Development Economics of China
2025, VOL. 9, NO. 7, 42-45
DOI: 10.47297/wspdecWSP2515-797308.20250907

A Review of Forecasting Tesla Stock Price Trends Using ARIMA Models in R

Jintao Liu

Qingdao Academy, Qingdao, Shandong, 266041, China

ABSTRACT

Forecasting Tesla (TSLA) stock prices is challenging due to significant price volatility and the complexity of financial markets. Among forecasting approaches, ARIMA models are favored due to their mature theoretical foundation and easily interpretable parameters. Increased computational power and data availability have enabled more sophisticated forecasting techniques. Methods for predicting Tesla stock prices from 2019 through 2024 are systematically examined in this review. The models considered include standard ARIMA, automated ARIMA (via the forecast package's auto.arima() function), and ARIMA-GARCH combinations that account for volatility clustering. The methodology blends literature analysis with empirical demonstrations. The principles of ARIMA modeling and their implementation in R are introduced, followed by an exploration of Tesla's stock price characteristics and preprocessing steps. The process of fitting these models and generating forecasts is demonstrated using historical price data. It is indicated that automated model selection streamlines the modeling process, while combining ARIMA with GARCH models significantly improves volatility forecasting. For instance, in one study a reduction in forecast error (RMSE) of roughly 15% was observed when using an ARIMA-GARCH model compared to a standalone ARIMA model. It is further demonstrated in comparative studies that ARIMA-GARCH models can reduce short-term forecast errors by roughly 10 – 20% relative to baseline methods (e.g. random walk models). Overall, ARIMA-based methods are found to be effective for short-term Tesla price forecasts when volatility is properly modeled. The findings suggest that hybrid approaches, which capture both linear trends and volatility dynamics, are shown to outperform simple ARIMA models. Future work may integrate deep learning techniques and multi-source data (such as macroeconomic indicators and sentiment analysis) to build more robust, adaptive forecasting systems.

KEYWORDS

ARIMA; Stock forecasting; Tesla; GARCH; R programming

1 Introduction

This paper provides a systematic review of research from 2019 to 2024 on forecasting Tesla ' s stock prices using ARIMA, Auto ARIMA, and ARIMA-GARCH models implemented in R, combining literature analysis with empirical demonstrations. It first introduces the principles of ARIMA and GARCH models and their implementation in R, then explores the statistical characteristics of Tesla ' s stock price data and the necessary preprocessing procedures. Building on this foundation, the study demonstrates model construction and forecasting through R code examples, including both Auto ARIMA and manual ARIMA approaches, and reviews empirical results from ARIMA-GARCH hybrid models. Finally, it discusses the strengths and limitations of these methods and proposes directions for future research.

2 ARIMA model principles and applications in financial time series

The ARIMA model is a classic tool in time series analysis, composed of three components:

(1) Autoregressive (AR): a linear combination of p lagged values is used to explain the current value.

(2) Integration (I): differencing is applied to remove non-stationarity, ensuring the series is stationary.

(3) Moving Average (MA): a linear combination of q lagged residuals is used to capture random shocks.

Let the original series be X_t. After applying d-th order differencing, a stationary series $Y_t = \Delta^d X_t$ is obtained. The ARMA(p,q) model can then be expressed as:

$$Y_t = \sum_{i = 1}^{p} \phi_i Y_{t - i} + \epsilon_t + \sum_{j = 1}^{q} \theta_j \epsilon_{t - j}$$

Here, ϵ_t is a white noise series with zero mean and constant variance. The model identification process typically involves the following steps:

Stationarity test: Test for stationarity using the ADF (Augmented Dickey-Fuller) and KPSS tests; if the series is non-stationary, apply differencing or log transformation.

Initial parameter identification: Estimate possible values of p and q by inspecting the ACF (Autocorrelation Function) and PACF (Partial Autocorrelation Function) plots.

Model selection criteria: Evaluate candidate models based on AIC or BIC and select the optimal parameter combination.

Diagnostic checking: Use the Ljung-Box test and residual analysis to evaluate the model fit, revising the model if necessary. In stock price forecasting, ARIMA models are primarily used to capture linear trends and short-term autocorrelation. For example, Zhang et al. (2020) applied an ARIMA(2,1,2) model to the CSI 300 index and achieved a 30-

day return RMSE below 1.8%, demonstrating a strong fit to trend changes. In contrast, Li Xiaohua (2022) conducted a comparative analysis of major US tech stocks (including Apple, Microsoft, and Tesla) and found that ARIMA models performed stably for short-term (5–10 day) forecasts but tended to lag during periods of extreme market volatility.

3 Auto ARIMA modeling in R: mechanism and practice

The auto.arima() function in the forecast package automates ARIMA parameter selection. Its core process includes:

(1) Determining differencing order d: Unit root tests (e.g., KPSS and ADF) are used to automatically determine the optimal number of differences.

(2) Traversing the parameter space: Within the specified ranges of p and q, stepwise search and heuristic algorithms are applied to fit candidate models.

(3) Selecting the optimal model: The best balance between fit and complexity is chosen based on criteria such as AICc or BIC.

(4) Residual checking: An automatic Ljung-Box test is performed to ensure that residuals are free of autocorrelation.

(5) Seasonality handling: If the series exhibits periodic patterns, seasonal differencing is automatically considered.

The auto. arima() function is convenient and efficient: researchers do not need to manually plot ACF/PACF or try multiple parameter combinations. It can also be applied to automate modeling for multiple stocks or market indices simultaneously. The function handles differencing, seasonality, and inclusion of constant or trend terms automatically. For example:

```
library(forecast)
f <- auto.arima(my_ts)
summary(f)
plot(forecast(f, h = 20))
```

This example shows automatic ARIMA modeling of a time series my_ts and plots the 20-period-ahead forecast.

Tesla Stock Price Characteristics and Data Preprocessing

Tesla has experienced significant price run-ups and pullbacks due to its innovative technology and market expectations since its IPO. Key events that have influenced its stock price include:

Production capacity expansion: Launch of the Shanghai Gigafactory (2020) and facilities in Berlin and Texas (2021) increased supply expectations.

Positive earnings surprises: Multiple quarterly earnings reports exceeded market expectations, leading to rapid short-term price jumps.

Macro factors: Changes in policy subsidies, commodity price fluctuations, and the impact of the COVID-19 pandemic have all significantly affected Tesla's stock price.

In data processing, historical Tesla (TSLA) stock data can be obtained, for example, from Yahoo Finance using the quantmod package:

```
library(quantmod)
tsla <- getSymbols("TSLA", src = "yahoo", from = "2019-01-01", to = "2024-12-31", auto.assign = FALSE)
close <- Cl(tsla)
```

Next, handle missing values and take logarithms:

```
close_filled <- na.locf(close) # Carry forward last observation to fill NAs
log_close <- log(close_filled) # Log transformation
```

We then compute log returns and ensure stationarity:

```
returns <- diff(log_close) # Log returns
library(tseries)
adf.test(log_close) # Test stationarity of the log price
nd <- ndiffs(log_close) # Determine number of differences needed
diffed <- diff(log_close, differences = nd)
adf.test(diffed) # Test stationarity of the differenced series
```

Finally, we split the data into training and testing sets. For example, use data from January 1, 2019 to December 31, 2023 as the training set, and data from January 1, 2024 to December 31, 2024 as the testing set.

4 ARIMA and Auto ARIMA Model Building Examples

We now demonstrate both automatic and manual ARIMA modeling in R for Tesla's log-close price series: library (forecast)

```
# Automatic ARIMA modeling
auto_model <- auto.arima(log_close, seasonal = FALSE, ic = "aicc")
summary(auto_model)
# Manual ARIMA modeling (e.g., ARIMA(2,1,2))
manual_model <- Arima(log_close, order = c(2,1,2), include.constant = TRUE)
summary(manual_model)
# Model diagnostics
checkresiduals(auto_model)
checkresiduals(manual_model)
# Forecasting and evaluation
auto_fc <- forecast(auto_model, h = 365)
manual_fc <- forecast(manual_model, h = 365)
accuracy(auto_fc, test_data)
accuracy(manual_fc, test_data)
```

The above example shows fitting, diagnosing, and forecasting using both automatic and manual ARIMA models, and compares their forecasting performance.

5 ARIMA-GARCH combined model and empirical results

The GARCH (Generalized Autoregressive Conditional Heteroskedasticity) model focuses on modeling the conditional variance of a time series, capturing the volatility clustering often seen in financial data.

R Implementation

In R, we can fit an ARIMA-GARCH model by first obtaining residuals from an ARIMA mean model, and then fitting a GARCH model to those residuals using the rugarch package:

```
library(rugarch)
# Assume auto_model is the fitted ARIMA mean model and we have residuals
residuals <- residuals(auto_model)
# Specify a GARCH(1,1) model for the residuals
garch_spec <- ugarchspec(
mean.model = list(armaOrder = c(0,0)), # No additional ARMA terms in the mean
variance.model = list(model = "sGARCH", garchOrder = c(1,1)),
distribution.model = "std" # Student's t-distribution for innovations
)
# Fit the GARCH model
garch_fit <- ugarchfit(spec = garch_spec, data = residuals)
show(garch_fit)
# Combined forecasting
garch_fc <- ugarchforecast(garch_fit, n.ahead = 365)
```

6 Empirical findings

Empirical studies combining ARIMA and GARCH models have shown improved forecasting performance on Tesla stock data:

Wang [3] found that an ARIMA(2,1,2)-GARCH(1,1) model applied to 2020–2021 Tesla data reduced the forecasting RMSE by about 15% compared to a single ARIMA model.

Li [4] introduced an EGARCH model to capture asymmetric volatility, further improving forecast accuracy for both upward and downward movements.

Model Evaluation Metrics and Comparative Analysis

Common metrics and criteria for evaluating forecasting models include:

Information criteria: AIC and BIC measure model fit quality relative to complexity.

Prediction error metrics: RMSE (Root Mean Square Error), MAE (Mean Absolute Error), MAPE (Mean Absolute Percentage Error), and R2 are used to quantify forecasting performance.

Benchmark comparisons: Experiments comparing ARIMA-GARCH with simple benchmarks and machine learning models (such as random walk, naive methods, LSTM, and SVR) have been conducted. For example:

ARIMA-GARCH has been shown to improve RMSE by 10%–20% for short-term (30–90 days) forecasting.

LSTM (a deep learning model) often outperforms traditional models when large amounts of data are available, but can be less stable when data are scarce or noisy.

7 Discussion and future work

Hybrid models: Combining ARIMA with machine learning (e.g., ARIMA-LSTM, ARIMA-SVR) may capture both linear and nonlinear patterns effectively.

Multisource data integration: Incorporating macroeconomic indicators, industry indices, news sentiment, and social media signals for multidimensional forecasting.

High-frequency trading: Modeling minute-by-minute or tick data to explore microstructure effects and order flow information.

Real-time online learning: Implementing online updating algorithms and automated monitoring to enable periodic model retraining and performance tracking in real time.

8 Conclusion

This paper has reviewed the application of ARIMA-based models in R for short-term Tesla stock price forecasting. By presenting theoretical principles, R code examples, and literature comparisons, we have highlighted the automation advantages of auto. arima() and the effectiveness of ARIMA-GARCH combined models in volatility modeling. Future research may further integrate deep learning methods and heterogeneous data sources to build more accurate, robust, and updatable forecasting systems, providing stronger technical support for investment decisions and risk management.

References

[1] Zhang J., et al. (2020). Research on forecasting the CSI 300 index using an ARIMA model. Financial Engineering.

[2] Li X. (2022). Forecast and comparative analysis of technology stock prices. International Finance.

[3] Wang H. (2022). Study on electric vehicle industry stock volatility combined with GARCH models. Quantitative & Technical Economics Research.

[4] Li J. (2023). Application of EGARCH in highly volatile financial time series. Statistical Research.

[5] Hyndman R. J., Koehler A. B., Ord K. M., & Snyder R. D. (2008). Forecasting with Exponential Smoothing: The State Space Approach. Springer, Berlin.

[6] Hyndman R. J., & Khandakar Y. (2008). Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, 27(3), 1–22.