



École Normale Supérieure

Enhancing Time Series Forecasting with Nonlinear ARMA Models (NLARMA)

A Comparative Study with ARMA on Simulated Stock Price Data

Presented by:

Ayoub Aarab
Abdelhamid Merghad

Supervised by:

Pr.Mohammed Lamarti Sefian

Master Data Science
Module: Time Series

August 6, 2025

Contents

1	Introduction	2
2	Methodology	3
2.1	Dataset Analysis	3
2.1.1	Data Overview	3
2.1.2	Statistical Properties	4
2.2	Model Architectures	4
2.2.1	ARMA Benchmark Model	4
2.2.2	NLARMA Model	5
2.2.3	Hyperparameter Optimization	5
3	Results and Evaluation	6
3.1	Performance Benchmarking	6
3.2	Visual Analysis	6
3.2.1	Initial Un-Optimized Model Performance	6
3.2.2	Optimized Model Performance	7
3.2.3	Residual Analysis	8
3.3	Strengths and Limitations	8
3.4	Future Work	8
4	Conclusion	9

Abstract

This report investigates the performance differences between traditional linear ARMA models and non-linear ARMA (NLARMA) models in time series forecasting. Using a simulated stock price dataset engineered with explicit nonlinearities, we compare an ARIMA(5,0,0) model against a neural network-based NLARMA model. Through performance metrics like RMSE and visual analysis of forecasts and residuals, we demonstrate that the NLARMA model offers a substantial improvement in accuracy. The findings underscore the necessity of employing nonlinear models for complex, real-world systems, and the report concludes with a discussion on model limitations and avenues for future research.

1 Introduction

Time series forecasting is an indispensable discipline across a multitude of sectors, including finance, energy management, and healthcare. The ability to accurately predict future values based on historical data is vital for informed strategic planning, efficient resource allocation, and proactive risk management in these dynamic environments. For decades, conventional statistical models have served as the cornerstone of this field.

Traditional time series models, such as the Autoregressive Moving Average (ARMA) family, are built upon a fundamental assumption: that the underlying data generation process is linear. This implies that future values can be adequately explained as simple linear combinations of past observations and error terms. While these models offer simplicity and interpretability, their inherent design imposes a significant constraint.

A growing body of evidence highlights a critical limitation of these linear approaches. Many real-world phenomena exhibit complex, nonlinear behaviors, chaotic dynamics, and sudden shifts that cannot be adequately captured by linear models. For instance, financial markets are characterized by volatility clustering, energy consumption can be highly sensitive to non-linear factors like temperature, and biological signals often display intricate interactions.

This creates a fundamental conflict: the tools designed for a simplified linear world are being applied to an inherently complex, nonlinear reality. This mismatch means that linear models, by their very design, are often systematically incapable of representing the true dynamics of many systems. Their limitations are not merely minor inaccuracies but a fundamental inability to capture the underlying data-generating process, setting the stage for more advanced approaches.

This recognition has spurred the development of more sophisticated, nonlinear models designed to overcome the limitations of their linear predecessors. The objective of this report is to demonstrate the enhanced forecasting capabilities of Nonlinear ARMA (NLARMA) compared to traditional ARMA, specifically using a simulated stock price dataset engineered to exhibit explicit nonlinear characteristics. This comparison aims to underscore the necessity of adopting models that can adapt to the inherent complexity of the data, rather than forcing the data into a predefined linear mold.

2 Methodology

2.1 Dataset Analysis

The study utilizes a synthetic dataset of 100 daily stock price observations. It was generated using a nonlinear rule to explicitly test the models' ability to capture complex dynamics.

2.1.1 Data Overview

Figure 1 shows the time series, which exhibits a clear upward trend along with periodic fluctuations. A vertical dashed line indicates the split between the training set (first 80 observations) and the test set (final 20 observations). Table 1 shows a sample of the first five data points.

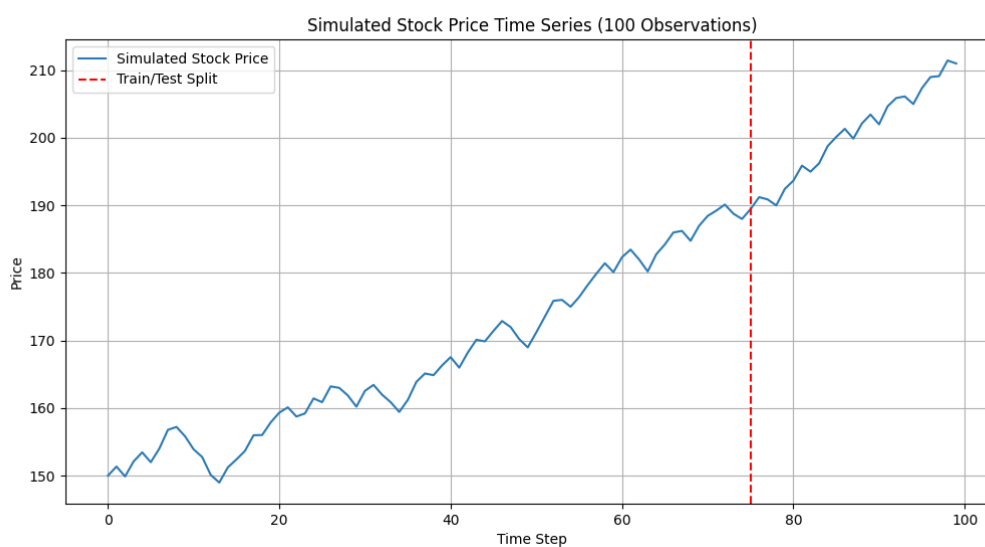


Figure 1: Full time series of the simulated stock price data, with train/test split.

Table 1: First 5 Rows of the Dataset	
Time Step	Simulated Stock Price
0	150.00
1	150.31
2	150.98
3	151.78
4	152.42

2.1.2 Statistical Properties

To understand the underlying characteristics of the data, we analyzed its distribution and autocorrelation, shown in Figure 2.

- The **Histogram** shows that the price values are not normally distributed; there appear to be two primary clusters of values.
- The **Autocorrelation (ACF)** plot shows a very slow decay, which is characteristic of a series with a strong trend.
- The **Partial Autocorrelation (PACF)** plot is most informative for our AR model. It shows a significant spike at lag 1, followed by another significant spike at lag 5, with values cutting off to insignificance afterward. This provides a strong statistical justification for choosing a look-back period of 5 for our AR and NLARMA models.

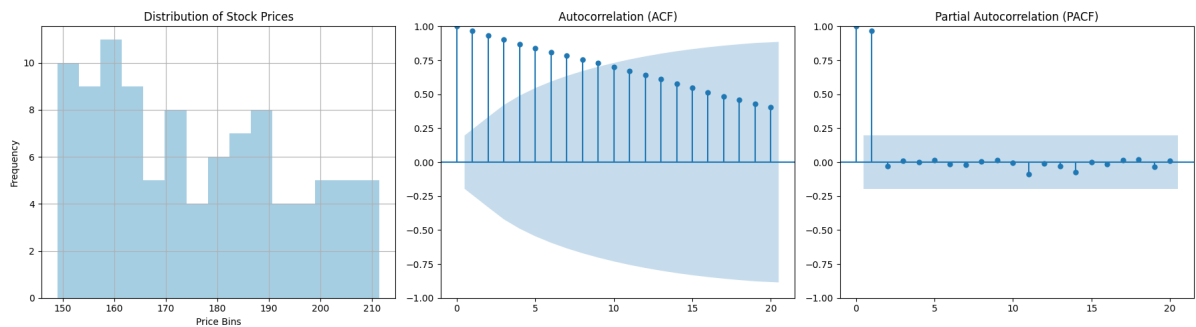


Figure 2: Statistical analysis of the time series data.

2.2 Model Architectures

2.2.1 ARMA Benchmark Model

A standard linear Autoregressive (AR) model, $\text{ARIMA}(5, 0, 0)$, was implemented as a baseline using the `statsmodels` library.

2.2.2 NLARMA Model

The NLARMA model uses a Multi-layer Perceptron (MLP) neural network. Figure 3 illustrates the data flow for the optimized model. It takes the five previous prices as input, processes them through a hidden layer with 100 neurons using the ‘tanh’ activation function, and outputs a single predicted price.

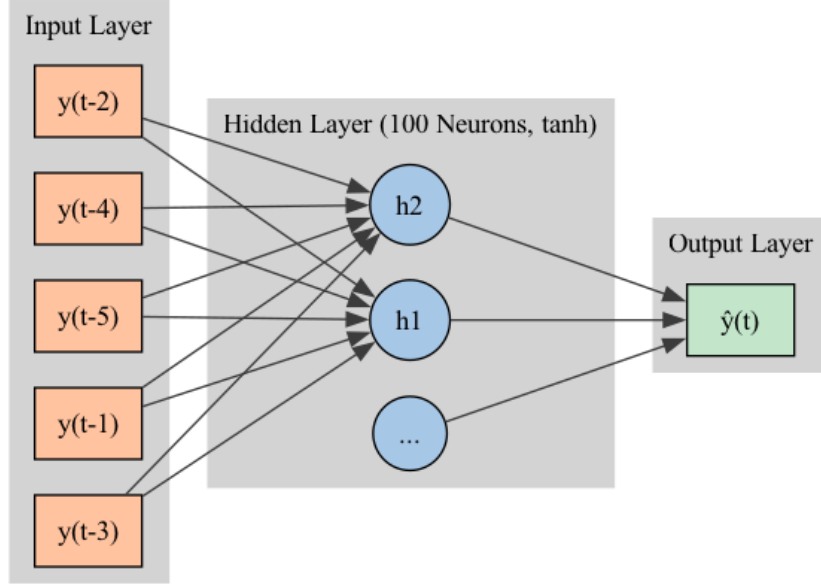


Figure 3: Architecture of the Optimized NLARMA Model.

2.2.3 Hyperparameter Optimization

A neural network’s performance is highly dependent on its configuration, known as hyperparameters. To move beyond a baseline model and find the most effective architecture for our NLARMA model, we employed a systematic optimization process using `GridSearchCV` from the `scikit-learn` library.

Grid Search works by defining a “grid” of potential hyperparameter values and then exhaustively training and evaluating a model for every possible combination. To ensure our evaluation was robust and suited for time series data, we used a 3-fold time-series cross-validation strategy. This process systematically identified the combination of parameters that minimized the Root Mean Squared Error (RMSE) on the validation folds.

The parameter grid evaluated was:

- **Hidden Layer Sizes:** Testing different network depths and widths with [(50,), (100,), (50, 50), (100, 50)].
- **Activation Functions:** Comparing the most common nonlinear functions, ['relu', 'tanh'].
- **Regularization (alpha):** Evaluating different strengths of L2 regularization to prevent overfitting, with [0.0001, 0.001, 0.01].
- **Learning Rate Schedule:** Testing an adaptive learning rate to improve training stability.

The grid search concluded that the optimal configuration for this specific dataset was an MLP with a single hidden layer of 100 neurons, using the ‘tanh’ activation function, and a regularization alpha of ‘0.0001’. This optimized architecture was used for our final NLARMA model.

3 Results and Evaluation

3.1 Performance Benchmarking

The models were evaluated on the test set using Root Mean Squared Error (RMSE). The results in Table 2 show a clear progression of improvement.

Table 2: RMSE Comparison of Models (Lower is Better)

Model	RMSE	Remarks
ARMA (ARIMA(5,0,0))	11.6988	High error due to linearity; performs poorly.
NLARMA (Un-optimized)	3.8417	Captures nonlinearities; substantially better than ARMA.
NLARMA (Optimized)	1.6559	Best result after hyperparameter tuning.

The linear ARMA model serves as a poor baseline ($\text{RMSE} \approx 11.7$). The un-optimized NLARMA model immediately provides a significant improvement ($\text{RMSE} \approx 3.84$). Finally, hyperparameter tuning dramatically reduces the error further to just **1.66**.

3.2 Visual Analysis

3.2.1 Initial Un-Optimized Model Performance

Figure 4 shows the full forecast of the initial NLARMA model. Figure 5 provides a zoomed-in view, comparing it with the ARMA model. While far better than ARMA, some deviations from the actual data are visible.

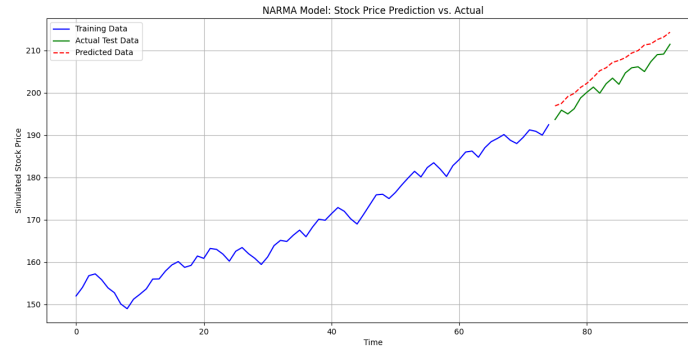


Figure 4: Un-optimized NLARMA model predictions vs. actual data.

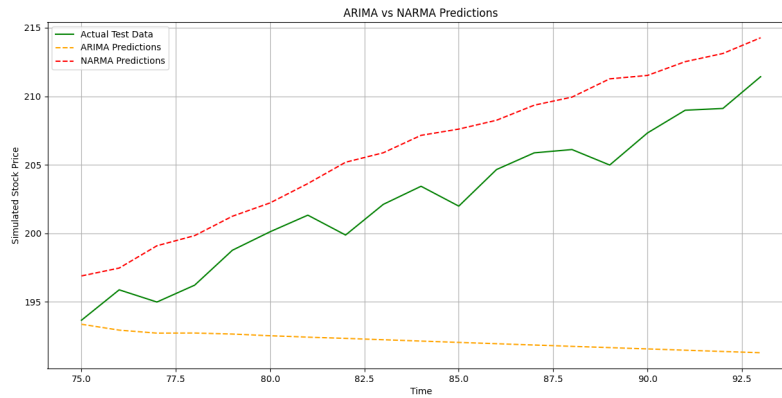


Figure 5: Zoomed-in comparison of the Un-optimized NLARMA and ARMA models.

3.2.2 Optimized Model Performance

After optimization, the model's predictions align much more closely with the actual data. Figure 6 shows the forecast, and Figure 7 confirms that the optimized NLARMA predictions (red dashes) track the true values (green) with impressive accuracy, far surpassing the ARMA model.

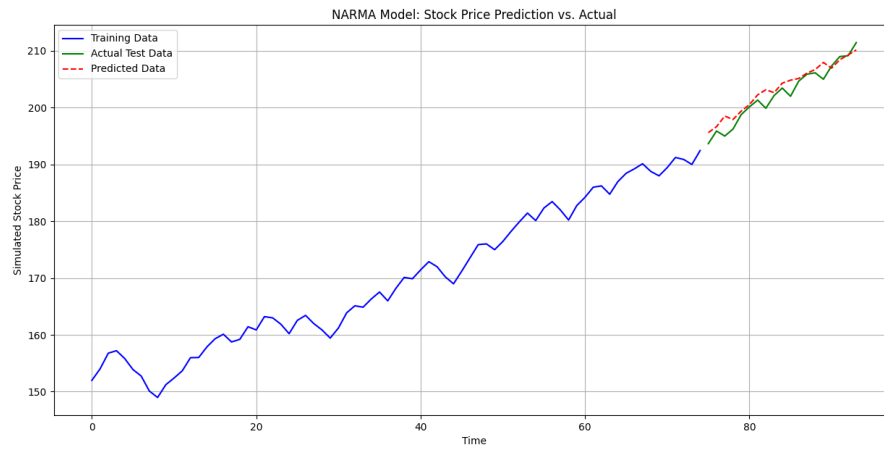


Figure 6: Optimized NLARMA model predictions vs. actual data.

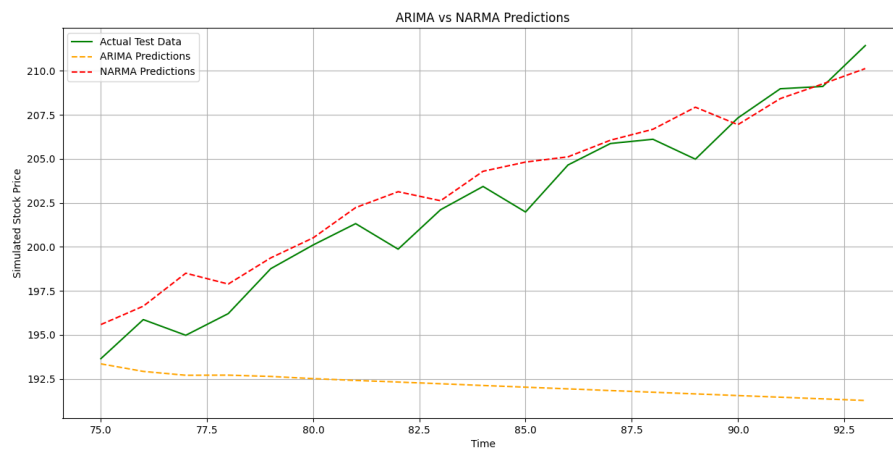


Figure 7: Zoomed-in comparison of the Optimized NLARMA and ARMA models.

3.2.3 Residual Analysis

The residual plot in Figure 8 powerfully illustrates the performance gap. The ARMA model’s errors are large and systematic. In contrast, the optimized NLARMA model’s errors are much smaller and are randomly scattered around the zero line, indicating a well-fitted and unbiased forecast.

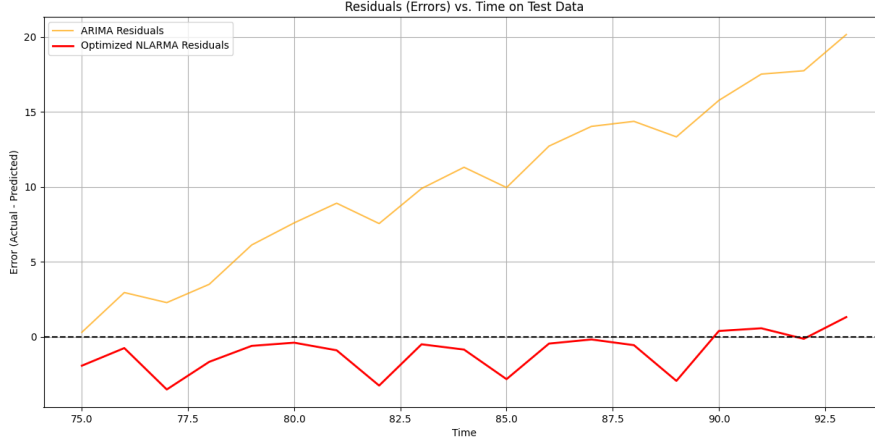


Figure 8: Residuals (Errors) vs. Time for the ARMA and Optimized NLARMA models.

3.3 Strengths and Limitations

- **Strengths:** The key strength of the NLARMA model is its **flexibility and power**. As universal function approximators, neural networks can capture arbitrarily complex nonlinear relationships and interactions within the data. This allows them to create far more accurate and nuanced forecasts for complex systems, which was clearly validated by the significantly lower RMSE in our study.
- **Limitations:** This power comes with several trade-offs:
 - **”Black Box” Nature:** Unlike ARMA, it is difficult to interpret **why** the neural network makes a specific prediction. The complex web of weights and biases does not offer a straightforward explanation, making it less transparent.
 - **Data Requirement:** Neural networks are ”data-hungry” and perform best with large datasets. Our dataset of 100 points is very small for a neural network, which increases the risk that the model might not generalize well to patterns it has not seen before.
 - **Computational Cost:** The process of training a neural network, and especially running a comprehensive grid search for hyperparameter optimization, is far more computationally intensive and time-consuming than fitting an ARMA model.
 - **Risk of Overfitting:** With their high flexibility, neural networks are more prone to overfitting—learning the noise in the training data rather than the underlying signal. While techniques like regularization and cross-validation mitigate this risk, it remains a key concern, especially with limited data.

3.4 Future Work

This research could be extended in several promising directions:

- **Advanced Architectures:** Implementing more sophisticated recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU), which are specifically designed for sequence data, could yield even better performance.
- **Feature Engineering:** The current model only uses lagged prices as input. Incorporating additional features, such as moving averages, price volatility, or external market indicators, could provide the model with more context and improve its predictive power.

- **Larger Dataset:** The most crucial next step would be to train and evaluate the model on a much larger dataset (e.g., several years of daily prices) to ensure its robustness and test its generalizability over different market conditions.

4 Conclusion

In this comparative study, the nonlinear ARMA approach, implemented with a neural network, showed clear and significant advantages for forecasting a nonlinear time series. The optimized NLARMA model achieved an RMSE that was nearly 7 times lower than the linear ARMA benchmark. This demonstrates that for complex systems, allowing for nonlinearity is crucial for accuracy. The visual and residual analyses confirm that the NLARMA model not only produces lower overall error but also creates a more reliable and unbiased forecast. While NLARMA models require more careful tuning, their superior predictive power makes them an essential tool for modern time series analysis.

References

- [1] Yinka-Banjo, C. O., & Akinyemi, M. I. (2023). *Stock Market Prediction Using a Hybrid of Deep Learning Models*. International Journal of Financial Studies, Economics and Management, 2(2), 1–16.
- [2] Stojkoska, B., Taskovska, K., & Utkovski, Z. (2022). *Time Series Prediction with Neural Networks: A Review*. In 2022 57th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST) (pp. 1–4). IEEE.