Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 10, pp. 3585–3595
DOI : 10.47974/JIOS-2486

VAdam : A volatility-aware adaptive gradient optimization framework for financial time-series learning

Aditya Nagdiya *
Vivek Kapoor †
Vrinda Tokekar §
Department of Information Technology
Institute of Engineering & Technology
Devi Ahilya Vishwavidyalaya
Indore 452017
Madhya Pradesh
India

Abstract

Financial time-series forecasting is difficult because of constantly changing market conditions, non-stationary behavior, and sudden changes in volatility which can negatively affect optimization stability during deep neural network training. Existing adaptive optimization methods such as Adam and its variants employ fixed optimization dynamics that do not directly respond to changing market conditions, which may lead to inefficient learning under varying volatility regimes. This study proposes VAdam, a Volatility-Aware Adaptive Gradient Optimization Framework for financial time-series learning. The proposed method builds on the conventional Adam optimizer by incorporating a volatility-conditioned update mechanism that dynamically adjusts learning rate and momentum behavior according to normalized market volatility. The method was evaluated on real cryptocurrency datasets obtained from Yahoo Finance, including Bitcoin, Ethereum, Solana, XRP, and Cardano, using a controlled optimizer-isolation benchmark with a shared MLP forecasting model. A comparison with SGD, RMSProp, Adam, AdamW, and Nadam demonstrates that VAdam achieves modest but consistent improvements over standard Adam, including 0.65% lower mean MSE, 0.67% lower price sMAPE, and 1.28% fewer epochs to convergence. The largest improvement was observed for XRP, where VAdam reduced MSE by 4.65% relative to Adam. The findings suggest that volatility-aware optimization can improve Adam-family

* *E-mail:* **aditya22.ad@gmail.com (Corresponding Author)**
† *E-mail:* **vkapoor@ietdavv.edu.in**
§ *E-mail:* **vrindatokekar@yahoo.com**

© Taru Publications

optimization behavior while maintaining a simple drop-in optimization framework for financial forecasting tasks.

Subject Classification: *Primary 93A30, Secondary 49K15.*

Keywords: *Financial time-series forecasting, Adaptive optimization, Volatility-aware learning, Deep learning, Adam optimizer, Cryptocurrency forecasting, Gradient-based optimization, Neural networks.*

1. Introduction

Financial time-series forecasting is an important problem in quantitative finance with applications in algorithmic trading, portfolio management, and risk assessment. In recent times, cryptocurrency markets have attracted significant attention because of their high volatility, nonstationary behavior, and rapidly changing market dynamics. These features make accurate forecasting particularly challenging and create greater demands on the optimization algorithms used to train deep learning models.

Traditional volatility modeling approaches such as ARCH, GARCH, and HAR-RV have been widely used to characterize financial market fluctuations [1–3]. More recently, deep learning architectures including recurrent neural networks, Long Short-Term Memory (LSTM) networks, and transformer-based models have shown strong forecasting performance by capturing complex nonlinear temporal dependencies in financial data [4–6]. Several studies have successfully applied deep learning techniques to cryptocurrency forecasting tasks [7–10].

While substantial research has focused on improving forecasting architectures, less attention has been paid to the role of optimization algorithms. Optimization methods strongly affect convergence speed, training stability, and final predictive performance. Adaptive gradient methods such as Adam, AdamW, Nadam, AdaBound, AdaBelief, and RAdam are now widely used because of their ability to automatically adjust parameter updates during training [11–16]. However, these methods primarily rely on gradient statistics and typically employ fixed optimization dynamics throughout the learning process.

This assumption may be limiting in financial forecasting environments where market conditions continuously evolve. During periods of elevated volatility, aggressive parameter updates may destabilize training, whereas stable market conditions may permit faster and more responsive learning. Existing adaptive optimizers generally do not explicitly incorporate market volatility into the optimization process, potentially limiting their effectiveness under nonstationary financial regimes. Furthermore, recent studies have highlighted convergence limitations of adaptive optimization methods and emphasized the importance of dynamically adapting optimization behavior under changing learning environments [17–19].

To overcome this limitation, this paper proposes VAdam, a Volatility-Aware Adaptive Gradient Optimization Framework for financial time-series learning. The proposed method extends the conventional Adam optimizer by introducing volatility-conditioned learning-rate and momentum adaptation mechanisms. Instead of relying solely on gradient information, VAdam incorporates normalized market volatility directly into parameter-update dynamics, enabling the optimizer to respond to changing market regimes during training.

The proposed framework is evaluated on five cryptocurrency datasets obtained from Yahoo Finance, namely Bitcoin (BTC), Ethereum (ETH), Solana (SOL), XRP, and Cardano (ADA). Comparative experiments are performed against SGD, RMSProp, Adam, AdamW, and Nadam using a controlled optimizer-isolation benchmark with an identical forecasting architecture across all methods.

The main contributions of this study are as follows:

- A volatility-aware extension of the Adam optimizer is proposed for financial time-series forecasting.
- Dynamic learning-rate and momentum adaptation mechanisms are introduced using normalized market volatility.
- Extensive experiments are conducted across multiple cryptocurrency datasets under identical training conditions.
- Results demonstrate that VAdam achieves consistent improvements over standard Adam-family optimizers while maintaining computational simplicity and stable convergence behavior.

2. Related Work

Financial forecasting has traditionally relied on statistical volatility models such as ARCH and GARCH, which estimate time-varying variance in financial markets [1, 2]. Extensions such as the HAR-RV model further improved volatility forecasting by incorporating information across multiple temporal horizons [3]. Although these methods remain widely used in financial econometrics, their ability to model highly nonlinear and rapidly changing cryptocurrency markets is limited.

Recent advances in deep learning have significantly improved financial time-series forecasting performance. Recurrent architectures such as LSTM and GRU networks have demonstrated strong capability for capturing temporal dependencies in sequential financial data. More recently, transformer-based architectures and attention mechanisms have achieved state-of-the-art performance across a wide range of forecasting tasks [5, 6]. Several studies have successfully applied deep learning techniques to cryptocurrency forecasting, demonstrating the effectiveness of neural models in capturing complex market dynamics [7–10].

In parallel, substantial research has focused on improving optimization algorithms for deep neural networks. Adam remains one of the most widely

adopted adaptive optimization methods because it combines momentum estimation with adaptive learning-rate adjustment [11]. Numerous extensions have subsequently been proposed, including Nadam [12], AdamW [13], AdaBound [14], AdaBelief [15], and RAdam [16]. These methods improve convergence behavior through mechanisms such as momentum modification, weight-decay decoupling, bounded learning rates, variance rectification, and adaptive gradient estimation.

Despite their effectiveness, existing adaptive optimizers primarily rely on gradient statistics and generally employ fixed optimization dynamics throughout training. Recent studies have highlighted convergence limitations of adaptive methods and emphasized the importance of dynamically adapting optimization behavior under changing learning conditions [17–19]. However, relatively limited research has explored optimization frameworks that explicitly incorporate financial market volatility into the learning process.

Therefore, a research gap exists between volatility modeling in financial forecasting and adaptive optimization in deep learning. To address this gap, this work proposes VAdam, a volatility-aware adaptive optimization framework that directly integrates normalized market volatility into learning-rate and momentum adaptation mechanisms for financial time-series learning.

3. Proposed Methodology

This section describes the proposed Volatility-Aware Adaptive Gradient Optimization (VAdam) framework for financial time-series learning. VAdam is designed to improve optimization stability under changing market conditions by incorporating financial volatility directly into the parameter-update process.

3.1 *Framework Overview*

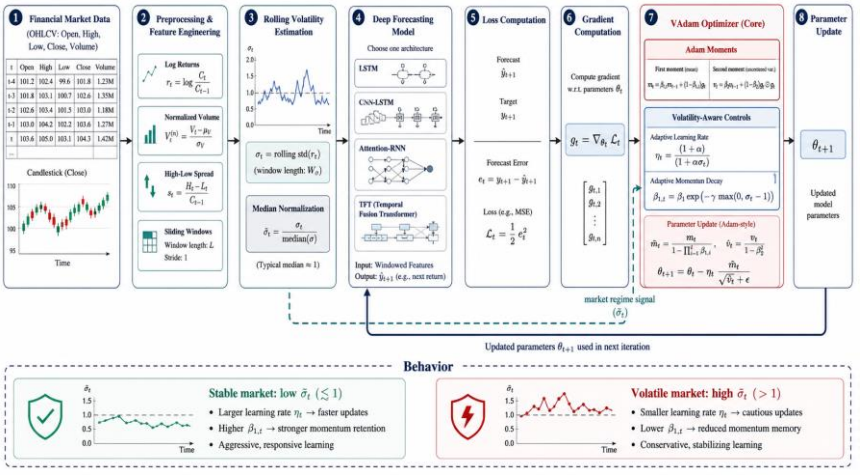

Figure 1
Overall VAdam framework architecture

The overall VAdam framework is shown in Figure 1. The framework consists of three main stages: (i) financial feature generation, (ii) rolling volatility estimation, and (iii) volatility-aware optimization during neural network training.

Financial market data consisting of Open, High, Low, Close, and Volume (OHLCV) observations are transformed into forecasting features including logarithmic returns, volatility measures, and normalized market indicators. These features are then supplied to the forecasting model.

3.2 *Volatility Estimation*

To capture changes in market conditions, rolling volatility is estimated using logarithmic returns:

$$r_t = \ln\left(\frac{P_t}{P_{t-1}}\right) \tag{1}$$

Rolling volatility over a window of length w is computed as:

$$\sigma_t = \sqrt{\frac{1}{w}\sum_{i=t-w+1}^{t}(r_i - \bar{r})^2} \tag{2}$$

where $\bar{r}$ denotes the mean return within the rolling window.

To obtain a scale-independent volatility signal, normalization is performed using:

$$\bar{\sigma}_t = \frac{\sigma_t}{\text{median}(\sigma)} \tag{3}$$

The normalized volatility value acts as a market-regime indicator and is used to regulate optimization behavior.

3.3 *VAdam Optimization Framework*

The proposed VAdam optimizer builds on Adam by introducing volatility-conditioned adaptation into both learning-rate and momentum dynamics. The complete update process is shown in Figure 2.

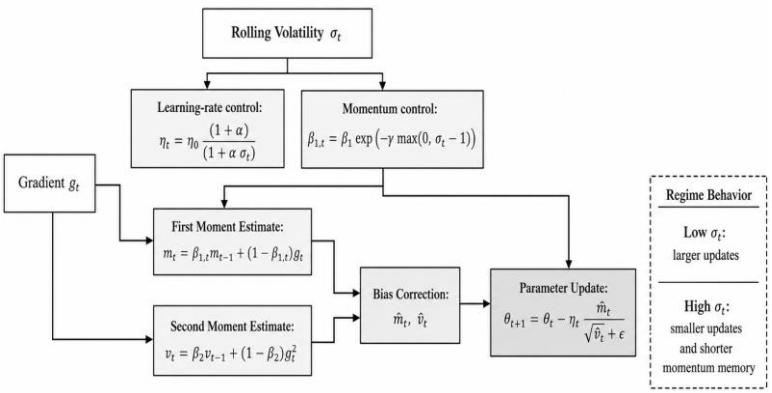

Figure 2
Volatility-aware update mechanism of the proposed VAdam optimizer

Dynamic Learning-Rate Adaptation: the learning rate is adjusted according to normalized market volatility:

$$\eta_t = \eta_0 \frac{(1+\alpha)}{(1+\alpha\bar{\sigma}_t)} \tag{4}$$

where η_0 denotes the initial learning rate and α controls volatility sensitivity.

Dynamic Momentum Adaptation:The first-moment coefficient is modified as:

$$\beta_{1,t} = \beta_1 \exp(-\gamma \max(0, \bar{\sigma}_t - 1)) \tag{5}$$

where γ controls momentum sensitivity to volatility.

Parameter Update: Adam moment estimates are computed as:

$$m_t = \beta_{1,t} m_{t-1} + (1 - \beta_{1,t}) g_t \tag{6}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \tag{7}$$

with bias correction:

$$\widehat{m}_t = \frac{m_t}{1-\beta_{1,t}^t} \tag{8}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{9}$$

The final VAdam update rule is defined as:

$$\theta_{t+1} = \theta_t - \eta_t \frac{\widehat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{10}$$

By adjusting both learning rate and momentum according to market volatility, VAdam enables more conservative updates during highly volatile periods while maintaining efficient learning under stable market conditions.

4. Experimental Setup

The experiments used daily cryptocurrency market data obtained from Yahoo Finance, including Bitcoin (BTC), Ethereum (ETH), Solana (SOL), XRP, and Cardano (ADA). The datasets contained between 2,238 and 3,068 daily observations and were used for next-step return forecasting. Each dataset contains Open, High, Low, Close, and Volume (OHLCV) observations. For all datasets, logarithmic returns, rolling volatility, normalized volume, and spread-based features were generated. Features were normalized using Min-Max scaling prior to model training.

To maintain the time order and prevent information leakage, each dataset was chronologically divided into 70% training, 15% validation, and 15% testing subsets.

The main purpose of this study is to evaluate optimization behavior rather than forecasting architectures. Therefore, all experiments employed the same MLP-64 forecasting model consisting of a single hidden layer with 64 neurons, ReLU activation, dropout regularization, and a linear output layer.Using the same architecture in every experiment ensures that performance differences originate from the optimization algorithms rather than model design.The proposed VAdam optimizer was compared against SGD, RMSProp, Adam [11],

AdamW [13], and Nadam [12].All optimizers were evaluated under identical initialization, training data, model architecture, and hyperparameter settings.Forecasting performance was evaluated using Mean Squared Error (MSE), Mean Absolute Error (MAE), Symmetric Mean Absolute Percentage Error (sMAPE), and Directional Accuracy (DA). Optimization behavior was assessed using epochs-to-convergence and validation-loss stability.

Training stability was measured using validation-loss variance, where lower values indicate smoother convergence behavior. Experiments were implemented in Python using PyTorch with a learning rate of 0.001, batch size of 32, training horizon of 100 epochs, 64 hidden units, ReLU activation, dropout rate of 0.20, and MSE loss. For VAdam, three additional parameters were introduced: volatility sensitivity (α), momentum sensitivity (γ), and rolling volatility window length (w). These parameters were selected using validation-set tuning.

5. Results and Discussion

This section examines the effectiveness of the proposed VAdam framework and compares its performance against conventional optimization algorithms across multiple cryptocurrency forecasting tasks.

Table 1 summarizes the average forecasting and optimization performance of all evaluated optimizers across the five cryptocurrency datasets.

Table 1

Average Performance Comparison Across Optimizers

Optimizer	MSE	MAE	Price sMAPE (%)	Directional Accuracy (%)	Epochs	Stability
SGD	0.001656	0.0277	2.7693	51.49	17.8	0.000069
RMSProp	0.002455	0.0362	3.6188	49.50	26.6	0.007279
Adam	0.002952	0.0396	3.9613	50.79	15.6	0.000367
AdamW	0.002951	0.0396	3.9607	50.79	15.6	0.000367
Nadam	0.003080	0.0401	4.0045	51.75	14.4	0.000501
VAdam	0.002933	0.0394	3.9347	50.22	15.4	0.000373

The findings suggest that VAdam achieves modest but consistent improvements over Adam and AdamW in terms of MSE, MAE, and sMAPE while maintaining comparable convergence speed. Compared with Adam, VAdam achieves lower forecasting error through volatility-aware adaptation of learning-rate and momentum dynamics.

Figure 3 further illustrate the comparative forecasting performance across optimization methods.

Although SGD achieved the strongest overall forecasting accuracy under the selected MLP architecture, VAdam consistently improved upon Adam-family optimizers, demonstrating the effectiveness of incorporating volatility information directly into the optimization process.

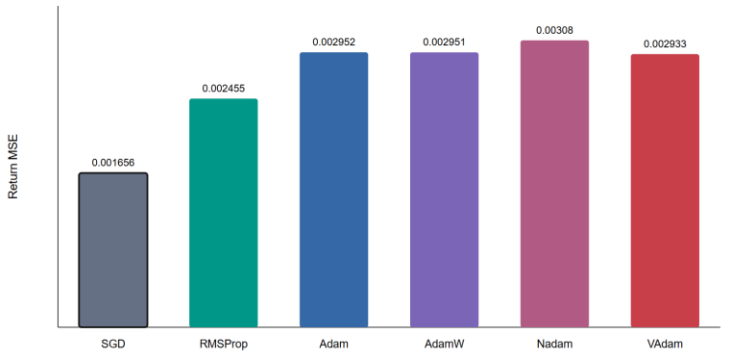

Figure 3
Average MSE across optimization methods

Beyond forecasting accuracy, optimization stability and convergence speed were evaluated using epochs-to-convergence and validation-loss behavior.

The convergence results indicate that VAdam converges slightly faster than Adam and AdamW while maintaining stable optimization behavior. In contrast, RMSProp required significantly more training epochs and exhibited larger fluctuations during optimization.

To further analyze training dynamics, the validation-loss convergence curves for Bitcoin forecasting are shown in Figure 4.

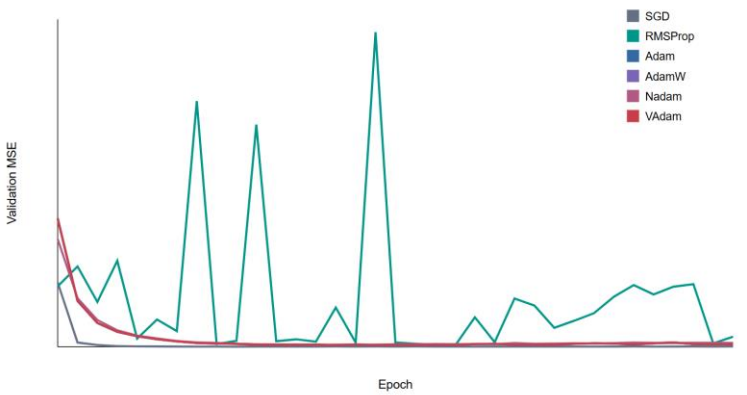

Figure 4
Validation-loss convergence behavior for Bitcoin forecasting

The smoother convergence trajectory observed for VAdam suggests that volatility-aware adaptation helps reduce unstable parameter updates during training, thereby improving optimization robustness under changing market conditions.

To evaluate robustness across different market environments, asset-level performance was analyzed using the average results shown in Table 2.

Table 2
Asset-Level Performance Summary

Asset	MSE	MAE	Price sMAPE (%)	Directional Accuracy (%)	Epochs
ADA	0.003898	0.0441	4.3998	48.57	20.83
BTC	0.000919	0.0226	2.2572	51.54	18.00
ETH	0.002228	0.0353	3.5295	53.84	11.33
SOL	0.003659	0.0460	4.5962	49.25	19.50
XRP	0.002651	0.0376	3.7586	50.58	18.17

The findings show considerable variation across assets because of differing volatility characteristics and market dynamics. Bitcoin achieved the lowest forecasting error, while Solana and ADA exhibited higher prediction errors due to their stronger volatility and less stable market behavior.

Importantly, the improvement of VAdam was most evident for XRP and other assets exhibiting larger volatility fluctuations. This observation supports the central hypothesis of this work that volatility-aware optimization becomes increasingly beneficial under no stationary market conditions.

Overall, the findings show that incorporating market volatility directly into optimization dynamics can improve Adam-family optimization behavior while preserving computational simplicity and stable convergence characteristics.

6. Conclusion and Future Work

This study presented VAdam, a Volatility-Aware Adaptive Gradient Optimization Framework for financial time-series learning. The proposed method extends the conventional Adam optimizer by incorporating normalized market volatility into learning-rate and momentum adaptation mechanisms. By dynamically adjusting optimization behavior according to prevailing market conditions, VAdam aims to improve training stability and forecasting performance under nonstationary financial environments. Experiments conducted on five cryptocurrency datasets, namely Bitcoin, Ethereum, Solana, XRP, and Cardano, demonstrated that VAdam achieved modest but consistent improvements over Adam and AdamW across multiple forecasting metrics. The results further indicated that the benefits of volatility-aware optimization become more apparent under assets exhibiting stronger volatility fluctuations, supporting the effectiveness of incorporating market information directly into optimization dynamics. Although the observed improvements were moderate, the findings suggest that domain-aware optimization represents a promising direction for financial forecasting applications. Future studies will examine the application of VAdam to more advanced forecasting architectures, including LSTM, GRU, CNN-LSTM, and transformer-based models [5, 6]. Additional

research may also explore alternative volatility estimation techniques, multi-horizon forecasting tasks, and broader financial markets to further evaluate the robustness and generalizability of the proposed framework.

References

[1] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007 (1982), doi: 10.2307/1912773.

[2] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *J. Econometrics*, vol. 31, no. 3, pp. 307–327 (1986), doi: 10.1016/0304-4076(86)90063-1.

[3] F. Corsi, "A simple approximate long-memory model of realized volatility," *J. Financ. Econometrics*, vol. 7, no. 2, pp. 174–196 (2009), doi: 10.1093/jjfinec/nbp001.

[4] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 379, no. 2194, Art. no. 20200209 (2021), doi: 10.1098/rsta.2020.0209.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 5998–6008 (2017).

[6] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *Int. J. Forecast.*, vol. 37, no. 4, pp. 1748–1764 (2021), doi: 10.1016/j.ijforecast.2021.03.012.

[7] M. M. Patel, S. Tanwar, R. Gupta, and N. Kumar, "A deep learning-based cryptocurrency price prediction scheme for financial institutions," *J. Inf. Secur. Appl.*, vol. 55, Art. no. 102583 (2020), doi: 10.1016/j.jisa.2020.102583.

[8] M. Saad, J. Choi, D. Nyang, J. Kim, and A. Mohaisen, "Toward characterizing blockchain-based cryptocurrencies for highly accurate predictions," *IEEE Syst. J.*, vol. 14, no. 1, pp. 321–332 (2020).

[9] R. Kaur, M. Uppal, D. Gupta, S. Juneja, S. Y. Arafat, J. Rashid, J. Kim, and R. Alroobaea, "Development of a cryptocurrency price prediction model: Leveraging GRU and LSTM for Bitcoin, Litecoin and Ethereum," *PeerJ Comput. Sci.*, vol. 11, Art. no. e714 (2025).

[10] W. Badar, S. Ramzan, A. Raza, N. L. Fitriyani, M. Syafrudin, and S. W. Lee, "Enhanced interpretable forecasting of cryptocurrency

prices using autoencoder features and a hybrid CNN-LSTM model," *Mathematics*, vol. 13, no. 12, Art. no. 1908 (2025), doi: 10.3390/math13121908.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.* (ICLR), (2015).

[12] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proc. Int. Conf. Learn. Represent. (ICLR) Workshop,* (2016).

[13] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. Int. Conf. Learn. Represent. (ICLR),* (2019).

[14] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," in *Proc. Int. Conf. Learn. Represent. (ICLR),* (2019).

[15] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan, "AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, pp. 18795–18806 (2020).

[16] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. Int. Conf. Learn. Represent. (ICLR),* (2020).

[17] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of Adam and beyond," in *Proc. Int. Conf. Learn. Represent. (ICLR),* (2018).

[18] A. Samanta and S. Williamson, "Adaptive learning guided by bias-noise-alignment diagnostics," *arXiv:2512.24445* (2025).

[19] D. Dhabliya, A. Lavhale, S. Thakur, R. M. Gomathi, A. Kumar, and S. De, "Enhancing cryptographic security through zero-knowledge proofs in theoretical mathematics," *J. Discrete Math. Sci. Cryptogr.*, vol. 29, no. 2-B, pp. 913–921 (2026), doi: 10.47974/JDMSC-2542.

Received March, 2026