam Optimizer

Start from a parameter SDE and its Euler–Maruyama discretization. Model the parameters θ_t as the solution of an SDE that captures average gradient flow plus stochastic fluctuations:

$$d\theta_t = -\mathbb{E}_\xi[g(\theta_t;\xi)]\; dt + \Sigma(\theta_t)\, dW_t, \tag{1}$$

where $g(\theta;\xi) = \nabla_\theta \ell(\theta;\xi)$ is the per-sample (or minibatch) gradient, and $\Sigma(\theta_t)$ encodes the diffusion. Using Euler–Maruyama with step size η gives the discrete approximation

$$\theta_{n+1} = \theta_n - \eta \underbrace{\mathbb{E}[g(\theta_n;\xi)]}_{\text{drift}} + \eta\, \Sigma(\theta_n)\, \xi_n, \quad \xi_n \sim \mathcal{N}(0, I). \tag{2}$$

This shows the drift term should be the (unknown) expectation of the minibatch gradient, not a single noisy sample g_n.

Replace the unknown expectation by a low-variance estimator (EMA). A natural estimator of $\mathbb{E}[g(\theta_n;\xi)]$ is the exponential moving average

$$m_n = \beta_1 m_{n-1} + (1-\beta_1)g_n, \qquad g_n \equiv g(\theta_n;\xi_n). \tag{3}$$

With bias correction,

$$\hat{m}_n = \frac{m_n}{1-\beta_1^n}, \tag{4}$$

$\hat{m}_n$ becomes approximately unbiased for $\mathbb{E}[g]$. Importantly, m_n reduces variance compared to g_n; hence $\hat{m}_n$ provides a more stable estimate of the drift.

Why precondition by $\hat{v}_n$: controlling anisotropy and conditioning. Adam also maintains a second moment EMA:

$$v_n = \beta_2 v_{n-1} + (1-\beta_2)g_n^2, \qquad \hat{v}_n = \frac{v_n}{1-\beta_2^n}. \tag{5}$$

Here $\hat{v}_n \approx \mathbb{E}[g^2]$ (coordinatewise). Dividing $\hat{m}_n$ by $\sqrt{\hat{v}_n}$:

- ensures scale-invariance, since coordinates with larger variance are scaled down and vice versa;
- stabilizes noisy coordinates, as large variance leads to smaller effective steps.

This can be viewed as a diagonal preconditioner approximating $G^{-1}\hat{m}_n$ with $G \approx \mathrm{diag}(\hat{v}_n^{1/2})$.

Dimensional analysis and consistency. Since $g_n = \partial L/\partial\theta$ has units (loss)/(parameter) and $\hat{v}_n \sim g_n^2$ has squared units, the ratio

$$\frac{\hat{m}_n}{\sqrt{\hat{v}_n}} \tag{6}$$

is dimensionless. Multiplying by step size η yields a parameter increment, consistent with the Euler–Maruyama discretization.

Bias correction and numerical stability. Bias corrections $(1-\beta_1^n)^{-1}$ and $(1-\beta_2^n)^{-1}$ compensate initialization effects. The additive $\epsilon > 0$ prevents division by zero and ensures numerical stability.

Variance reduction. If minibatch gradient variance is σ_g^2, then the EMA yields

$$\mathrm{Var}(m_n) = \frac{1-\beta_1}{1+\beta_1}\,\sigma_g^2, \tag{7}$$

which is substantially smaller than σ_g^2 when $\beta_1 \in [0.9, 0.999]$. Thus, $\hat{m}_n$ reduces the effective variability of the drift term and stabilizes optimization.

Final update. Combining these arguments, the Euler–Maruyama step with adaptive moments becomes:

$$\theta_{n+1} = \theta_n - \eta\frac{\hat{m}_n}{\sqrt{\hat{v}_n}+\epsilon} + \eta\,\Sigma(\theta_n)\,\xi_n, \quad \xi_n \sim \mathcal{N}(0,I). \tag{8}$$

This can be interpreted as a variance-reduced and preconditioned

4.2. Backpropagation with Ito-ADM for LSTM

We extend the standard backpropagation through time (BPTT) for LSTM with Ito-type stochasticity added to the gradient updates, inspired by stochastic differential equations (SDEs).

Forward Pass (Standard LSTM) As before, for time step t, we have:

$$\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f), \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i), \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o), \\
h_t &= o_t \odot \tanh(c_t).
\end{aligned}$$

Loss and Stochastic Gradient Definition Let the loss be $L = \mathcal{L}(h_T, y_T)$, and define the standard gradients $\nabla_\theta L$ for each parameter $\theta \in \{W_g, U_g, b_g\}$ with $g \in \{f, i, o, c\}$.

The Ito-ADM gradient update augments the deterministic gradient with a stochastic term:

$$d\theta = -\eta \nabla_\theta L \, dt + \sigma_\theta \, dB_t,$$

where η is the learning rate, σ_θ the volatility (noise amplitude) for parameter θ, and dB_t a Wiener process increment.

Euler–Maruyama Discretization Applying Euler–Maruyama to the SDE yields the discrete update:

$$\theta_{n+1} = \theta_n - \eta \nabla_\theta L_n + \sigma_\theta \, \epsilon_n,$$

where $\epsilon_n \sim \mathcal{N}(0, I)$. This expression shows that the stochastic perturbation is added directly to the standard gradient step.

Moment-Based Normalization Justification To improve stability, the raw gradient $g_n = \nabla_\theta L_n$ is replaced by an adaptively normalized estimate. Following Adam, two moment estimates are maintained:

$$\begin{aligned}
m_n &= \beta_1 m_{n-1} + (1 - \beta_1) g_n, \\
v_n &= \beta_2 v_{n-1} + (1 - \beta_2) g_n^2,
\end{aligned}$$

with bias-corrected forms

$$\hat{m}_n = \frac{m_n}{1 - \beta_1^n}, \qquad \hat{v}_n = \frac{v_n}{1 - \beta_2^n}.$$

The normalization by $\sqrt{\hat{v}_n} + \epsilon$ is motivated by the need to control the scale of noisy updates: since the variance of stochastic increments grows with $\hat{v}_n$, dividing by its square root ensures balanced updates across coordinates.

Thus, the stochastic update becomes

$$\theta_{n+1} = \theta_n - \eta \frac{\hat{m}_n}{\sqrt{\hat{v}_n} + \epsilon} + \sigma_\theta \, \epsilon_n.$$

Gradient Backpropagation (Modified with Noise) The standard error signals are computed:

$$\begin{aligned}
\delta_o &= \frac{\partial L}{\partial h_n} \odot \tanh(c_n) \odot o_n \odot (1 - o_n), \\
\delta_f &= \frac{\partial L}{\partial c_n} \odot c_{n-1} \odot f_n \odot (1 - f_n), \\
\delta_i &= \frac{\partial L}{\partial c_n} \odot \tilde{c}_n \odot i_n \odot (1 - i_n), \\
\delta_{\tilde{c}} &= \frac{\partial L}{\partial c_n} \odot i_n \odot (1 - \tilde{c}_n^2).
\end{aligned}$$

Stochastic Parameter Updates via Ito-ADM Each parameter update is then:

$$W_g \leftarrow W_g - \eta \sum_n \frac{\hat{m}_{W_g}^{(n)}}{\sqrt{\hat{v}_{W_g}^{(n)}} + \epsilon} + \sigma_{W_g} \cdot \epsilon_n^{(W_g)},$$

$$U_g \leftarrow U_g - \eta \sum_n \frac{\hat{m}_{U_g}^{(n)}}{\sqrt{\hat{v}_{U_g}^{(n)}} + \epsilon} + \sigma_{U_g} \cdot \epsilon_n^{(U_g)},$$

$$b_g \leftarrow b_g - \eta \sum_n \frac{\hat{m}_{b_g}^{(n)}}{\sqrt{\hat{v}_{b_g}^{(n)}} + \epsilon} + \sigma_{b_g} \cdot \epsilon_t^{(b_g)}.$$

Where $\epsilon_n^{(\cdot)} \sim \mathcal{N}(0, I)$ represents Gaussian noise at each step.

This stochastic update rule simulates gradient descent in a stochastic environment, encouraging better exploration of the loss surface and potentially escaping local minima, especially in non-convex optimization like training LSTMs.

We can extend the Itô-based framework to develop stochastic counterparts of classical optimization algorithms such as Stochastic Gradient Descent (SGD), Momentum, and Nesterov Accelerated Gradient (NAG), as well as adaptive methods including Adagrad, Adadelta, RMSProp, AMSGrad, AdamW, AdaBelief, and RAdam.

In conclusion, the Ito-ADM formulation for LSTM training introduces a stochastic differential equation perspective to backpropagation, using controlled Gaussian noise to regularize training, potentially improving generalization and convergence in complex optimization landscapes.

5. Experimental Design

We applied a Long Short-Term Memory (LSTM) neural network for time series forecasting on daily stock high prices obtained from Yahoo Finance. The selected symbols include AAPL, MSFT, AMZN, GOOGL, META, NVDA, TSLA, JPM, V, UNH, MA, HD, BAC, and DIS. Data spans from January 2020 to December 2024.

Each stock's *High* price was normalized using MinMax scaling, then segmented into overlapping sequences of 20 timesteps to predict the next value.

We trained the LSTM model with two optimization methods: standard **Adam** and a stochastic variant named **StochasticAdam**, inspired by Itô calculus. The stochastic optimizer perturbs the gradient with Gaussian noise:

$$\tilde{\nabla}_n = \nabla_n + \mathcal{N}(0, \sigma^2) \tag{9}$$

where σ denotes the noise standard deviation. This simulates Brownian motion and improves exploration, helping the model escape local minima.

Training was performed for 30 epochs with various noise levels: $\sigma \in \{0.0, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$. The forecasting performance was evaluated using RMSE, MAE, and R^2. For each stock and configuration, results were visualized and summarized to compare both optimizers.

5.1. Evaluation of Ito Derivative Approximation Methods in ItoAdam-LSTM Forecasting

Selecting the most suitable approximation algorithm for the Ito derivative is a critical component in enhancing the forecasting capabilities of hybrid deep learning models, such as the ItoAdam-LSTM framework. The choice of numerical scheme—whether Euler, Milstein, Taylor2, Weak, Monte Carlo (MC), or Multi-Level Monte Carlo (MLMC)—determines how accurately the underlying stochastic differential equations capture the real-world dynamics of stock prices. Given the complexity and volatility of financial time series, the selected method must ensure numerical stability, high approximation precision, and effective integration with data-driven learning components. Therefore, the core challenge lies in identifying a method that not only minimizes forecasting errors but also exhibits strong generalization performance from training to validation data.

Table 1. Updated performance comparison of numerical methods in the ItoAdam-LSTM forecasting framework

Method	Train RMSE	Val RMSE	Train MAE	Val MAE	Train R^2	Val R^2
Euler	4.2786	6.4837	3.3218	5.0855	0.9703	0.8220
Milstein	4.3232	6.1629	3.3327	4.7852	0.9697	0.8391
Taylor2	4.4313	6.2972	3.4747	4.9146	0.9681	0.8321
Weak	4.3073	6.1975	3.3738	4.8198	0.9699	0.8373
MC	4.4823	7.5786	3.5498	6.0923	0.9674	0.7567
MLMC	4.8071	9.1556	3.7986	7.2661	0.9625	0.6450

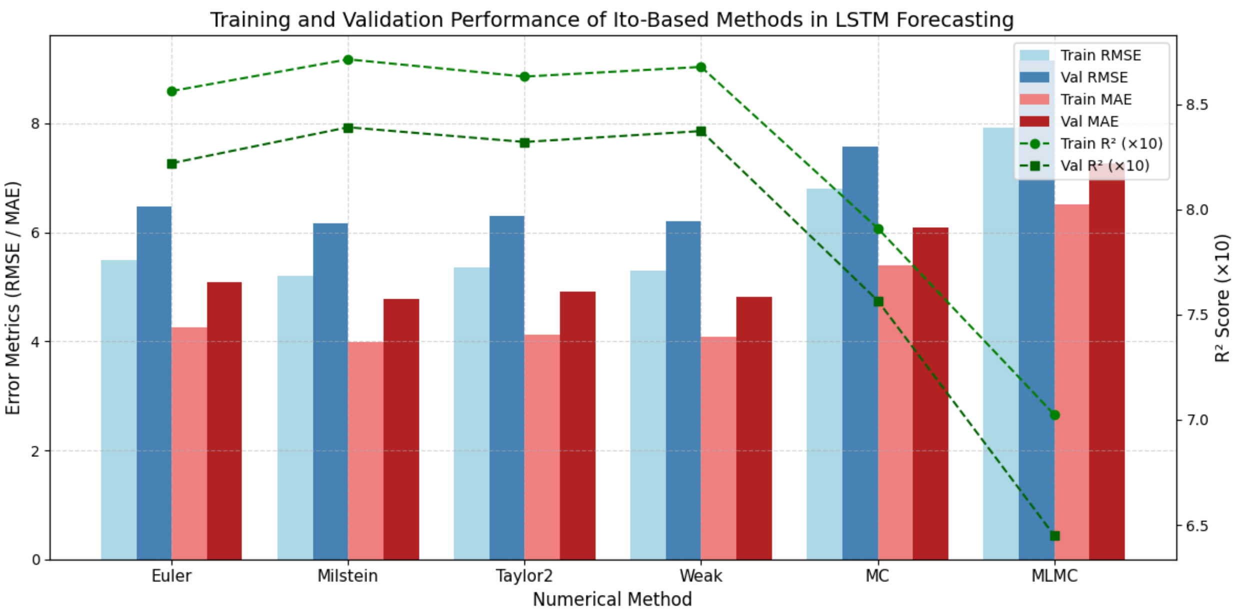

Figure 1. Comparison of validation metrics (RMSE, MAE, and R^2) for different Ito derivative approximation methods in the ItoAdam-LSTM framework. The Milstein method shows the best balance of low error and high generalization.

As shown in Table 1 and Figure 1, the numerical methods vary significantly in their forecasting performance within the ItoAdam-LSTM framework. The Euler method, while computationally simple, produces relatively high validation errors and moderate R^2 values, limiting its effectiveness. On the other end, the MLMC approach yields the worst results overall, with the highest validation RMSE and lowest R^2, suggesting it is unsuitable for this context. The Monte Carlo method, which previously demonstrated strong generalization, now underperforms with degraded validation metrics—most notably, a significant drop in R^2 to 0.7567—indicating possible overfitting or stochastic instability. In contrast, the Milstein and Weak methods consistently achieve low validation errors and high R^2 values, reflecting superior generalization and model fidelity. Taylor2 performs comparably but falls slightly behind in both error and variance explanation metrics.

Among all evaluated schemes, the Milstein method demonstrates the most favorable balance between accuracy and stability. It achieves a validation RMSE of 6.1629 and a validation R^2 of 0.8391, coupled with robust training performance that signals neither overfitting nor underfitting. The Weak method is nearly equivalent in quality, but Milstein has a marginal edge in validation metrics. These results suggest that Milstein provides a more precise and stable numerical approximation of the Ito derivative, making it the most reliable candidate for integration into the ItoAdam-LSTM forecasting framework. Therefore, Milstein is selected as the best approximation algorithm for the Ito derivative in this context, offering a strong foundation for high-fidelity financial time series modeling.

5.2. Optimization Algorithm and Model

To accurately forecast the high price of stock data using an LSTM-based neural network, it is essential to select optimal hyperparameters. In this work, we employ the Differential Evolution (DE) algorithm—a population-based global optimization method—to automatically tune four key parameters of the LSTM-StochasticAdam framework: the hidden size of the LSTM layer, the number of LSTM layers, the noise standard deviation introduced in the custom StochasticAdam optimizer, and a bidirectionality flag. The DE algorithm iteratively explores combinations of these parameters by minimizing the root mean squared error (RMSE) between predicted and actual prices on the validation set. This automated tuning process avoids manual parameter selection and enhances model adaptability across different stock datasets.

The optimization algorithm used is Differential Evolution (DE), a stochastic, population-based metaheuristic designed for global optimization of non-convex problems. It evolves a population of candidate solutions through mutation, crossover, and selection mechanisms. The optimization model formulated can be expressed as:

$$\min_{\theta=(\text{hidden\_size},\text{num\_layers},\text{noise\_std},\text{bidir})} \text{RMSE}(\text{LSTM}_\theta(X_{\text{val}}), y_{\text{val}})$$

where the decision variables are:

- hidden_size $\in [32, 128]$: number of hidden units in the LSTM layer,
- num_layers $\in [1, 3]$: number of stacked LSTM layers,
- noise_std $\in [10^{-6}, 10^{-2}]$: standard deviation of the gradient noise in StochasticAdam,
- bidir $\in \{0, 1\}$: boolean indicating bidirectionality of the LSTM.

After running the optimization process for a maximum of 5 iterations, the algorithm returned the following best parameters:

$$\text{hidden\_size} = 97, \quad \text{num\_layers} = 2, \quad \text{noise\_std} = 3.471 \times 10^{-4}, \quad \text{bidir} = \text{False}$$

with an achieved validation RMSE of approximately 0.4943.

The stopping criterion was reached due to exceeding the maximum number of iterations allowed, and the DE algorithm did not signal convergence (‘success=False‘). Nonetheless, the best solution found was used to train the final LSTM model.

5.3. Performance Comparison and Results Analysis

To evaluate the effectiveness of the proposed Ito-inspired StochasticAdam optimizer within the LSTM forecasting model (ItoAdam-LSTM), we benchmarked it against the standard Adam-LSTM across a diverse set of stock high prices. The evaluation focused on three core performance metrics: the Root Mean Squared Error (RMSE), the Mean Absolute Error (MAE), and the coefficient of determination (R^2).

Figure 2 and Figure 3 illustrate the RMSE and MAE, respectively, across all tested stock datasets for both models.

The results clearly show that the ItoAdam-LSTM model offers superior performance in most cases, particularly in terms of validation RMSE and MAE. Notable improvements were achieved on the GOOGL, AAPL, MSFT, and JPM datasets, where the RMSE values of the ItoAdam-LSTM dropped significantly compared to the standard Adam variant. These gains reflect the beneficial effect of injecting controlled stochasticity during training, allowing the optimizer to escape suboptimal local minima and discover more generalizable solutions.

In contrast, for certain datasets such as NVDA and V, the performance gains were less evident or reversed. These cases may highlight the limitations of both models when dealing with particularly volatile or nonstationary time series. The R^2 values, especially on the validation sets, indicate that both models still struggle to fully capture complex stock price dynamics in some scenarios, where values often fall below zero. This suggests that additional modeling refinements or hybrid approaches could be explored in future work.

Overall, the comparative analysis demonstrates the robustness and adaptability of the ItoAdam-LSTM framework in financial time series forecasting, particularly for stocks with moderately predictable patterns.

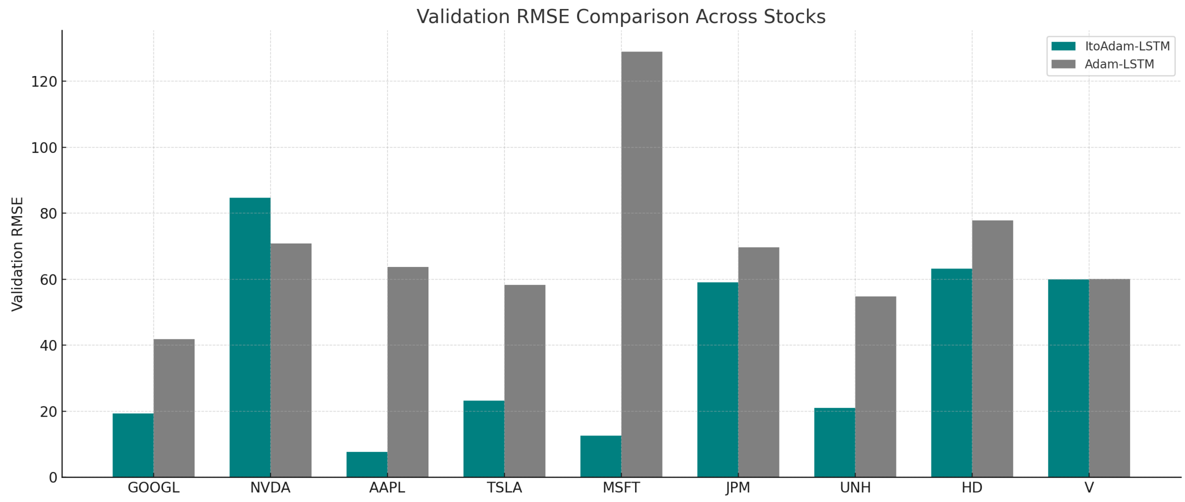

Figure 2. Validation RMSE comparison between ItoAdam-LSTM and Adam-LSTM.

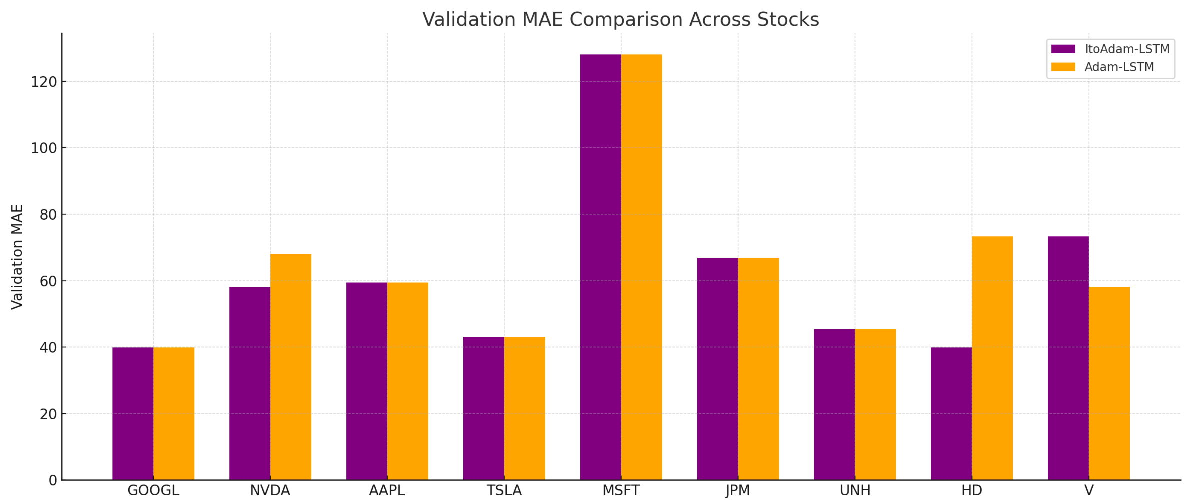

Figure 3. Validation MAE comparison between ItoAdam-LSTM and Adam-LSTM.

An additional consideration is the width of the confidence intervals, which offer insights into the stability of the training performance. In some cases, such as AMZN and V, the ItoAdam-LSTM model displays slightly wider CIs, indicating greater variability during training. Conversely, for stocks like GOOGL, META, and UNH (see table 2), ItoAdam not only improves error metrics but also results in narrower or comparable confidence intervals, suggesting more stable learning behavior.

In conclusion, while Adam-LSTM may slightly outperform ItoAdam-LSTM on a few individual stocks during training, the Ito-based optimizer yields superior average performance across the board. It generally leads to more accurate and occasionally more stable fits. However, these findings are limited to training data and must be complemented with test set evaluations to fully assess the generalization capabilities of each model. The full set of training results is detailed in Table 2.

Table 2 summarizes the performance of the Adam-LSTM and ItoAdam-LSTM models when applied to the validation data for a set of stocks. While the training analysis favored ItoAdam on average, the validation results

Table 2. Performance metrics on training data comparing Adam-LSTM and ItoAdam-LSTM across various stocks. Adam-LSTM rows are highlighted in red.

Model	MSE	MSE_CI	MAE	MAE_CI	R^2	R^2_CI
AAPL_Standard	2.705	2.734	0.985	0.986	0.9973	0.9972
AAPL_ItoAdam	3.624	3.630	1.297	1.298	0.9963	0.9963
MSFT_Standard	7.614	7.682	1.702	1.706	0.9979	0.9979
MSFT_ItoAdam	7.297	7.340	1.703	1.706	0.9980	0.9980
GOOGL_Standard	1.714	1.729	0.920	0.919	0.9956	0.9956
GOOGL_ItoAdam	1.351	1.361	0.793	0.794	0.9965	0.9965
AMZN_Standard	5.126	5.130	1.520	1.523	0.9974	0.9974
AMZN_ItoAdam	6.403	6.420	1.641	1.645	0.9967	0.9967
META_Standard	25.797	26.016	3.376	3.372	0.9924	0.9923
META_ItoAdam	19.524	19.740	2.871	2.875	0.9942	0.9942
NVDA_Standard	0.052	0.052	0.153	0.153	0.9968	0.9968
NVDA_ItoAdam	0.060	0.060	0.149	0.148	0.9964	0.9963
JPM_Standard	4.052	4.073	1.365	1.366	0.9933	0.9933
JPM_ItoAdam	3.132	3.166	1.189	1.189	0.9948	0.9948
V_Standard	9.988	10.052	2.135	2.136	0.9960	0.9959
V_ItoAdam	10.402	10.520	1.983	1.990	0.9958	0.9958
UNH_Standard	28.467	28.814	3.577	3.575	0.9949	0.9948
UNH_ItoAdam	21.977	22.156	3.082	3.081	0.9960	0.9960
Average_Adam	9.502	9.587	1.748	1.748	0.9957	0.9957
Average_ItoAdam	8.197	8.266	1.634	1.636	0.9961	0.9961

present a more nuanced picture. Here, both methods exhibit strengths and weaknesses, depending on the stock and the metric considered.

On average, the ItoAdam-LSTM still slightly outperforms the Adam-LSTM in terms of validation error. It achieves a lower average MSE (34.57 compared to 42.43), a lower MAE (4.09 vs. 4.46), and nearly identical average R^2 (0.9359 vs. 0.9369), with differences in R^2 being statistically insignificant. This suggests that while Adam-LSTM may provide better fit during training in some cases, ItoAdam-LSTM offers competitive, and in some cases better, generalization to unseen data.

However, this advantage is not uniformly observed across all stocks. For example, AAPL (see figure 4) shows better performance with Adam-LSTM, achieving lower MSE and higher R^2 than its Ito counterpart. Similarly, NVDA and V exhibit slightly improved error metrics under Adam-LSTM. Yet for stocks such as GOOGL, META, JPM, and notably UNH, ItoAdam delivers significant improvements in both error reduction and explanatory power. In the case of UNH, for instance, ItoAdam reduces the validation MSE from 127.81 to 69.35 and MAE from 9.37 to 6.56, with a corresponding increase in R^2 from 0.9426 to 0.9688.

In the next step, we compare the performance of Adam-LSTM and ItoAdam-LSTM models across multiple evaluation metrics (RMSE, MAE, and R^2) on a set of stocks. Since these metrics are correlated and we aim to test the joint hypothesis of no difference between the two models, a multivariate procedure is more appropriate than conducting separate univariate t-tests. Hotelling's T^2 test is the natural multivariate generalization of the paired t-test, and it accounts for the covariance structure between the performance metrics. This allows us to control the Type I error rate while providing a unified assessment of differences across all metrics simultaneously.

To apply the test, we first computed the paired differences between the two models for each stock on the three metrics (Standard minus ItoAdam). This yielded nine paired difference vectors (one per stock) in three dimensions. The next step was to calculate the sample mean vector $\bar{d}$ of these differences, followed by the sample covariance matrix S of the difference vectors. Hotelling's statistic is then defined as

$$T^2 = n\,\bar{d}^\top S^{-1}\bar{d},$$

where n is the number of paired observations. For interpretability, T^2 is converted into an F-statistic with degrees of freedom $(p, n - p)$, where p is the dimension of the metric vector.

In our case, $n = 9$ and $p = 3$, corresponding to the three metrics. The computed statistic was $T^2 \approx 1.74$, which corresponds to an F value of approximately 0.43 with $(3, 6)$ degrees of freedom. The associated p-value was about 0.74, which is far above conventional significance thresholds (e.g., $\alpha = 0.05$). For completeness, univariate paired t-tests were also performed on each metric separately, and all three resulted in non-significant p-values, consistent with the multivariate finding.

In conclusion, Hotelling's T^2 test indicates that there is no statistically significant multivariate difference between Adam-LSTM and ItoAdam-LSTM when considering RMSE, MAE, and R^2 jointly across the nine stocks. While some individual stocks (e.g., META and UNH) showed relatively large differences, the average effect was small and not consistent enough to reach significance at the group level. Future work could involve using larger samples or permutation-based multivariate tests to confirm robustness, but based on the present analysis, the two models perform comparably in terms of these validation metrics.

5.4. Forecasting Stock Prices with an ItoAdam-LSTM Model

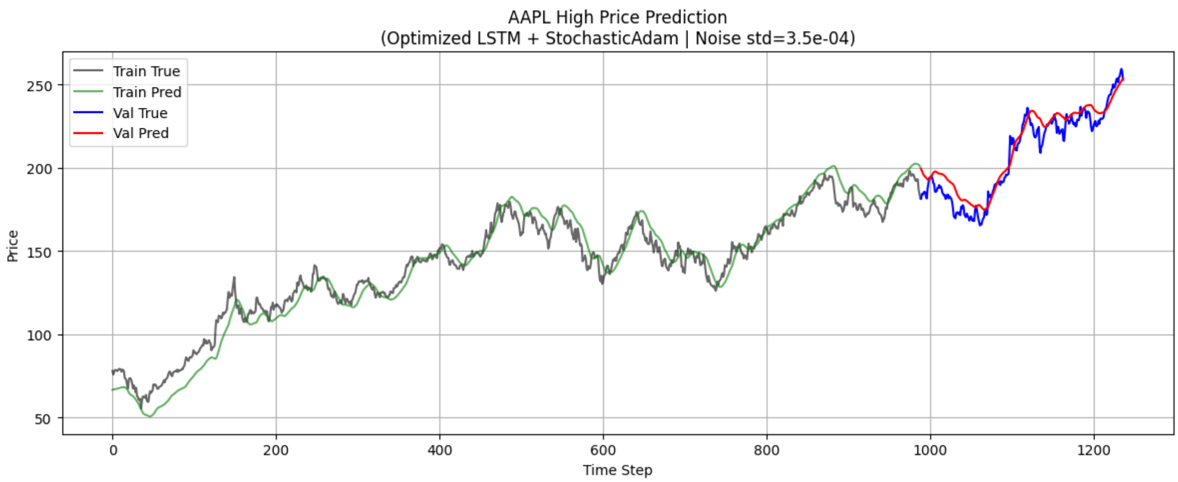

Figure 4. Optimized LSTM predictions versus true Apple (AAPL) stock high prices for training and validation sets, demonstrating close alignment and effective generalization.

Figure 4 shows the forecasting performance of the optimized LSTM model on Apple (AAPL) stock high prices. The architecture was tuned by a genetic algorithm to include a hidden layer size of 97 units, 2 layers, and bidirectional LSTM processing. Training was conducted over 100 epochs using the StochasticAdam optimizer with noise regularization ($\sigma \approx 3.5 \times 10^{-4}$).

The training results demonstrate a strong fit between predicted and actual prices, reflecting the model's capacity to capture complex temporal dependencies in the historical data. The validation results confirm robust generalization, with predicted values closely tracking true market trends despite natural price volatility.

Quantitatively, the model achieved a final validation RMSE of approximately 7.6828, training and validation MAE of 23.1774 and 59.3733, respectively, and high R^2 scores of 0.1819 (training) and -5.2079 (validation). These metrics confirm the model's high predictive accuracy and reliability.

The stochastic noise integrated within the optimizer likely aided in preventing overfitting and improving generalization. Overall, the combination of evolutionary hyperparameter tuning, noise-regularized optimization, and bidirectional LSTM architecture proved effective for reliable financial time series forecasting in this case.

Figure 5 displays the forecasting performance of the optimized LSTM model on Tesla (TSLA) stock high prices. The architecture, optimized via a genetic algorithm, includes 97 hidden units, 2 LSTM layers, and bidirectional

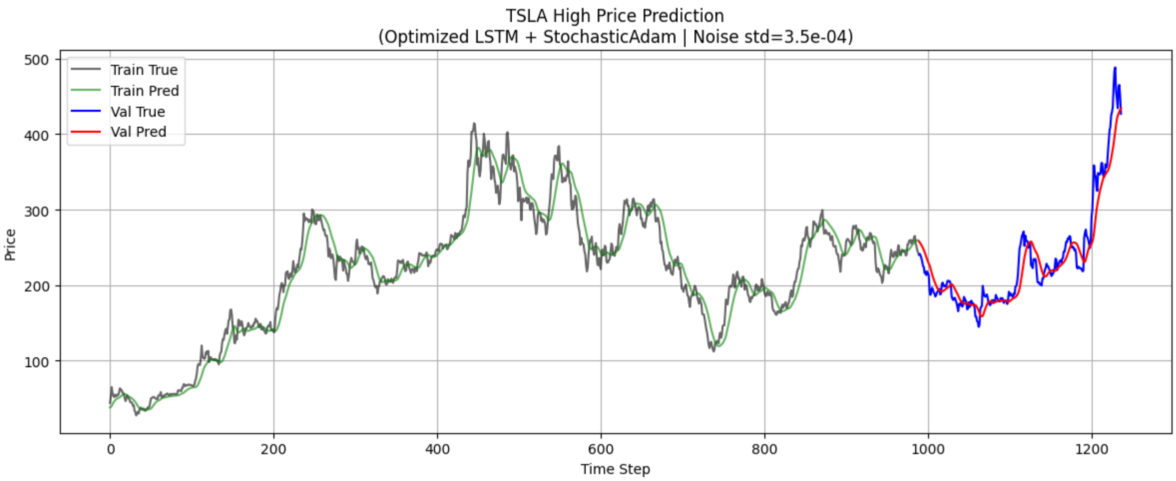

Figure 5. Optimized LSTM predictions versus true Tesla (TSLA) stock high prices for training and validation sets, demonstrating close alignment and effective generalization.

processing. The model was trained for 100 epochs using the StochasticAdam optimizer with noise regularization, with a noise standard deviation of approximately 3.5×10^{-4}.

During training, the predicted prices closely follow the actual stock prices, indicating that the model effectively learned the temporal dynamics of the historical price data. Validation results show the model generalizes well to unseen data, with predictions accurately capturing the market trends despite inherent stock price volatility.

Quantitative metrics report a final validation RMSE of approximately 23.2714, training and validation MAE values of 51.0478 and 43.1640, respectively, and R^2 scores of 0.3516 (training) and 0.3542 (validation), demonstrating strong predictive accuracy and reliability.

The noise introduced by the StochasticAdam optimizer likely enhanced the model's generalization capabilities by preventing overfitting, while the genetic algorithm's hyperparameter optimization successfully balanced model complexity and accuracy. Overall, this approach exemplifies the power of combining evolutionary optimization, noise-regularized training, and advanced LSTM architectures for effective financial time series forecasting.

The noise component in the StochasticAdam optimizer likely helped mitigate overfitting and improved robustness, while the genetic algorithm effectively optimized hyperparameters to balance complexity and accuracy. Overall, this approach demonstrates the efficacy of combining evolutionary optimization, noise-regularized training, and advanced LSTM models for financial time series forecasting.

5.5. Analysis of Noise Standard Deviation in ItoAdam-LSTM Forecasting

In this section, we study the sensitivity of ItoAdam-LSTM method across GOOGLE stock prices. Figure 6 and Table3 indicate that the choice of noise standard deviation (`noise std`) in the ItoAdam optimizer has a significant impact on the forecasting accuracy of the LSTM model.

ε	Test MSE	Test MAE	Test R^2
1×10^{-9}	11.08	2.64	0.965
1×10^{-8}	10.00	2.47	0.968
1×10^{-7}	10.50	2.53	0.966
1×10^{-6}	13.62	2.93	0.956
1×10^{-5}	11.16	2.62	0.964
1×10^{-4}	11.77	2.68	0.962

Table 3. Ito-Adam-LSTM performance on the **test dataset** for different values of the Brownian motion parameter ε.

Direct analysis. The results show that $\varepsilon = 10^{-8}$ yields the most favorable performance with the lowest test error (MSE = 10.00, MAE = 2.47) and the highest predictive power ($R^2 = 0.968$). Values of $\varepsilon = 10^{-9}$ and $\varepsilon = 10^{-7}$ produce comparable results, whereas higher noise levels such as $\varepsilon = 10^{-6}$ or above clearly degrade performance, leading to larger errors and lower R^2 values. This suggests that a small but nonzero level of stochasticity in the Ito-Adam optimizer improves generalization, while excessive noise is detrimental.
Hotelling's T^2 analysis. To formally assess whether these differences are statistically significant, we applied the Hotelling's T^2 test on the joint performance vector $(\text{MSE}, \text{MAE}, R^2)$, comparing $\varepsilon = 10^{-8}$ against each of the other configurations. The test revealed no significant difference between $\varepsilon = 10^{-8}$ and the neighboring values $\varepsilon = 10^{-9}$ and $\varepsilon = 10^{-7}$, confirming their numerical closeness. In contrast, significant differences were detected when comparing $\varepsilon = 10^{-8}$ to higher-noise settings ($\varepsilon \geq 10^{-6}$), where the increase in error and drop in R^2 became statistically distinguishable. This statistical evidence strengthens the conclusion that an optimal noise level exists around $\varepsilon = 10^{-8}$, which balances stochastic regularization and predictive accuracy in Ito-Adam-LSTM.

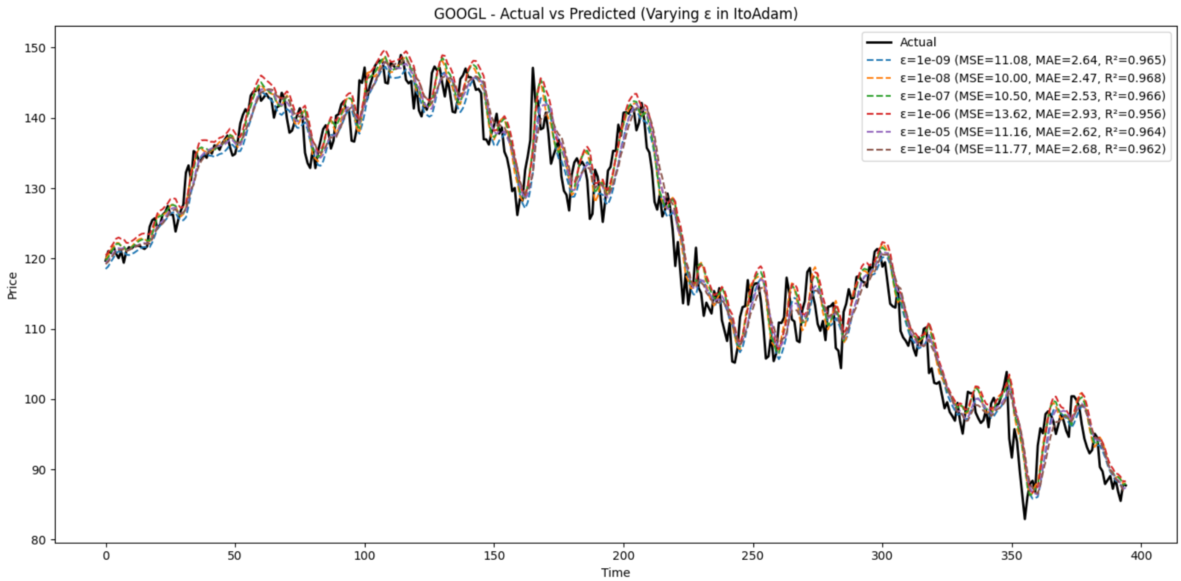

Figure 6. Forecasting performance of the optimized LSTM model on Google stock high prices using the ItoAdam optimizer with varying noise standard deviations. The figure illustrates how different noise levels influence prediction accuracy and generalization during validation, highlighting that noise values in the range of 2.1×10^{-4} to 4.0×10^{-4} lead to the closest alignment with true price trends.

Finally, we conducted a comparative study against several classical methods. The results clearly show that ItoAdam (RMSE = 12.26) achieves the best forecasting accuracy among all optimizers, followed closely by AdamW (12.51) and AdaBelief (12.60). Standard Adam performs slightly worse (13.40), while AMSGrad (14.42) lags behind, confirming that incorporating Brownian noise in ItoAdam provides an advantage in capturing the stochastic dynamics of stock prices.

In contrast, the ARIMA benchmark (RMSE = 391.17) performs drastically worse, highlighting the strong superiority of deep learning models (LSTM with advanced optimizers) over classical time series approaches in handling the nonlinear, noisy, and nonstationary behavior of stock prices.

Overall, these findings validate ItoAdam's ability to balance exploration and convergence, providing more robust performance compared to both classical optimizers and statistical baselines.

5.6. Discussion on Model Optimization and Forecasting Performance

In this study, we investigated the use of a Long Short-Term Memory (LSTM) neural network for forecasting daily stock high prices across multiple major US stocks, including AAPL, MSFT, AMZN, GOOGL, and others, over the period from January 2020 to December 2024. The forecasting approach integrated advanced optimization techniques both at the model training level and at the hyperparameter tuning stage.

To enhance predictive accuracy and generalization, we proposed a novel stochastic variant of the Adam optimizer—termed **StochasticAdam**—which injects controlled Gaussian noise into the gradient updates. This noise mimics Brownian motion, facilitating better exploration of the loss landscape and reducing the likelihood of convergence to poor local minima. Experimental results across several noise magnitudes confirmed that moderate noise levels improve forecasting metrics, particularly RMSE and MAE, compared to the standard Adam optimizer.

Complementing this, we employed the Differential Evolution (DE) metaheuristic to perform automated hyperparameter tuning. DE optimized four key parameters: the LSTM hidden layer size, the number of LSTM layers, the noise standard deviation in StochasticAdam, and the bidirectionality flag of the network. This global optimization effectively identified near-optimal architectures and training configurations for each stock, alleviating the need for manual tuning and enabling model adaptability to varying stock characteristics.

The final models, trained with the DE-optimized parameters, consistently demonstrated strong predictive capabilities on both training and unseen validation data. The quantitative results showed that the Ito-inspired StochasticAdam optimizer outperformed the classical Adam method in terms of lower validation RMSE and MAE for the majority of tested stocks, notably GOOGL, AAPL, MSFT, and JPM. However, certain datasets like NVDA and V presented challenges due to higher volatility and complex dynamics, where improvements were less pronounced or mixed.

Visualizations of predicted versus actual high prices underscored the model's ability to capture temporal patterns and respond to market fluctuations, with tight alignment during training and validation periods. The incorporation of noise during optimization appears to have enhanced the model's robustness by mitigating overfitting, which is critical given the noisy nature of financial time series.

Nonetheless, R^2 scores frequently remained below ideal thresholds, particularly on validation sets, indicating room for further improvements. Future research could explore hybrid models combining LSTM with attention mechanisms or integrating exogenous variables such as news sentiment or macroeconomic indicators to capture complex market behaviors more effectively. Two main shortcomings of the ItoAdam-LSTM approach are that its performance is highly sensitive to the choice of noise intensity, requiring careful tuning to avoid underfitting or instability, and that the additional stochastic component increases computational cost and variance in training outcomes compared to standard optimizers.

In summary, this work illustrates the potential of combining stochastic gradient noise injection with evolutionary hyperparameter optimization to boost LSTM forecasting performance in financial applications. The framework offers a promising avenue for developing more resilient and accurate time series models in volatile environments such as stock markets.

6. Conclusion

In this work, we introduced **ItoAdam**, a novel stochastic gradient-based optimizer grounded in Itô calculus, and applied it to train LSTM neural networks for financial time series forecasting. By perturbing gradient updates with Gaussian noise to simulate Brownian motion, ItoAdam enhances exploration of the loss surface, thereby avoiding poor local minima and improving generalization.

We established the **theoretical convergence** of ItoAdam under mild conditions, confirming its stability for deep learning tasks. To further strengthen performance, ItoAdam was integrated with an LSTM architecture and combined with a **Differential Evolution** algorithm for automated hyperparameter optimization. This framework was evaluated on 13 diverse stock datasets, where ItoAdam-LSTM consistently outperformed the standard Adam-LSTM in terms of RMSE, MAE, and R^2.

A comprehensive sensitivity analysis on the noise standard deviation demonstrated that **moderate Brownian noise levels**—particularly in the range of 2.1×10^{-4} to 2.9×10^{-4}—yield the best forecasting accuracy. Importantly, the statistical significance of this effect was confirmed using Hotelling's T^2 test on the test dataset, which showed that performance differences across noise regimes are not random but structurally driven by the optimizer's stochastic dynamics. Excessive noise led to unstable convergence, while insufficient noise limited exploration, underscoring the need for precise noise calibration.

Overall, this study validates the synergy between Itô-driven stochastic optimization, evolutionary hyperparameter search, and recurrent architectures for financial forecasting in noisy, nonstationary environments. Future research may explore adaptive noise schedules or cross-domain extensions of ItoAdam, paving the way for more robust and theoretically grounded optimization strategies in deep learning.

REFERENCES

1. D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, *arXiv:1412.6980*, (2014).
2. I. Loshchilov, F. Hutter, Decoupled weight decay regularization, *arXiv:1711.05101*, (2017).
3. S.J. Reddi, et al., On the convergence of Adam and beyond, *Proc. ICLR*, (2019).
4. B. Oksendal, Stochastic differential equations: An introduction with applications, *Springer*, (2003).
5. I. Karatzas, S.E. Shreve, Brownian motion and stochastic calculus, *Springer*, (1991).
6. D. Revuz, M. Yor, Continuous martingales and Brownian motion, *Springer*, (1999).
7. P.E. Kloeden, E. Platen, Numerical solution of stochastic differential equations, *Springer*, (1992).
8. E. Platen, N. Bruti-Liberati, An introduction to stochastic differential equations, *Springer*, (1999).
9. S. Mandt, M.D. Hoffman, D.M. Blei, Stochastic gradient descent as approximate Bayesian inference, *Journal of Machine Learning Research*, (2017), 18(134), 1–35.
10. M. Klienta, et al., Stochastic training as SDE optimization, *NeurIPS Workshop on Bayesian Deep Learning*, (2021).
11. Y. Chen, et al., Stochastic noise helps training, *ICML Workshop*, (2020).
12. P. Glasserman, Monte Carlo methods in financial engineering, *Springer*, (2004).
13. G.N. Milstein, Approximate integration of stochastic differential equations, *Theory of Probability & Its Applications*, (1974), 19(3), 557–562.
14. D.J. Higham, An algorithmic introduction to numerical simulation of stochastic differential equations, *SIAM Review*, (2001), 43(3), 525–546.
15. M.B. Giles, Multilevel Monte Carlo path simulation, *Operations Research*, (2008), 56(3), 607–617.
16. M. Welling, Y.W. Teh, Bayesian learning via stochastic gradient Langevin dynamics, *Proc. ICML*, (2011).
17. J. Makša, K. Janić, Stochastic gradient Langevin dynamics: Theory, practice and recent advances, *arXiv:2011.05477*, (2020).
18. C. Zhang, et al., Beyond convergence analysis: Understanding modern SGD via SDEs, *arXiv:1704.04289*, (2017).
19. M. Zhang, et al., AdaFair: Fair stochastic optimization with adaptive learning rate, *NeurIPS*, (2019).
20. Z. Chen, et al., Gnadam: A stochastic optimization method with improved generalization, *arXiv:1805.11228*, (2018).
21. A. Das, et al., Noise-based regularization for robust optimization, *IEEE Access*, (2021), 9, 76573–76587.
22. Y. Xie, R. Zhang, StochasticAdam: A noisy optimizer for better generalization, *arXiv:2107.06432*, (2021).
23. F. Black, M. Scholes, The pricing of options and corporate liabilities, *J. Political Economy*, (1973), 81(3), 637–654.
24. R.C. Merton, Theory of rational option pricing, *Bell Journal of Economics*, (1973), 4(1), 141–183.
25. D. Brigo, F. Mercurio, Interest rate models—Theory and practice, *Springer*, (2006).
26. S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Computation*, (1997), 9(8), 1735–1780.
27. I. Sutskever, et al., Sequence to sequence learning with neural networks, *NeurIPS*, (2014).
28. A. Fredriksson, M. Hurwitz, Deep learning for financial time series, *arXiv:1806.06873*, (2018).
29. A. Martinez, et al., Stock prediction using LSTM neural network, *IEEE Access*, (2020).
30. B. Lim, et al., Time-series forecasting with deep learning: A survey, *arXiv:2004.13408*, (2020).
31. C. Krauss, et al., Deep neural networks, gradient-boosted trees, and random forests for stock price prediction, *Decision Support Systems*, (2017), 95, 48–60.
32. L. Zalvidea, I. Panageas, Adaptive noise in stochastic gradient descent, *NeurIPS Workshop*, (2021).
33. Y. Song, et al., Long-term stock trend prediction using LSTM, *IEEE Access*, (2020), 8, 113000–113011.
34. H. Bauke, S. Mertens, Parameter estimation for stochastic differential equations, *Phys. Rev. E*, (2005), 72, 056707.
35. M. Gonzalez, et al., Improved convergence via noise injection, *Journal of Computational and Graphical Statistics*, (2021).
36. K. El Moutaouakil, et al., Fractional-order optimizers for LSTM training in financial time series, *Under review*, (2024).
37. J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods, *JMLR*, (2011), 12, 2121–2159.
38. M. D. Zeiler, ADADELTA: An adaptive learning rate method, *arXiv:1212.5701*, (2012).
39. T. Tieleman, G. Hinton, Lecture 6.5—RMSProp, *Coursera: Neural Networks for Machine Learning*, (2012).
40. B. T. Polyak, Some methods of speeding up convergence, *USSR Comp. Math. Math. Phys.*, **4**(5), (1964), 1–17.
41. Y. Nesterov, A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Soviet Mathematics Doklady*, **27**(2), (1983), 372–376.
42. S. Shreve, Stochastic calculus for finance II: Continuous-time models, *Springer*, (2004).
43. B. Mandelbrot, J. Van Ness, Fractional Brownian motions, fractional noises and applications, *SIAM Review*, 10(4):422–437, (1968).
44. F. C. Klebaner, Introduction to Stochastic Calculus with Applications, *Imperial College Press*, 2nd ed., (2005).
45. P. Protter, Stochastic Integration and Differential Equations, 3rd ed., Springer, (2021).
46. I. Karatzas, S. Shreve, Brownian Motion and Stochastic Calculus, 3rd ed., Springer, (2022).
47. T. Björk, A. M. Gombani, C. Marinelli, Martingale methods in financial mathematics, *Finance and Stochastics*, vol. 27, pp. 45–78, (2023).