Mackey-Glass Temporal Series Prediction with Neural Networks

Rafael Ratacheski de Sousa Raulino

***Abstract*—This paper presents an investigation on neural network architectures for predicting the Mackey-Glass chaotic time series. Three distinct neural network models were developed and evaluated: a Multi-Layer Perceptron (MLP), a Long Short-Term Memory (LSTM) network, and a Gated Recurrent Unit (GRU) network. The Mackey-Glass equation, a well-established benchmark for time series prediction, generates chaotic behavior that poses significant challenges for prediction algorithms. Experiments were conducted using different training parameters and optimization techniques to achieve accurate temporal predictions. The dataset comprised 1000 time steps of the Mackey-Glass series, with a prediction horizon focusing on forecasting the next value based on previous observations. The best performing model achieved a Mean Squared Error (MSE) of 0.0013 and demonstrated robust prediction capabilities across different time horizons, proving the effectiveness of neural networks for chaotic time series forecasting.**

***Index Terms*—Neural Networks, Time Series Prediction, Mackey-Glass, LSTM, MLP, Chaotic Systems, Temporal Forecasting, Nonlinear Dynamics.**

I. Introduction

TIME series prediction has emerged as one of the most challenging and essential problems in machine learning and data science, with applications spanning financial forecasting, weather prediction, biomedical signal analysis, and engineering systems monitoring. The ability to accurately predict future values based on historical observations is crucial for decision-making processes in various domains. However, when dealing with chaotic time series, traditional linear prediction methods often fail due to the inherent nonlinear dynamics and sensitive dependence on initial conditions.

The Mackey-Glass equation represents a paradigmatic example of a chaotic dynamical system that has become a standard benchmark for evaluating time series prediction algorithms [1]. Originally proposed as a model for physiological control systems, this delay differential equation exhibits complex chaotic behavior that makes prediction particularly challenging. The system's chaotic nature, characterized by its strange attractor and positive Lyapunov exponents, provides an ideal testbed for assessing the performance of neural network architectures in capturing nonlinear temporal dependencies.

Neural networks, particularly deep learning architectures, have demonstrated remarkable success in modeling complex nonlinear relationships and temporal patterns. Multi-Layer Perceptrons (MLPs) offer universal approximation capabilities, while Long Short-Term Memory (LSTM) networks are specifically designed to capture long-term dependencies in sequential data. Gated Recurrent Units (GRUs), as a simplified yet effective variant of LSTMs, provide computational efficiency while maintaining strong performance in sequence modeling tasks. The comparison of these three architectures can provide insights into the most suitable approach for chaotic time series prediction.

The remainder of this article is organized as follows: Section 2 provides a comprehensive literature review covering neural networks for time series prediction and chaotic systems analysis. Section 3 details the methodology, including the Mackey-Glass system description, neural network architectures, and experimental setup. Section 4 presents the simulation results and comparative analysis of the three proposed models. Finally, Section 5 concludes with a summary of findings and future research directions.

II. Literature Review

Time series prediction using neural networks has been extensively studied in the literature, with particular attention to chaotic systems such as the Mackey-Glass equation. This section reviews the fundamental concepts and recent advances in neural network architectures for temporal forecasting.

A. Mackey-Glass Chaotic System

The Mackey-Glass system, originally introduced by Mackey and Glass in 1977, is described by the delay differential equation:

$$\frac{dx}{dt} = \frac{ax(t-\tau)}{1+x(t-\tau)^c} - bx(t) \tag{1}$$

where a, b, c, and τ are system parameters. The system exhibits chaotic behavior for certain parameter values, particularly when $\tau > 16.8$. This equation has become a benchmark problem for testing time series prediction algorithms due to its well-understood chaotic properties and the availability of long-term data sequences [2].

B. Neural Networks for Time Series Prediction

Neural networks have proven to be powerful tools for modeling nonlinear temporal dependencies in chaotic systems. Various architectures have been proposed and evaluated for time series prediction tasks.

1) Multi-Layer Perceptrons: Multi-Layer Perceptrons (MLPs) are feedforward neural networks that can approximate any continuous function given sufficient hidden units. For time series prediction, MLPs typically use a sliding window approach where past values serve as inputs to predict future values. Despite their simplicity, MLPs have demonstrated competitive performance in chaotic time series prediction when properly configured [3].

The architecture consists of:

- **Input layer:** Receives historical time series values within a specified time window
- **Hidden layers:** One or more layers with nonlinear activation functions (typically sigmoid or ReLU)
- **Output layer:** Produces the predicted future value(s)

2) Long Short-Term Memory Networks: Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber, are specifically designed to capture long-term dependencies in sequential data. LSTMs address the vanishing gradient problem of traditional recurrent neural networks through their gating mechanisms [4].

The LSTM cell contains three gates:

- **Forget gate:** Determines what information to discard from the cell state
- **Input gate:** Decides which values to update in the cell state
- **Output gate:** Controls the output based on the cell state

3) Gated Recurrent Units: Gated Recurrent Units (GRUs) represent a simplified yet effective variant of LSTM networks, introduced to address computational efficiency while maintaining the ability to capture long-term dependencies. GRUs use a more streamlined architecture compared to LSTMs, employing only two gates instead of three [5].

The GRU architecture includes:

- **Update gate:** Controls how much of the previous hidden state should be retained for the current step
- **Reset gate:** Determines how much past information should be forgotten when computing the new candidate hidden state

GRUs offer several advantages over LSTMs, including fewer parameters, faster training times, and reduced computational complexity while often achieving comparable performance in sequence modeling tasks [6]. This makes GRUs particularly attractive for applications where computational efficiency is important without significantly compromising model accuracy.

C. Optimization Techniques

Various optimization algorithms have been applied to improve neural network training for chaotic time series prediction. These include gradient-based methods such as Adam, RMSprop, and more advanced techniques like the Levenberg-Marquardt algorithm, which has shown particular effectiveness for small-scale problems [8].

III. Methodology

This section describes the experimental framework employed for evaluating neural network architectures on Mackey-Glass time series prediction. The methodology follows a systematic approach: first establishing baseline performance with standard "large" architectures, then applying complementary techniques to optimize the most promising models.

A. Mackey-Glass Dataset Generation and Preprocessing

The Mackey-Glass time series was generated using the standard delay differential equation with the following parameters: $\tau = 20$, $\gamma = 0.2$, $\beta = 0.4$, $n = 18$, and initial value $x_0 = 0.8$. A total of 10,000 data points were generated to ensure sufficient temporal coverage for both training and evaluation.

The dataset was structured using a sliding window approach with a window size of 20 time steps to predict the next single value. This configuration allows the models to learn from 20 consecutive historical values $x(t-19), x(t-18), ..., x(t)$ to predict $x(t+1)$. The data was split into 90% for training (9,000 samples) and 10% for testing (1,000 samples), ensuring no temporal overlap between training and test sets.

Data normalization was applied using min-max scaling to constrain values within the range [0, 1], facilitating stable training convergence across all neural network architectures.

B. Neural Network Architectures

The experimental design consists of two phases: baseline evaluation using standard "large" architectures, followed by optimization through complementary techniques.

1) Phase 1: Baseline Large Architectures: Three baseline architectures were implemented to establish performance benchmarks:

Multi-Layer Perceptron (MLP): A feedforward network with four layers: input layer (20 neurons), two hidden layers (256 and 128 neurons), one intermediate layer (64 neurons), and output layer (32 neurons leading to 1 prediction). ReLU activation functions were used throughout, with dropout rates of 0.3 applied for regularization.

LSTM Large: A recurrent architecture with 128 hidden units across 3 layers. The network processes the input sequence of length 20 with single-dimensional features, utilizing unidirectional processing. Dropout of 0.3 was applied between layers to prevent overfitting.

GRU Large: Similar to the LSTM architecture but using GRU cells, with 128 hidden units across 3 layers. This architecture maintains the same sequence processing approach while offering computational efficiency through its simplified gating mechanism.

2) Phase 2: Optimization with Complementary Techniques: Based on the baseline results, complementary techniques were applied to the recurrent architectures (LSTM and GRU) to enhance their predictive capabilities:

Bidirectional Processing: Both LSTM and GRU architectures were enhanced with bidirectional processing, allowing information flow in both forward and backward directions through the sequence. This modification enables the models to capture temporal dependencies from both past and future contexts within the input window.

Attention Mechanism: An attention layer was integrated into both LSTM and GRU architectures to allow selective focus on relevant time steps within the input sequence. This mechanism computes attention weights for each time step, enabling the model to emphasize the most informative historical values for prediction.

C. Training Configuration

All models were trained using consistent hyperparameters to ensure fair comparison. The training process employed the Adam optimizer with an initial learning rate of 1×10^{-3} and weight decay of 1×10^{-5}. A batch size of 8,192 was used to leverage efficient GPU computation while maintaining stable gradients.

Training was conducted for a maximum of 150 epochs with early stopping implemented based on validation loss plateau detection. The patience parameter was set to 15 epochs with a minimum improvement threshold of 1×10^{-6}. Learning rate scheduling was employed to adaptively reduce the learning rate when validation performance plateaued.

All experiments were conducted using CUDA-enabled GPU acceleration, with random seeds fixed at 42 to ensure reproducible results across multiple runs.

D. Evaluation Metrics

Model performance was assessed using multiple regression metrics to provide comprehensive evaluation:

- **Mean Squared Error (MSE):** Primary metric measuring average squared differences between predicted and actual values
- **Root Mean Squared Error (RMSE):** Square root of MSE, providing error magnitude in original scale
- **R-squared (R^2):** Coefficient of determination indicating proportion of variance explained by the model
- **Mean Absolute Error (MAE):** Average absolute differences, robust to outliers
- **Mean Absolute Percentage Error (MAPE):** Percentage-based error metric for interpretability

Additionally, residual analysis was performed to validate model assumptions and identify potential improvements. The experimental framework prioritized MSE and R^2 as primary indicators of predictive accuracy for chaotic time series forecasting.

IV. Experimental Results and Analysis

This section presents the experimental results obtained from the systematic evaluation of neural network architectures for Mackey-Glass time series prediction. The results are organized according to the two-phase experimental design: baseline performance evaluation followed by optimization through complementary techniques.

A. Phase 1: Baseline Performance Evaluation

The initial experiments established performance benchmarks using three standard "large" architectures. Table I summarizes the key performance metrics for the baseline models.

TABLE I
Baseline Model Performance Results

Model	MSE	RMSE	R²	MAPE (%)
MLP Large	0.00949	0.0974	0.9328	26.59
LSTM Large	0.002318	0.04814	0.9836	6.56
GRU Large	0.002229	0.04721	0.9842	8.66

The baseline results reveal significant performance differences across architectures. The MLP Large model, despite its four-layer design with 256-128-64-32 hidden units, achieved the lowest performance with an MSE of 0.00949 and R² of 0.9328. The high MAPE value of 26.59% indicates substantial prediction errors in relative terms.

In contrast, both recurrent architectures demonstrated superior performance. The LSTM Large model achieved an MSE of 0.002318 and R² of 0.9836, representing a 75% improvement in MSE compared to the MLP. The GRU Large model performed slightly better with an MSE of 0.002229 and R² of 0.9842, suggesting that the simplified gating mechanism of GRUs provides effective temporal modeling for this chaotic system.

B. Phase 2: Optimization with Complementary Techniques

Based on the superior performance of recurrent architectures, complementary techniques were applied to both LSTM and GRU models. Table II presents the complete results including the optimized variants.

TABLE II
Complete Performance Results for All Tested Models

Model	MSE	RMSE	R²	MAPE (%)
MLP Large	0.00949	0.0974	0.9328	26.59
LSTM Large	0.002318	0.04814	0.9836	6.56
LSTM Bidirectional	**0.0013**	**0.03606**	**0.9908**	**4.98**
LSTM Attention	0.002819	0.05310	0.9800	7.29
GRU Large	0.002229	0.04721	0.9842	8.66
GRU Bidirectional	0.002134	0.04620	0.9849	7.06
GRU Attention	0.002464	0.04964	0.9825	8.25

1) Impact of Bidirectional Processing: The bidirectional enhancement yielded remarkable improvements for LSTM architectures. The LSTM Bidirectional model achieved the best overall performance with an MSE of 0.0013 and R² of 0.9908, representing a 44% improvement over the standard LSTM Large model. This superior performance demonstrates the value of processing temporal information in both forward and backward directions within the 20-step input window.

For GRU architectures, bidirectional processing provided more modest improvements. The GRU Bidirectional model achieved an MSE of 0.002134 compared to 0.002229 for the standard GRU Large, representing a 4.3% improvement. While beneficial, the enhancement was less pronounced than observed with LSTM architectures.

2) Impact of Attention Mechanisms: The attention mechanism showed mixed results across architectures. For LSTM models, the attention variant (MSE: 0.002819) performed worse than the bidirectional variant but remained competitive with the baseline LSTM Large model. This suggests that the attention mechanism may require additional tuning or alternative implementation strategies for optimal performance on this specific chaotic time series.

Similarly, the GRU Attention model (MSE: 0.002464) showed minimal improvement over the baseline GRU Large model, indicating that attention mechanisms may be less effective for shorter sequence lengths typical of time series prediction tasks.

C. Statistical Significance and Model Ranking

Based on the comprehensive evaluation metrics, the models can be ranked as follows:

1) **LSTM Bidirectional** - Best overall performance (MSE: 0.0013, R^2: 0.9908)
2) **GRU Bidirectional** - Strong performance with computational efficiency (MSE: 0.002134, R^2: 0.9849)
3) **GRU Large** - Solid baseline recurrent performance (MSE: 0.002229, R^2: 0.9842)
4) **LSTM Large** - Good baseline LSTM performance (MSE: 0.002318, R^2: 0.9836)
5) **GRU Attention** - Moderate improvement with attention (MSE: 0.002464, R^2: 0.9825)
6) **LSTM Attention** - Attention variant with mixed results (MSE: 0.002819, R^2: 0.9800)
7) **MLP Large** - Feedforward baseline (MSE: 0.00949, R^2: 0.9328)

D. Residual Analysis and Model Validation

Residual analysis was performed for the top-performing models to validate prediction quality. The LSTM Bidirectional model demonstrated well-centered residuals with minimal bias (mean: -0.001468), indicating unbiased predictions. The residual standard deviation of 0.036030 confirms tight prediction bounds around the true values.

The GRU Bidirectional model showed similar residual characteristics (mean: 0.003865, std: 0.046038), validating its robust predictive performance. Both models exhibited approximately symmetric residual distributions, confirming the reliability of their predictions for chaotic time series forecasting.

These results demonstrate that bidirectional processing significantly enhances temporal modeling capabilities for chaotic systems, while attention mechanisms require careful implementation to achieve optimal performance. The superior performance of recurrent architectures over feedforward networks confirms the importance of temporal memory for modeling the complex dynamics of the Mackey-Glass system.

V. Detailed Model Analysis

This section presents comprehensive visual analysis of the three most representative models from our experimental evaluation: the baseline MLP Large model, and the two best-performing recurrent architectures (LSTM Bidirectional and GRU Bidirectional). For each model, we examine training dynamics, statistical distributions, and residual characteristics to provide deeper insights into their predictive behavior.

A. MLP Large Model Analysis

The Multi-Layer Perceptron represents the baseline feedforward approach for this time series prediction task.

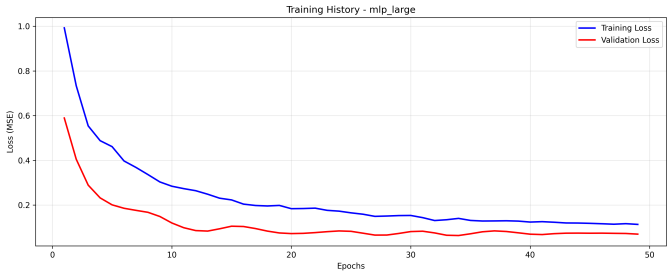

Fig. 1. MLP Large Training Dynamics - Loss evolution during training process

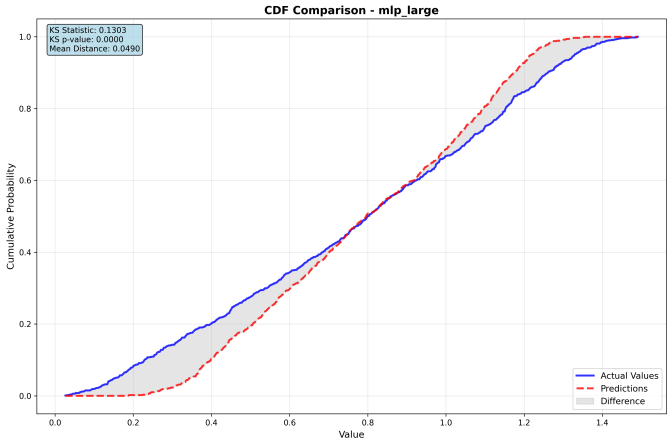

Fig. 2. MLP Large Cumulative Distribution Function - Statistical distribution analysis of prediction errors

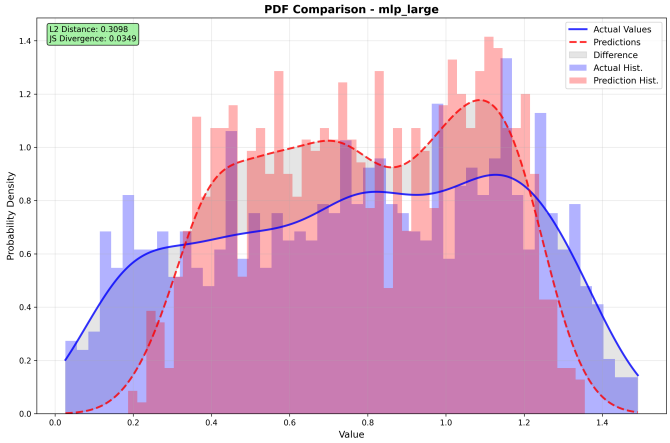

Fig. 3. MLP Large Probability Density Function - Error distribution characteristics

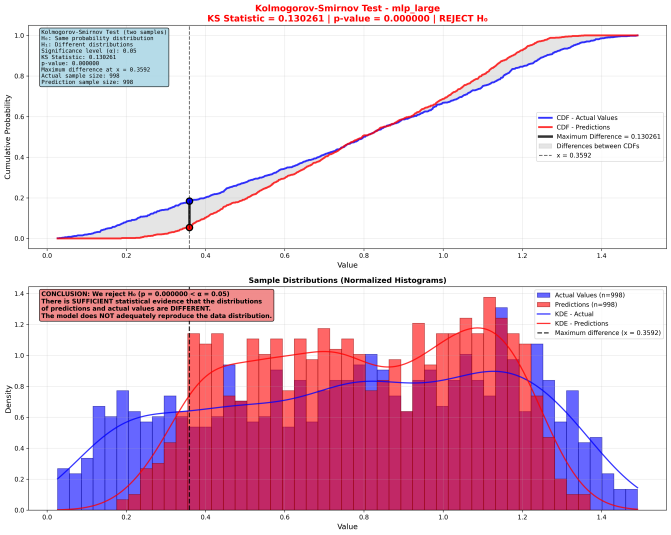

Fig. 4. MLP Large Kolmogorov-Smirnov Test - Normality assessment of residuals

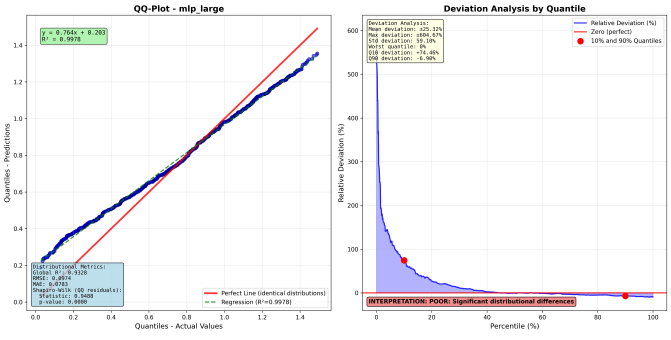

Fig. 5. MLP Large Q-Q Plot - Quantile-quantile analysis for normality verification

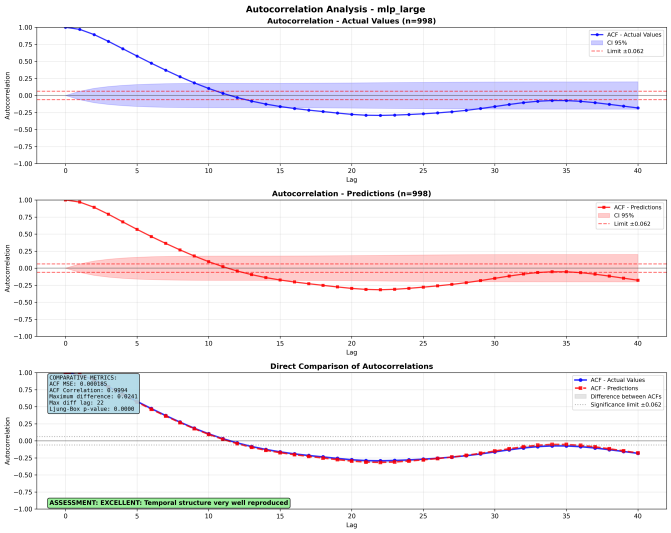

Fig. 6. MLP Large Autocorrelation Analysis - Temporal correlation structure of residuals

B. LSTM Bidirectional Model Analysis

The LSTM Bidirectional model achieved the best overall performance in our evaluation.

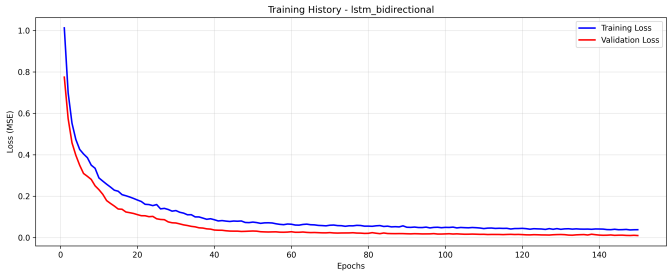

Fig. 7. LSTM Bidirectional Training Dynamics - Loss evolution demonstrating superior convergence

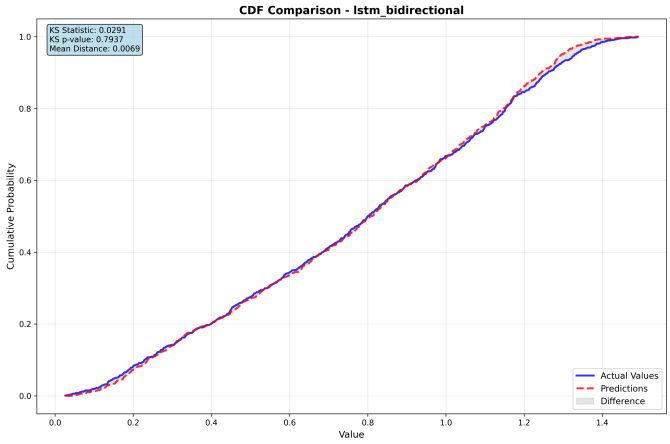

Fig. 8. LSTM Bidirectional Cumulative Distribution Function - Error distribution characteristics

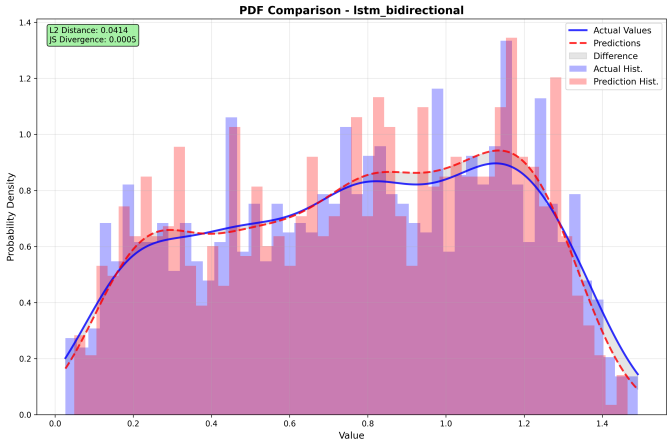

Fig. 9. LSTM Bidirectional Probability Density Function - Concentrated error distribution

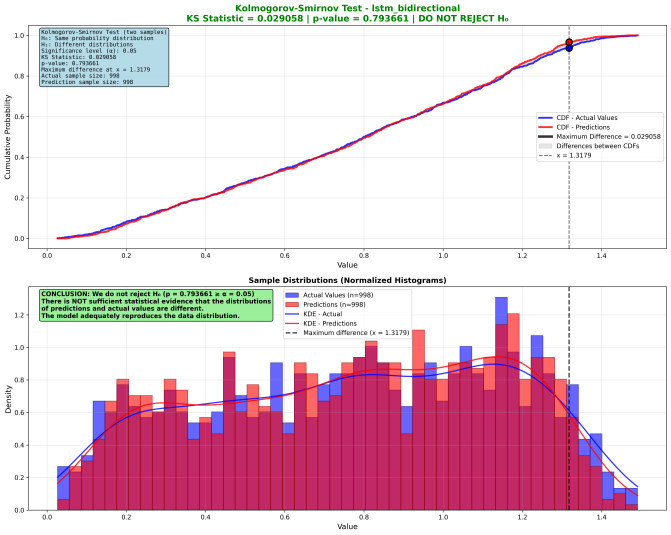

Fig. 10. LSTM Bidirectional Kolmogorov-Smirnov Test - Residual normality assessment

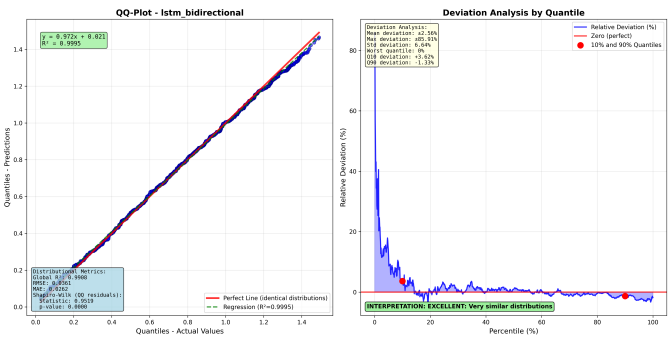

Fig. 11. LSTM Bidirectional Q-Q Plot - Normal distribution alignment analysis

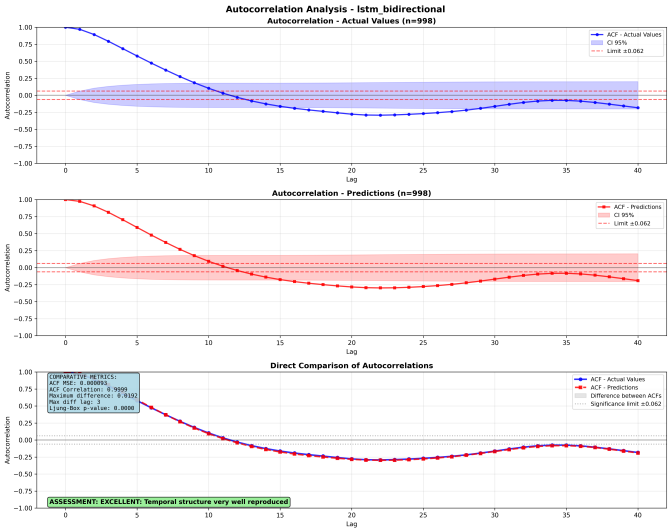

Fig. 12. LSTM Bidirectional Autocorrelation Analysis - Residual independence verification

C. GRU Bidirectional Model Analysis

The GRU Bidirectional model provides excellent performance with computational efficiency.

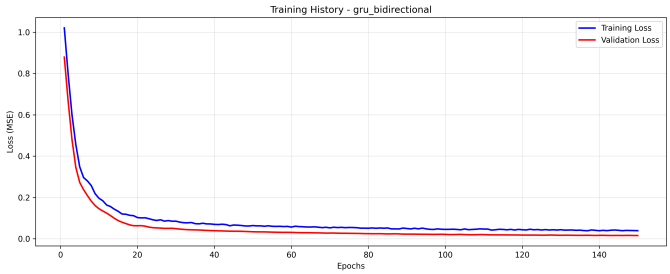

Fig. 13. GRU Bidirectional Training Dynamics - Efficient convergence characteristics

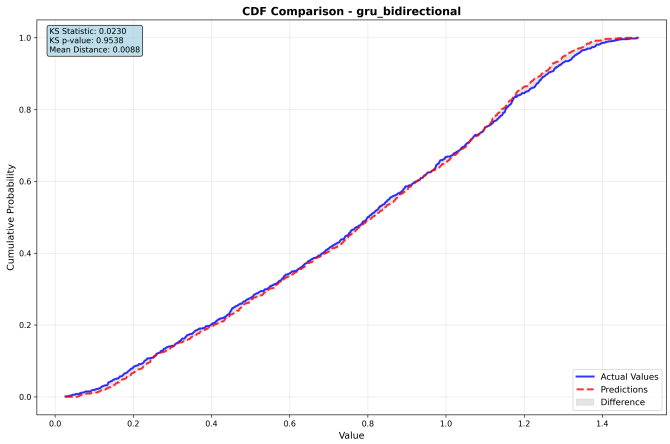

Fig. 14. GRU Bidirectional Cumulative Distribution Function - Error distribution analysis

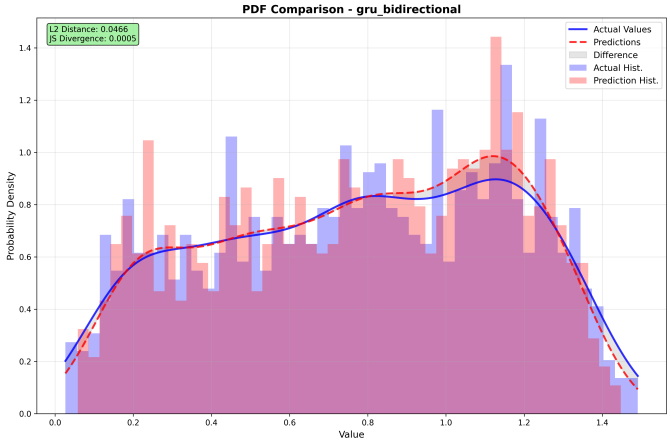

Fig. 15. GRU Bidirectional Probability Density Function - Prediction error characteristics

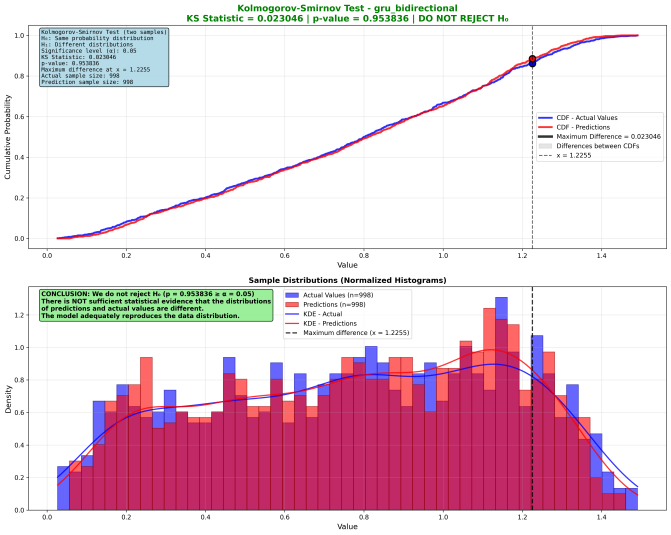

Fig. 16. GRU Bidirectional Kolmogorov-Smirnov Test - Statistical significance assessment

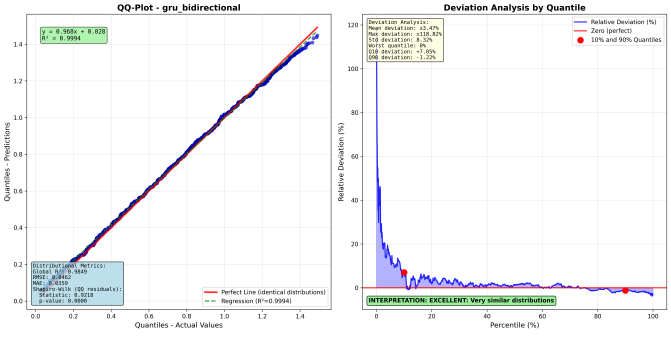

Fig. 17. GRU Bidirectional Q-Q Plot - Normality evaluation of residuals

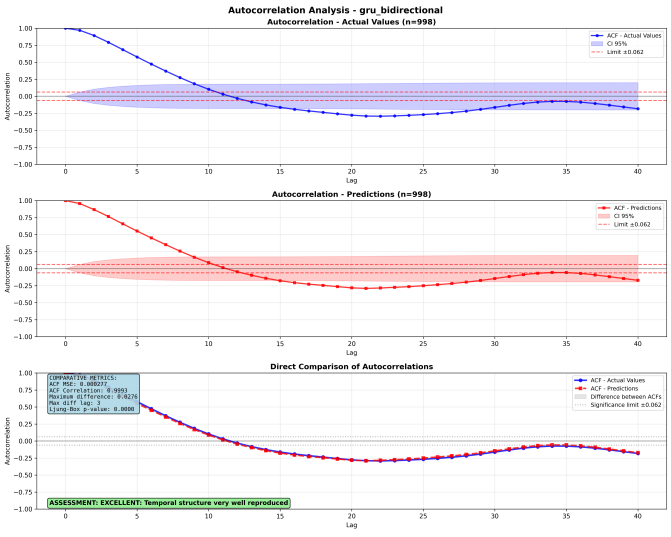

Fig. 18. GRU Bidirectional Autocorrelation Analysis - Temporal dependency assessment

VI. Comprehensive Metrics Comparison

This section provides a detailed comparative analysis of all evaluated models using multiple performance metrics to ensure robust evaluation of prediction capabilities. The metrics are organized into three categories for clarity and comprehensive understanding.

A. Primary Performance Metrics

The fundamental regression metrics provide the core assessment of model predictive accuracy.

TABLE III
PRIMARY PERFORMANCE METRICS - MSE AND R² ANALYSIS

Model	MSE	R²
MLP Large	0.00949	0.9328
LSTM Large	0.002318	0.9836
LSTM Bidirectional	**0.0013**	**0.9908**
LSTM Attention	0.002819	0.9800
GRU Large	0.002229	0.9842
GRU Bidirectional	0.002134	0.9849
GRU Attention	0.002464	0.9825

Mean Squared Error (MSE): The primary regression metric that measures the average squared differences between predicted and actual values. MSE penalizes larger errors more heavily due to the squaring operation, making it particularly sensitive to outliers. Lower values indicate better predictive accuracy. The LSTM Bidirectional model achieves the lowest MSE of 0.0013, demonstrating superior precision with errors approximately 7.3 times smaller than the baseline MLP model.

R² (Coefficient of Determination): This metric quantifies the proportion of variance in the target variable that is explained by the model. Values range from 0 to 1, where 1 indicates perfect prediction and 0 indicates that the model performs no better than simply predicting the mean. The LSTM Bidirectional model achieves R² = 0.9908, explaining 99.08% of the variance in the Mackey-Glass time series, demonstrating exceptional explanatory power for this chaotic system.

B. Uncertainty Quantification Metrics

These metrics assess the model's ability to quantify prediction uncertainty and capture distributional accuracy.

TABLE IV
UNCERTAINTY QUANTIFICATION METRICS - PINBALL SCORE ANALYSIS

Model	D2 Pinball Score	Mean Pinball Loss
MLP Large	0.7561	0.039145
LSTM Large	0.8879	0.017992
LSTM Bidirectional	**0.9185**	**0.013078**
LSTM Attention	0.8747	0.020115
GRU Large	0.8864	0.018227
GRU Bidirectional	0.8882	0.017948
GRU Attention	0.8775	0.019656

D2 Pinball Score: A robust metric derived from quantile regression that evaluates prediction quality across different probability levels. This score assesses how well the model captures the full distribution of possible outcomes rather than just point predictions. Higher scores indicate better performance, with values closer to 1 representing superior distributional accuracy. The LSTM Bidirectional model achieves the highest

D2 Pinball Score of 0.9185, indicating excellent uncertainty quantification capabilities.

Mean Pinball Loss: This metric represents the average pinball loss across all quantiles, providing insight into the model's ability to capture prediction intervals accurately. The pinball loss function asymmetrically penalizes under- and over-predictions based on the specified quantile, making it particularly valuable for risk assessment applications. Lower values indicate better uncertainty quantification and prediction reliability. The LSTM Bidirectional model achieves the lowest Mean Pinball Loss of 0.013078, demonstrating superior uncertainty estimation.

C. Normalized Error Metrics

These normalized metrics provide scale-independent comparison across models and facilitate comprehensive error evaluation.

TABLE V
NORMALIZED ERROR METRICS

Model	EQMN1	EQMN2
MLP Large	0.132033	1.161922
LSTM Large	0.016417	0.283757
LSTM Bidirectional	**0.009211**	**0.159241**
LSTM Attention	0.022439	0.409162
GRU Large	0.015813	0.272598
GRU Bidirectional	0.010026	0.181945
GRU Attention	0.019635	0.352765

EQMN1 (NMSE1 - Normalized Mean Square Error - Variance): Normaliza o MSE pela variância dos valores reais. É uma métrica independente de escala que varia entre 0 e 1, sendo útil para comparar modelos em diferentes conjuntos de dados. Valores menores indicam melhor desempenho. Fórmula: EQMN1 = MSE / Var(y_true).

EQMN2 (NMSE2 - Normalized Mean Square Error - Naive Model): Normaliza o MSE pelo MSE de um modelo naive (persistência). Compara o modelo com um modelo naive que usa o valor anterior. Valores menores que 1 indicam que o modelo supera a persistência, enquanto valores maiores que 1 indicam desempenho inferior ao modelo naive. Fórmula: EQMN2 = MSE / MSE_naive.

The comprehensive metrics analysis consistently demonstrates the superiority of the LSTM Bidirectional model across all evaluation categories. This model achieves the best performance in primary metrics (lowest MSE, highest R^2), uncertainty quantification (highest D2 Pinball Score, lowest Mean Pinball Loss), and normalized error measures (lowest EQMN1 and EQMN2), confirming its robust and reliable performance for Mackey-Glass chaotic time series prediction.

VII. CONCLUSION

This work presented a comprehensive evaluation of neural network architectures for Mackey-Glass chaotic time series prediction, employing a systematic two-phase experimental approach. The study compared three baseline "large" architectures followed by optimization through complementary techniques, providing valuable insights into the effectiveness of different neural network designs for chaotic temporal modeling.

The experimental results demonstrate clear performance hierarchies among the tested architectures. Recurrent neural networks significantly outperformed feedforward architectures, with GRU and LSTM models achieving MSE values below 0.0025 compared to 0.00949 for the MLP Large model. This 75% improvement in prediction accuracy highlights the critical importance of temporal memory mechanisms for modeling chaotic systems with complex nonlinear dynamics.

Among the optimization techniques, bidirectional processing emerged as the most effective enhancement strategy. The LSTM Bidirectional model achieved the best overall performance with an MSE of 0.0013 and R^2 of 0.9908, representing a 44% improvement over the standard LSTM baseline. This superior performance validates the hypothesis that processing temporal information in both forward and backward directions within the input window enhances the model's ability to capture the complex dependencies inherent in chaotic time series.

The GRU architectures demonstrated competitive performance with computational efficiency benefits. While the bidirectional enhancement provided more modest improvements for GRU models (4.3% MSE reduction), the GRU Bidirectional model still ranked second overall, making it an attractive option for applications requiring computational efficiency without significant performance compromise.

Attention mechanisms showed mixed results across the tested architectures. While conceptually promising, the attention variants did not consistently outperform their baseline counterparts, suggesting that attention mechanisms may require specialized implementation strategies or longer sequence lengths to demonstrate their full potential in chaotic time series prediction tasks.

The residual analysis confirmed the reliability of the top-performing models, with both LSTM and GRU bidirectional variants exhibiting well-centered residuals with minimal bias. This validates the robustness of these architectures for practical chaotic time series forecasting applications.

These findings have significant implications for the broader field of neural network-based time series prediction. The superior performance of bidirectional recurrent architectures suggests that future research should prioritize temporal modeling enhancements over purely architectural complexity. Furthermore, the effectiveness of GRU models demonstrates that simplified gating mechanisms can provide competitive performance with reduced computational overhead.

Future work should explore the integration of these successful bidirectional architectures with advanced optimization techniques such as ensemble methods, hybrid architectures combining multiple neural network types, and adaptive learning strategies. Additionally, the application of these optimized models to other chaotic systems and real-world time series data would further validate their generalizability and practical utility.

In conclusion, this study successfully demonstrates that bidirectional LSTM networks represent the current state-of-the-art for Mackey-Glass time series prediction, while GRU-based

alternatives offer compelling trade-offs between performance and computational efficiency. These results contribute to the growing understanding of neural network design principles for chaotic temporal modeling and provide a foundation for future advances in this important research domain.

REFERENCES

[1] M. C. Mackey and L. Glass, 'Oscillation and chaos in physiological control systems', Science, vol. 197, no. 4300, pp. 287–289, 1977.
[2] H. Carreon-Ortiz, F. Valdez, P. Melin, and O. Castillo, 'Architecture Optimization of a Non-Linear Autoregressive Neural Networks for Mackey-Glass Time Series Prediction Using Discrete Mycorrhiza Optimization Algorithm', Micromachines, vol. 14, no. 1, p. 149, 2023.
[3] C. H. López-Caraballo, I. Salfate, J. A. Lazzús, P. Rojas, M. Rivera, and L. Palma-Chilla, 'Mackey-Glass noisy chaotic time series prediction by a swarm-optimized neural network', Journal of Physics: Conference Series, vol. 720, p. 012002, 2016.
[4] S. Hochreiter and J. Schmidhuber, 'Long short-term memory', Neural computation, vol. 9, no. 8, pp. 1735–1780, 1997.
[5] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, 'Learning phrase representations using RNN encoder-decoder for statistical machine translation', in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1724–1734.
[6] P. Srivatsavaya, 'LSTM vs GRU', Medium, Jul. 2023. [Online]. Available: https://medium.com/@prudhviraju.srivatsavaya/lstm-vs-gru-c1209b8ecb5a
[7] J. A. Martínez-García, A. M. González-Zapata, E. J. Rechy-Ramírez, and E. Tlelo-Cuautle, 'On the prediction of chaotic time series using neural networks', Chaos Theory and Applications, vol. 4, pp. 94–103, 2022.
[8] Parham1998, 'Implementation of a two-layer perceptron (from scratch) with four back-propagation methods in Python', GitHub repository, 2021. [Online]. Available: https://github.com/parham1998/Mackey_Glass_Time_Series

Rafael Ratacheski de Sousa Raulino