Journal of Computer Science and Cybernetics, V.42, N.xx (2026), 1–
DOI no. 10.15625/1813-9663/24220

L-GRU: A NOVEL HYBRID AI FRAMEWORK FOR IMPROVING TIME SERIES FORECASTING

TRAN DINH DANG, SITTIXAY CHOUNLAMANY, NGUYEN QUOC DUNG*, PHAN THI GAM

Faculty of Engineering and Technology, Ha Tinh University, Cam Binh Commune, Ha Tinh Province, Viet Nam

Crossref Similarity Check
Powered by iThenticate

Abstract. Recurrent neural network (RNN) architectures such as LSTM and GRU have demonstrated remarkable effectiveness in modeling sequential data. However, LSTM possesses a complex structure with three distinct gating mechanisms, resulting in high computational cost, whereas GRU, though more streamlined, merges the hidden state and cell state, which may reduce its ability to manage long-term memory. This paper proposes the Long-term Gated Recurrent Unit (L-GRU), a novel hybrid architecture designed to overcome these limitations by integrating the strengths of both models. Experiments were conducted on four diverse datasets, including three financial datasets (Tesla stock, S&P 500 index, Bitcoin) and one meteorological dataset (Jena Climate) with an extended evaluation period of over 580 days to verify long-term stability. The results indicate that L-GRU achieves superior and stable performance across all datasets, outperforming traditional LSTM, GRU, and Transformer models, and particularly demonstrating strong generalization capability on both stochastic and physically structured data.

Keyword. Deep learning, GRU, LSTM, recurrent neural networks, time series forecasting.

1. INTRODUCTION

Time series forecasting is a critical yet challenging problem due to complex non-linear dependencies in highly volatile environments. While traditional statistical models struggle with non-stationarity, recurrent deep learning architectures like Long Short-Term Memory (LSTM) [1] and Gated Recurrent Unit (GRU) [2] have emerged as powerful alternatives [3]. However, LSTM's complex gating mechanism incurs high computational costs, whereas GRU's streamlined architecture merging the cell and hidden states [4] often compromises long-term memory retention. Furthermore, Transformer architectures, despite handling long-range dependencies well, introduce an $O(n^2)$ computational bottleneck [5, 6].

To address these limitations, this paper proposes the Long-term Gated Recurrent Unit (L-GRU), a novel hybrid architecture balancing LSTM's memory retention with GRU's computational efficiency. The main contributions are:

*Corresponding author.
E-mail addresses: dangitcntt55@gmail.com (T. D. Dang); sithixayjlmn@gmail.com (S. Chounlamany); dung.nguyenquoc@htu.edu.vn (N. Q. Dung); gam.phanthi@htu.edu.vn (P. T. Gam).

- **Novel Architecture:** L-GRU preserves an independent cell state for robust long-term memory while employing a streamlined, coupled update mechanism.
- **Structural Enhancements:** Integration of Peephole Connections, Decoupled Weight Matrices, and Layer Normalization to mitigate gradient degradation and accelerate convergence.
- **Comprehensive Evaluation:** Extensive empirical validation establishing L-GRU's superiority over standard LSTM, GRU, and state-of-the-art Transformers across multi-domain datasets.

The remainder of this paper is organized as follows: Section 2 presents a comprehensive review of existing literature, analyzing the evolution, strengths, and inherent limitations of fundamental sequential models, including LSTM [1], GRU [4], and advanced Transformer architectures [5, 7]. Section 3 thoroughly details the proposed L-GRU architecture, explaining the integration of its independent memory cell and decoupled gating mechanisms, alongside a theoretical analysis demonstrating its resistance to gradient degradation. Section IV provides an extensive experimental evaluation across diverse domains encompassing highly volatile financial markets and deterministic meteorological environments [8] and includes rigorous comparative analyses, ablation studies, and empirical assessments of practical financial profitability. Finally, Section 5 synthesizes the key contributions of this research and delineates potential directions for future work.

2. RELATED WORK

The evolution of time series forecasting has been heavily influenced by advancements in deep learning, particularly the development of architectures capable of handling sequential data. Early approaches relied on basic Recurrent Neural Networks (RNNs); however, their inability to bridge long temporal gaps due to gradient instability necessitated the creation of more sophisticated models [3].

Long Short-Term Memory (LSTM) networks [1] became the gold standard for many forecasting tasks. By introducing a separate cell state and a sophisticated gating system, LSTMs effectively trapped gradients in the cell, allowing the network to learn long-term dependencies. Despite its success, the large number of parameters in LSTM cells makes them computationally expensive to train. To address the computational bottleneck, the Gated Recurrent Unit (GRU) [2] was introduced. By merging the cell state and hidden state, and reducing the gating mechanism, GRU achieved comparable performance to LSTM on many tasks while requiring fewer parameters [4]. However, recent studies indicate that for certain applications requiring strictly long-term context retention, GRU's simplified memory management can lead to performance degradation compared to LSTM.

In recent years, Transformer architectures [5] have been adapted for time series forecasting. While efficient Transformer variants reduce the $O(n^2)$ attention cost, they still tend to be memory-intensive for long sequences and can be highly sensitive to noise in non-stationary financial data [9, 10]. Recently, specialized architectures such as Autoformer [7], PatchTST [11], and iTransformer [12] have advanced the state-of-the-art in long-term forecasting by introducing auto-correlation, patch-based learning, and channel-independence. However, these models often require substantial computational resources and massive datasets compared to streamlined RNNs.

To strike a balance, researchers have increasingly explored hybrid architectures. Some studies have stacked LSTM and GRU layers sequentially to capture different levels of temporal abstraction [13, 14], or integrated attention mechanisms [15] and graph structures [16]. However, these stacking and ensembling approaches typically increase the overall parameter count and computational overhead.

Unlike existing stacking-based hybrids, our proposed L-GRU model deeply integrates the structural advantages of LSTM and GRU at the foundational cell level. By preserving the isolated cell state of LSTM while adapting the efficient coupled gating of GRU, L-GRU offers a unified, parameter-efficient solution tailored for highly volatile time series forecasting. Furthermore, various hybrid optimization schemes [17] and variation-based analysis frameworks [18] for time series forecasting have also been extensively evaluated within the JCC community to improve prediction accuracy.

3. PROPOSED HYBRID L-GRU AI ARCHITECTURE

This section summarizes the L-GRU architecture and experiments. Table 1 presents the complete formulation.

Table 1: Formulas of the Hybrid L-GRU Architecture

Component Name	Notation	Formula
Relevance Gate	r_t	$\sigma(\mathrm{LN}(W_{rh}h_{t-1} + W_{rx}x_t + W_{cr} \odot C_{t-1} + b_r))$
Candidate State	$\widetilde{C}_t$	$\mathrm{Mish}\left(W_C\left(r_t \odot h_{t-1}, x_t\right) + b_C\right)$
Update Gate	z_t	$\sigma(\mathrm{LN}(W_{zh}h_{t-1} + W_{zx}x_t + W_{cz} \odot C_{t-1} + b_z))$
Cell State	C_t	$z_t \odot C_{t-1} + (1 - z_t) \odot \widetilde{C}_t$
Output Gate	o_t	$\sigma(\mathrm{LN}(W_{oh}h_{t-1} + W_{ox}x_t + W_{co} \odot C_t + b_o))$
Hidden State	h_t	$o_t \odot \tanh(C_t)$

3.1. L-GRU (long-term gated recurrent unit) architecture

L-GRU targets long-term dependencies with a compact design by preserving the LSTM cell state and simplifying GRU-style control to balance capacity and computational cost. Figure 1 illustrates the internal architecture of the proposed L-GRU cell, detailing the flow of information from the previous hidden state h_{t-1} and current input x_t through the highly integrated gating mechanisms. Unlike standard architectures, L-GRU incorporates Peephole Connections, Decoupled Weight Matrices, and Layer Normalization directly into its core modules to stabilize gradients and enhance feature extraction.

As depicted in Figure 1, L-GRU features four highly integrated modules designed to overcome standard RNN limitations:

1. **Update Gate Module:** Fusing the forget and input functions, z_t employs decoupled weight matrices (W_{zx}, W_{zh}) to map inputs and hidden states independently. A peephole connection (W_{cz}) observes the previous long-term cell state C_{t-1}. To stabilize gradient propagation, the sum is normalized via Layer Normalization (LayerNorm) [19, 20] before applying the Sigmoid activation.

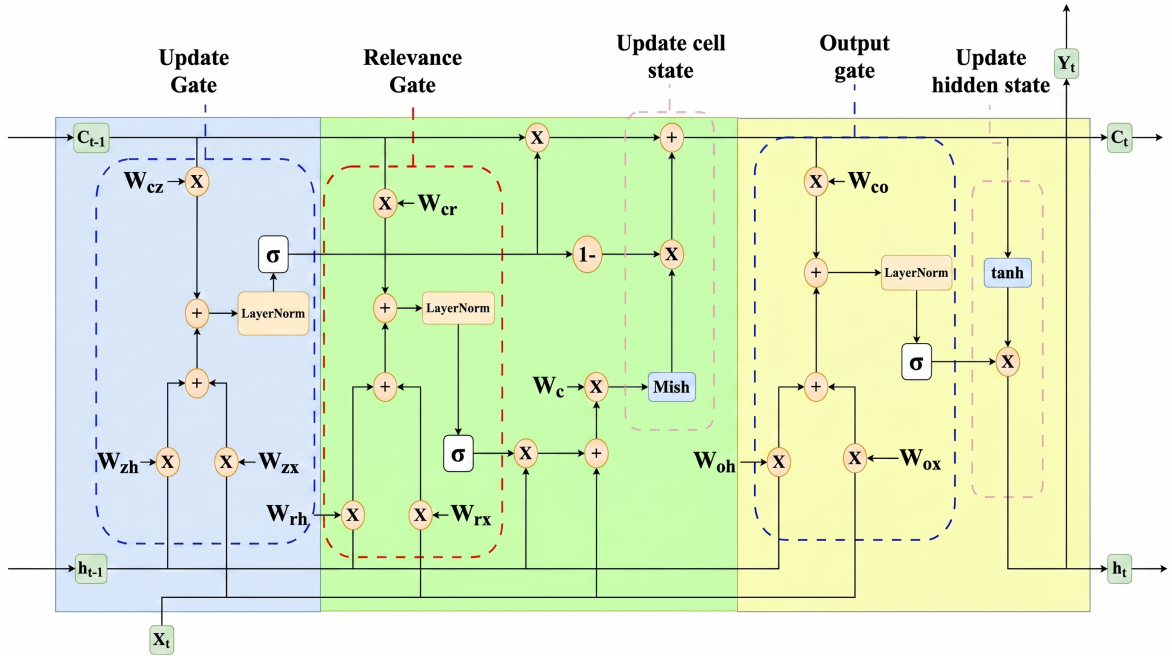


Figure 1: Block diagram of the proposed L-GRU architecture illustrating the integration of the independent cell state and the coupled gating mechanism.

2. **Relevance Gate & Candidate State:** r_t filters the short-term context h_{t-1} using similarly decoupled weights, peephole connections, and LayerNorm. Once scaled, it merges with x_t through a Mish activation function [21] to form the Candidate State ($\tilde{C}_t$), smoothing the optimization landscape compared to standard tanh.

3. **Cell State Update Module:** This module linearly interpolates C_{t-1} and $\tilde{C}_t$ guided by z_t. This additive operation avoids the gradient vanishing typical of multiplicative interactions in deep sequences, robustly locking in long-term memory.

4. **Output Gate & Hidden State:** The output gate (o_t) filters information from the *newly updated* C_t (accessed via peephole W_{co}). h_t is derived by gating the tanh-squashed cell state, ensuring semantic focus prevents memory dilution.

Figure 2 visually contrasts these structural refinements against traditional architectures.

3.2. Theoretical analysis of gradient flow in L-GRU

In standard GRUs, backpropagation relies heavily on the single update gate; a saturated z_t causes the error signal to decay exponentially, leading to vanishing gradients. Conversely, L-GRU's architecture mitigates this decay. By decoupling linear transformations and retaining an additive LSTM-style cell state interpolation mechanism, L-GRU establishes a continuous information highway. This ensures the analytical gradient $\frac{\partial h_t}{\partial h_{t-k}}$ flows backwards steadily across extended time steps without exponential degradation, balancing robust memory retention with structural efficiency.

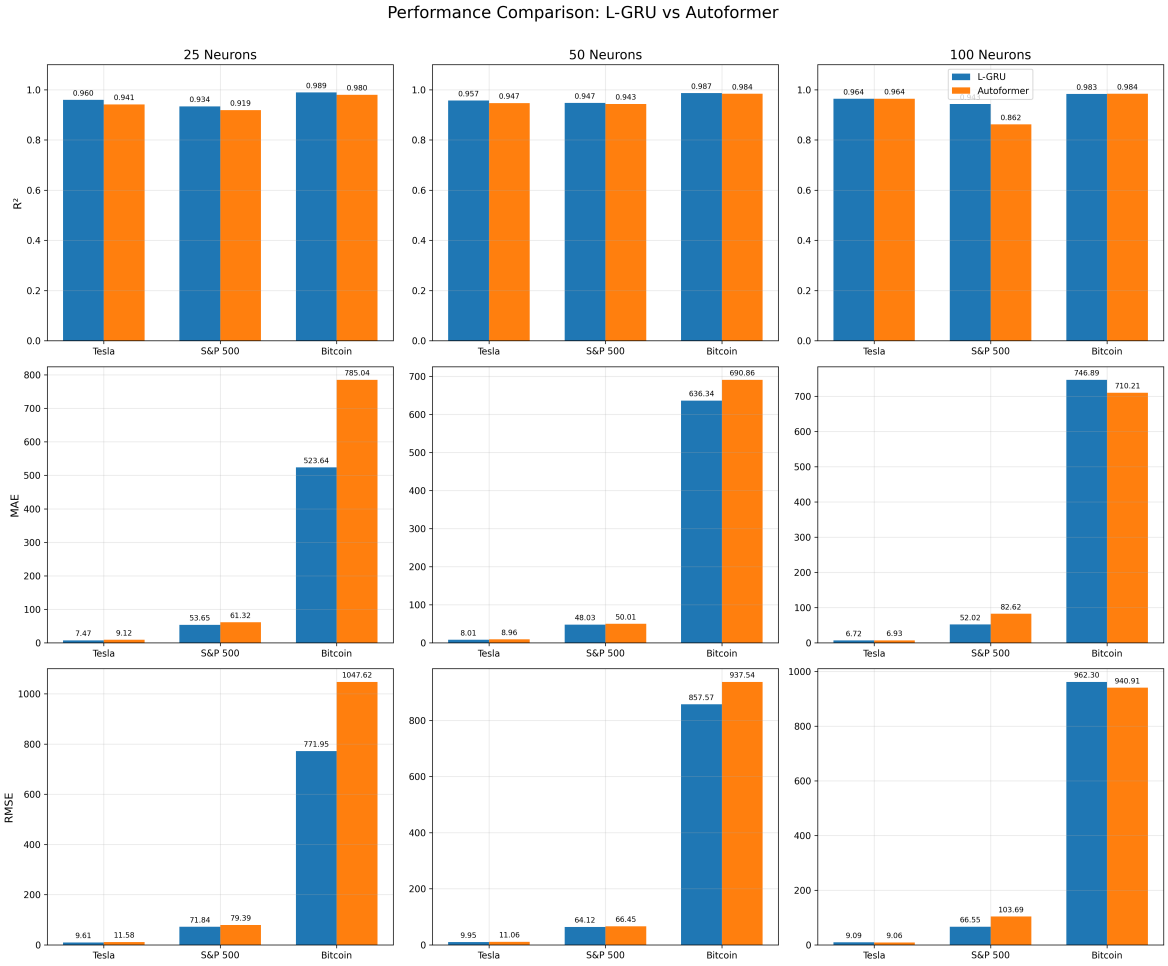


Figure 2: Detailed structural comparison between the standard models and the proposed L-GRU architecture.

4. EXPERIMENTS AND RESULTS EVALUATION

4.1. Implementation and application workflow

To evaluate the effectiveness and stability of the L-GRU model in multi-domain time series forecasting, we conducted experiments on four datasets: three financial datasets (Tesla stock price – TSLA, S&P 500 index, Bitcoin price – BTC) and one meteorological dataset (Jena Climate). These datasets were collected from publicly available sources, namely Yahoo Finance and the Max Planck Institute, ensuring reliability and representativeness for both stochastic data (finance) and physically governed data (weather).

a) *Data preprocessing*

We follow a standard preprocessing pipeline: MinMaxScaler normalization to [0, 1] for stable training, an 80/20 train-test split, and a sliding-window setup that forms input sequences (e.g., 60 days) with next-step labels (the following day) to learn temporal dependencies.

b) *Model architecture and experimental setup*

We implement a single L-GRU layer with 50 neurons followed by a Dense output layer; 50 units balance capacity with overfitting risk and computational cost, while smaller settings may underfit volatile financial series.

For fair comparison, the **Transformer** baseline uses a vanilla setup with 2 encoder layers, 4 attention heads, and a feed-forward dimension of 128 for short-to-medium horizon forecasting.

Table 2 summarizes the specific experimental parameter settings employed for the L-GRU model and baselines to ensure rigor and reproducibility.

Table 2: Experimental parameter settings for the proposed model

Parameter	Value
Sequence Length (Window Size)	60 (Finance) / 24 (Weather)
Hidden Size (Neurons)	50
Batch Size	32
Optimizer	Adam
Learning Rate	0.001
Loss Function	Mean Squared Error (MSE)
Max Epochs	100
Early Stopping Patience	15

All model implementations computationally leveraged PyTorch. Key hyperparameters including MSE loss, Adam optimizer, a batch size of 32, and 100 maximum epochs with early stopping (patience of 15) were configured based on standard preliminary tuning to ensure stable convergence across noisy financial data. A fixed random seed guaranteed reproducibility.

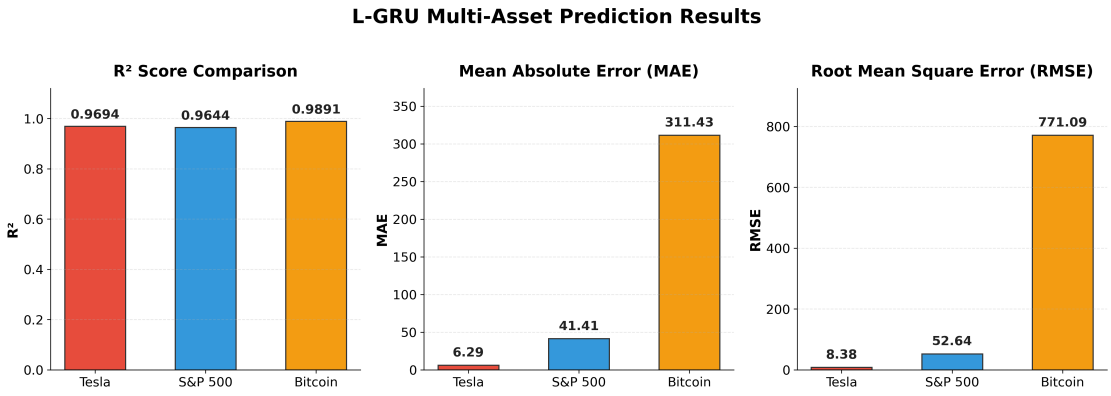


Figure 3: Experimental results of the model on the dataset

4.2. Experimental results

a) *Model performance evaluation*

L-GRU is benchmarked against standard LSTM, GRU, and Transformer architectures using four standard regression metrics: Mean Absolute Error (MAE), Mean Squared Er-

ror (MSE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). Figure 3 and Figure 4 summarize the experimental performance.

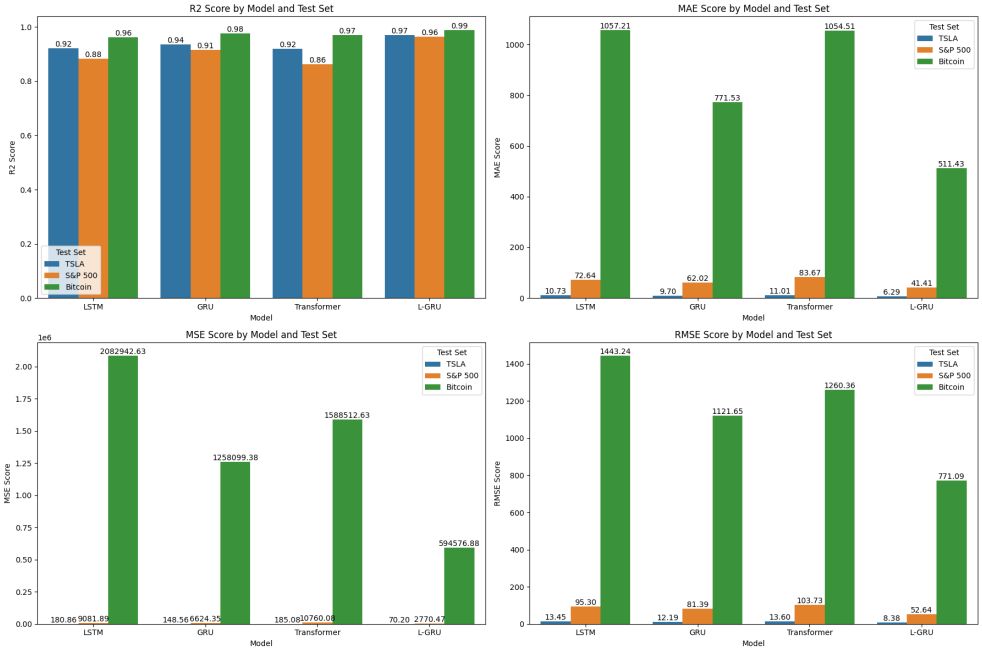


Figure 4: Performance Comparison among Models on Three Test Sets

The performance metrics (Figure 4) reveal that L-GRU consistently outperforms the baseline models across datasets:

- **TSLA:** L-GRU achieves the highest R^2 (0.9694) outperforming LSTM (+5.24%), GRU (+3.66%), and Transformer (+5.45%) while slashing MAE by 41.38% compared to LSTM.
- **S&P 500:** L-GRU attains top accuracy (R^2=0.9644) and dramatically cuts MSE to roughly 30.5% of LSTM's error, demonstrating exceptional stability.
- **Bitcoin:** Despite extreme non-linear volatility, L-GRU generalizes robustly, realizing the lowest MAE (511.43) and highest R^2 (0.9891).

Overall, L-GRU's structural enhancements, peephole connections, decoupled weight matrices, and LayerNorm successfully optimize contextual transmission, yielding simultaneous improvements across all metrics.

b) *Computational cost analysis of the model*

We further compare computational cost (parameters, training time) and run an ablation study to quantify each component's contribution.

Table 3 indicates that L-GRU achieves a favorable complexity-efficiency balance. It reduces the parameter count by 12% compared to LSTM, shortening training time by ∼1.7s/epoch. The basic L-GRU variant further minimizes computational cost (10.1s/epoch), closely matching GRU while maintaining superior performance.

c) *Sensitivity analysis of neuron count*

Table 3: Comparison of Parameter Count and Training Time

Model	Number of Parameters (Params)	Training Time / epoch (s)
LSTM	∼130K	15.2
GRU	∼98K	11.8
L-GRU (Full Version)	∼114K	13.5
L-GRU (Basic Version)	∼82K	10.1

To investigate the impact of model capacity on forecasting performance and justify the selection of 50 neurons for the main experiments, we conducted a sensitivity analysis by varying the number of neurons in the L-GRU layer to 25, 50, and 100.

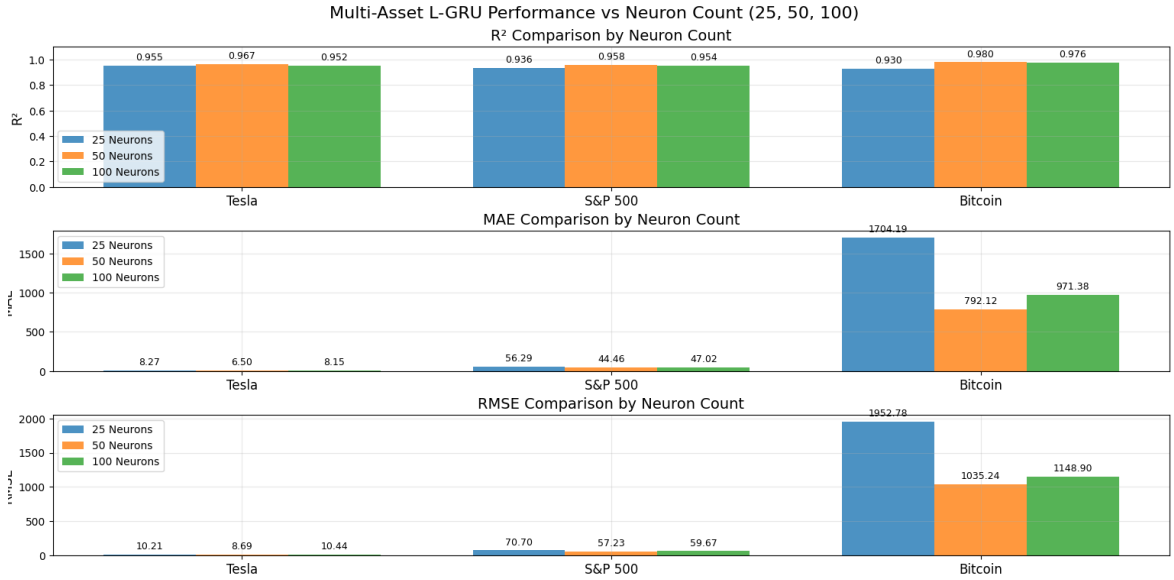


Figure 5: Multi-Asset L-GRU Performance vs Neuron Count (25, 50, 100)

Figure 5 shows that increasing neurons from 25 to 50 consistently improves R^2 and reduces RMSE (e.g., Bitcoin R^2 0.930 $\rightarrow$ 0.980), while 100 neurons yields diminishing returns (Bitcoin R^2 slightly drops to 0.976) at much higher computational cost; thus, 50 neurons is the best accuracy-efficiency trade-off.

d) *Ablation study*

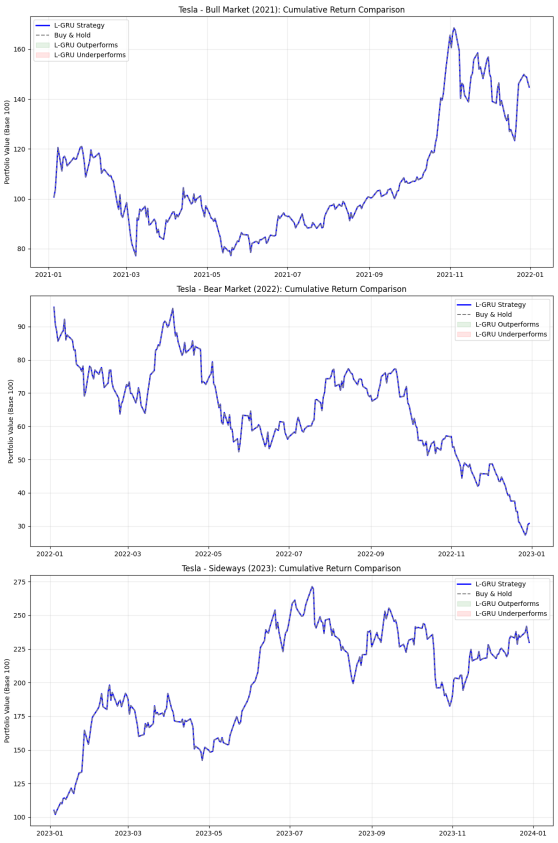
Ablation confirms each component's contribution: removing Layer Normalization or Decoupled Weights causes the largest drop in R^2 (from 0.9545 to ∼0.9420–0.9485), while Peephole Connections provide smaller but consistent gains, especially for volatile series such as Bitcoin.

e) *Evaluation of financial profitability and robustness*

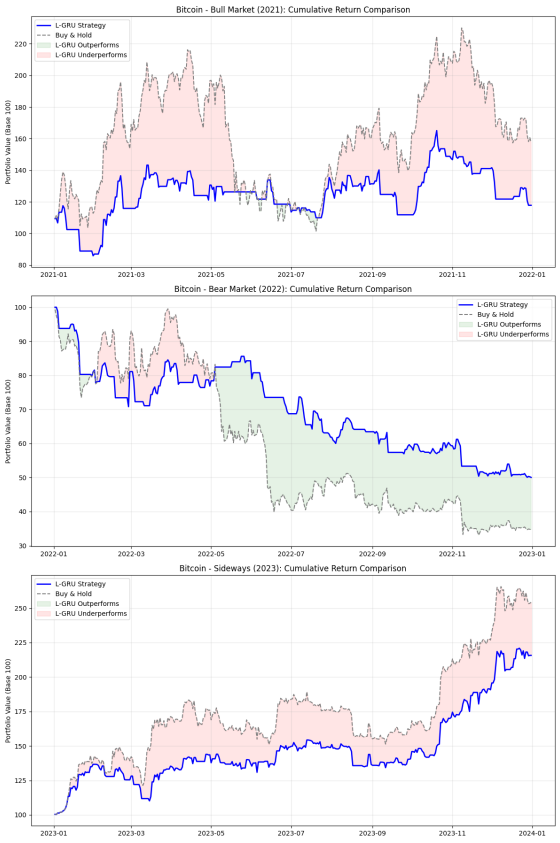
To assess practical viability, we implemented a simulated trading strategy across three market regimes: Bull (2021), Bear (2022), and Sideways (2023). Investment potential is quantified using Return on Investment (ROI), Sharpe Ratio, Maximum Drawdown (MDD), and Mean Absolute Percentage Error (MAPE).

Table 4: Ablation study across all three datasets

L-GRU Variant	TSLA	S&P 500	Bitcoin
L-GRU (Full Version)	0.9694	0.9644	0.9891
- Without LayerNorm	0.9573	0.9485	0.9734
- Without Peephole	0.9432	0.9467	0.9634
- Without Decoupled Weights	0.9598	0.9423	0.9683
L-GRU (Basic Version)	0.9554	0.9420	0.9680



(a) Tesla Cumulative Returns (Bull, Bear, Sideways)



(b) Bitcoin Cumulative Returns (Bull, Bear, Sideways)

Figure 6: Cumulative Return Comparison between L-GRU Strategy and Buy & Hold

Table 5 summarizes the comparison: L-GRU is generally more stable, e.g., Tesla Sideways (2023) obtains ROI of 129.86% and Sharpe of 1.87 vs. LSTM obtains ROI of 39.11% and Sharpe of 0.95, and Bitcoin Bear (2022) lower drawdown (MDD of L-GRU is -49.95% vs. MDD of LSTM is -66.74%).

Figure 6 shows the L-GRU strategy often outperforming Buy & Hold by limiting bear-market losses and exploiting sideways volatility.

Table 5: Profitability and Risk Analysis across Different Market Regimes (2021-2023)

Asset	Regime	Model	RMSE	MAPE (%)	ROI (%)	Sharpe	MDD (%)
Tesla	Bull (2021)	L-GRU	9.39	2.48	44.81	0.95	-36.25
		LSTM	9.41	2.48	26.69	0.77	-45.46
		Transformer	9.58	2.56	19.98	0.64	-31.87
	Bear (2022)	L-GRU	11.33	3.28	-69.20	-1.49	-71.53
		LSTM	11.36	3.29	-56.27	-1.22	-58.24
		Transformer	11.49	3.31	-64.98	-1.59	-70.39
	Sideways (2023)	L-GRU	7.14	2.55	129.86	1.87	-32.72
		LSTM	7.19	2.57	39.11	0.95	-39.25
		Transformer	7.38	2.66	17.24	0.58	-34.66
Bitcoin	Bull (2021)	L-GRU	1905.59	3.12	17.79	0.48	-28.64
		LSTM	1911.14	3.13	57.64	0.81	-53.06
		Transformer	1906.72	3.11	0.73	0.30	-51.88
	Bear (2022)	L-GRU	1013.90	2.27	-49.95	-1.35	-49.95
		LSTM	1027.23	2.31	-65.30	-1.12	-66.74
		Transformer	1020.40	2.28	-60.39	-0.99	-65.10
	Sideways (2023)	L-GRU	655.37	1.52	115.74	1.92	-19.41
		LSTM	653.09	1.55	154.23	1.96	-20.06
		Transformer	653.37	1.52	93.29	1.64	-18.70

4.3. Comparative analysis with state-of-the-art transformers

To demonstrate SOTA relevance, we compared L-GRU with Autoformer [7] across three datasets with varying hidden sizes (HS: 25, 50, 100). Table 6 highlights L-GRU's superior parameter efficiency:

- **Small/Medium HS (25, 50):** L-GRU consistently outperforms Autoformer. On Bitcoin (HS=25), L-GRU reaches an R^2 of 0.9891 (vs 0.9799) and reduces MAE from 785.03 to 523.63.
- **Large HS (100):** L-GRU resists overfitting far better than Autoformer. On S&P 500, Autoformer's R^2 drops abruptly to 0.8619 (RMSE 103.69), whereas L-GRU maintains robust accuracy (R^2 0.9431, RMSE 66.54).

Ultimately, L-GRU matches or exceeds advanced specialized Transformers in highly volatile environments, completely mitigating the traditional GRU vanishing gradient issue at a fraction of the computational footprint.

4.4. Evaluation of generalization capability on meteorological data (Jena Climate)

To assess generalization beyond finance, we evaluate L-GRU on the Jena Climate dataset, a cyclic and more deterministic physical time series [22].

a) *Data description and experimental setup*

Using the Jena Climate benchmark [8, 23], we formulated a short-term IoT edge forecasting scenario using Temperature, Pressure, and Relative Humidity [24]. Data was downsampled to an hourly resolution to capture macro-level diurnal cycles. The model utilizes a

Table 6: Performance Comparison between Autoformer and L-GRU

Model	Asset	Hidden Size	R^2	MAE	MSE	RMSE
Autoformer	Bitcoin	25	0.9799	785.0391	1097510.3750	1047.6213
L-GRU	Bitcoin	25	0.9891	523.6357	595900.3125	771.9458
Autoformer	S&P 500	25	0.9190	61.3189	6303.0825	79.3920
L-GRU	S&P 500	25	0.9337	53.6501	5161.3105	71.8423
Autoformer	Tesla	25	0.9415	9.1165	134.1745	11.5834
L-GRU	Tesla	25	0.9597	7.4672	92.4105	9.6130
Autoformer	Bitcoin	50	0.9839	690.8571	878988.7500	937.5440
L-GRU	Bitcoin	50	0.9866	636.3365	735427.8750	857.5709
Autoformer	S&P 500	50	0.9433	50.0073	4415.3110	66.4478
L-GRU	S&P 500	50	0.9472	48.0333	4111.7422	64.1229
Autoformer	Tesla	50	0.9467	8.9637	122.2633	11.0573
L-GRU	Tesla	50	0.9568	8.0084	99.0694	9.9534
Autoformer	Bitcoin	100	0.9838	710.2144	885307.0625	940.9076
L-GRU	Bitcoin	100	0.9831	746.8863	926016.5000	962.2975
Autoformer	S&P 500	100	0.8619	82.6202	10752.4219	103.6939
L-GRU	S&P 500	100	0.9431	52.0162	4428.6006	66.5477
Autoformer	Tesla	100	0.9642	6.9292	82.1334	9.0628
L-GRU	Tesla	100	0.9640	6.7170	82.5899	9.0879

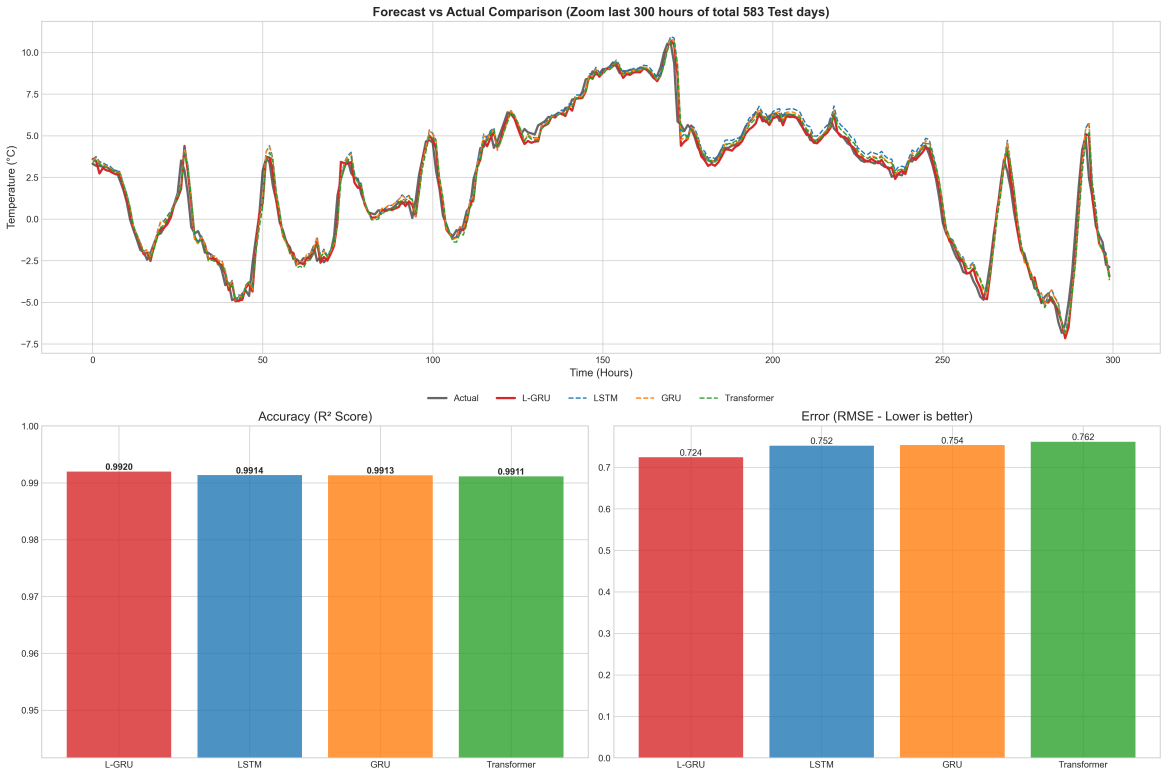


Figure 7: Actual vs. predicted temperature with 300-hour zoom (top) and 583-day R^2/RMSE summary (bottom).

Table 7: Forecasting Performance on the Jena Climate Dataset (Extended Test Set: 583 Days).

Model	R^2	RMSE	Training Time (s/epoch)
LSTM	0.9914	0.752	4.2
GRU	0.9913	0.754	3.1
Transformer	0.9911	0.762	5.8
L-GRU	0.9920	0.724	3.5

24-hour sliding window to predict the subsequent hour. Crucially, we extended the evaluation set to 583 days (∼14,000 hours) to rigorously assess multi-seasonal generalization. Training utilized Mixed Precision (FP16) for hardware efficiency.

b) *Results and discussion*

The experimental results of comparing L-GRU with the baseline models (LSTM, GRU and Transformer) on the extended test set are presented in Table 7 and Figure 7.

The extended 583-day test results reinforce the effectiveness of L-GRU:

- **Long-term stability and accuracy:** L-GRU achieves the best R^2 (0.9920) and the lowest RMSE (0.724), outperforming LSTM/GRU and improving upon Transformer RMSE (0.762).
- **Robust tracking:** Figure 7 shows close tracking of peaks and subtle variations with reduced lag.
- **Efficiency:** Training time (3.5s/epoch) remains close to GRU and is about 17% faster than LSTM, supporting deployment under limited resources.

5. CONCLUSIONS

This paper introduces the L-GRU, a hybrid recurrent architecture combining an independent memory cell with a streamlined gating mechanism, augmented by decoupled weights, peepholes, and Layer Normalization. Extensive evaluations across highly volatile financial markets and deterministic meteorological datasets demonstrate that L-GRU consistently outperforms standard LSTM, GRU, and state-of-the-art Transformer baselines. Generating lower errors, superior R^2 scores, and higher risk-adjusted returns, L-GRU ensures robust long-term memory retention and parameter efficiency. Future research will explore L-GRU deployments on resource-constrained edge devices and natural language processing tasks.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning Phrase Representations using RNN Encoder–Decoder for Statisti-

cal Machine Translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, 2014, pp. 1724–1734.

[3] X. Kong, Z. Chen, W. Liu, K. Ning, L. Zhang, S. M. Marier, Y. Liu, Y. Chen, and F. Xia, "Deep learning for time series forecasting: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 16, pp. 5079–5112, 2025.

[4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning*, Montreal, Canada, 2014.

[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 *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[6] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–28, 2022.

[7] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 22 419–22 430.

[8] Max Planck Institute for Biogeochemistry, "Jena climate dataset," [Online]. Available: https://www.bgc-jena.mpg.de/wetter/, 2016.

[9] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[10] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *International Conference on Learning Representations*. OpenReview.net, 2020.

[11] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*. OpenReview.net, 2023.

[12] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," in *International Conference on Learning Representations*. OpenReview.net, 2024.

[13] A. Farhadi, A. Zamanifar, A. Alipour, A. Taheri, and M. Asadolahi, "A hybrid LSTM-GRU model for stock price prediction," *IEEE Access*, vol. 13, pp. 1–1, 2025.

[14] F. Moodi, A. J. Rafsanjani, S. Zarifzadeh, and M. A. Z. Chahooki, "Advanced stock price forecasting using a 1d-cnn-gru-lstm model," *Journal of AI and Data Mining*, vol. 12, no. 3, pp. 393–408, 2024.

[15] Y. Li and Y. Pan, "Application of attention-based lstm hybrid models for stock price prediction," *IEEE Access*, vol. 12, 2024.

[16] B. Amiri, A. Haddadi, and K. F. Mojdehi, "Novel Hybrid GCN-LSTM Algorithm for Energy Stock Price Prediction," *IEEE Access*, vol. 13, p. 24816, 2025.

[17] P. D. Phong, N. D. Du, P. H. Hiep, and T. X. Thanh, "A hybrid pso-sa scheme for improving accuracy of fuzzy time series forecasting models," *Journal of Computer Science and Cybernetics*, vol. 38, no. 3, pp. 257–275, 2022.

[18] N. D. Hieu, N. C. Ho, P. D. Phong, V. N. Lan, and P. H. Hiep, "Scalable human knowledge about numeric time series variation and its role in improving forecasting results," *Journal of Computer Science and Cybernetics*, vol. 38, no. 2, pp. 103–130, 2022.

[19] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[20] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization Techniques in Training DNNs: Methodology, Analysis and Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10 173–10 196, 2023.

[21] D. Misra, "Mish: A self regularized non-monotonic neural activation function," in *Proceedings of the 31st British Machine Vision Conference (BMVC)*, 2020.

[22] L. Oliveira and G. Messori, "Deep learning for weather forecasting: a review of recent advances and open challenges," *Geoscientific Model Development*, vol. 17, pp. 2347–2376, 2024.

[23] A. Utku and U. Can, "An efficient hybrid weather prediction model based on deep learning," *International Journal of Environmental Science and Technology*, vol. 20, pp. 11 107–11 120, 2023.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things*, vol. 3, no. 5, pp. 637–646, 2016.

Received on February 13, 2026
Accepted on March 31, 2026