ECONOMETRICS
EKONOMETRIA
Advances in Applied Data Analysis

Econometrics. Ekonometria.
Advances in Applied Data Analysis
Year 2025, Vol. 29, No. 4
ISSN 2449-9994
journals.ue.wroc.pl/eada

Enhancing Forecast Accuracy: The Impact of Data Transformation in Time Series Models

Andrzej Dudek

Wroclaw University of Economics and Business

e-mail: andrzej.dudek@ue.wroc.pl

ORCID: 0000-0002-4943-8703

Quote as: Dudek, A. (2025). Enhancing Forecast Accuracy: The Impact of Data Transformation in Time Series Models. *Econometrics. Ekonometria. Advances in Applied Data Analysis*, *29*(4), 33-40.

DOI: 10.15611/eada.2025.4.03

JEL: C22, C53, C18

Abstract

Aim: The aim of the article was formulate suggestions on which preprocessing method is preferable for various forecasting algorithms, including machine learning approaches, particularly for forecasting stock values.

Methodology: Research study on actual stock values prediction on an example of 10 average NYSE enterprises, comparing five scenarios of data preparation.

Results: The results confirm theoretical assumptions and recommendations for the proper design of benchmark studies and real forecasting models.

Implications and recommendations: As stated in the literature of the subject data transformation for models based on stochastic processes, such as ARIMA and GARCH, transforming data to rates of return (a form of differentiation) is a desirable approach. For machine learning models, especially recurrent neural networks, such as the Long Short-Term Memory Network and the Gated Recurrent Unit, the min-max normalisation data transformation should be applied. For exponential Smoothing and Brownian motion methods, the best results were achieved for non-transformed (raw) data. The guidance relevant to benchmark studies and real forecasting models is presented in the final section of the paper. The central thesis was to emphasise, through the example of stock values forecasting, that proper benchmark studies and real-life applications should be designed in a way that ensures proper preprocessing is used for the given model. Using the same preprocessing for different models may sometimes yield misleading results.

Originality/value: The topic of data preparation and transformation, although commonly present in the literature of the subject, is rarely confirmed by research studies on real datasets. To the best of the author's knowledge, this type of analysis has not been conducted on real data to date.

Keywords: forecasting, stochastic process models, recurrent neural networks

1. Introduction

Data preprocessing plays a crucial role in the accuracy and reliability of forecasting tasks, especially when using deep learning models. One of the essential steps is data transformation, which ensures that the input time series is appropriately prepared for modelling. Direct comparisons of forecasting methods using identical raw datasets can lead to misleading conclusions, as each model type has different requirements for input structure. Thus, comparing model outputs without accounting for proper reverse transformation of the data can distort benchmark results. In particular, the use of raw data is typically unsuitable for deep learning models without preprocessing, as these models require normalised or integrated inputs to learn effectively.

Statistical forecasting models such as ARIMA and GARCH, on the other hand, inherently assume certain properties such as stationarity, and thus necessitate integration in cases of non-stationarity – an almost ubiquitous condition in real-world time series e.g. stock market prices and energy consumption. Consequently, the need for an appropriate transformation of input data is not only critical but also method-dependent, therefore the first and foundational step in any meaningful benchmark analysis should be the tailored preprocessing of data aligned with the requirements of the respective modelling approach.

It the paper, five preprocessing strategies for time series forecasting: raw data, min-max normalised data, integrated data (using rates of return), integrated and scaled to percentage form, and finally, integrated followed by min-max normalisation are compared based on real stock data time datasets.

2. Literature Review

When modelling stock price time series with ARIMA or GARCH family models, the use of raw prices is generally discouraged due to their strong non-stationarity and stochastic trends. Financial time series often exhibit unit roots, meaning that their statistical properties, such as mean and variance, change over time, violating the stationarity assumption required by ARIMA and GARCH processes (Box et al., 2016; Tsay, 2010). To address this issue, returns ratios – typically defined as logarithmic differences of prices – are commonly employed. Rates of return tend to stabilise the variance, reduce the presence of autocorrelation in levels, and transform the series into a stationary process more suitable for linear and nonlinear time-series modelling (Hamilton, 1994). This transformation is critical, as ignoring non-stationarity may lead to spurious regression results and misleading forecasts (Enders, 2015).

Moreover, the differentiation between price levels and rates of return is significant when volatility forecasting is the primary objective, as in GARCH-type models. GARCH models are designed to capture volatility clustering and conditional heteroskedasticity, features that are prominent in rates of return series but much less visible in raw prices (Engle, 1982; Bollerslev, 1986). Using rates of return or log-differenced series not only ensures stationarity but also amplifies the presence of volatility patterns necessary for practical estimation of conditional variance dynamics. Therefore, transformations such as first differencing or return rates calculation are indispensable preprocessing steps for both ARIMA-based forecasts of mean dynamics and GARCH-family models focused on volatility structures in financial time series (Francq & Zakoïan, 2019).

However, recurrent neural networks, such as the Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, have become popular tools for forecasting stock time series due to their ability to capture nonlinear temporal dependencies, whilst financial data, particularly stock prices and returns, typically exhibit large numerical ranges, volatility clustering, and heavy-tailed distributions, which can impede neural network training if left unscaled. Neural architectures are susceptible to input magnitude, and unscaled data often lead to exploding or vanishing gradients, slower convergence, and suboptimal weight updates (Goodfellow et al., 2016; Chollet & Allaire, 2018). Scaling techniques such as min–max normalisation or z-score standardisation improve the stability of gradient-based optimisation, ensure numerical homogeneity across features, and enhance the model's ability to learn temporal dependencies (Brownlee, 2019). Thus, data preprocessing through normalisation is the best practice as well as a necessary condition for reliable training of LSTM and GRU models on stock time series.

Apart from numerical stability, normalisation also improves predictive accuracy in financial forecasting tasks. Since neural networks internally rely on activation functions that operate efficiently within limited value ranges, normalised inputs help preserve dynamic patterns without distortion, allowing models to capture subtle variations in stock returns more effectively (Zhang et al., 2020). In empirical studies, normalised datasets consistently yield superior forecasting performance compared to raw data, particularly in volatile financial environments where large fluctuations can dominate raw sequences (Fischer & Krauss, 2018). Therefore, normalisation should be regarded as a critical preprocessing step that directly affects the convergence, generalisation ability, and forecasting accuracy of deep recurrent models in stock market applications.

3. Methodology

To confirm the recommendations from the subject literature, the author conducted the following research simulation study based on real stock time data series.

Ten companies listed on the stock exchange were selected in line with the following criteria:

1. **Market Cap Range:** these companies are large (typically \$20B–\$100B) but not among the top 50 mega-caps (e.g. Apple, Microsoft).
2. **Diverse Sectors:** includes industrials, materials, finance, and real estate to avoid overconcentration.
3. **Established Presence:** All are well-known, stable companies with substantial NYSE trading volumes.

The companies actually selected for the study are presented in Table 1.

Table 1. Companies whose stock values were used in the study

Ticker	Company Name	Description
DHR	Danaher Corporation	A global science and technology innovator specialising in healthcare, environmental, and industrial solutions.
PLD	Prologis, Inc.	A leading real estate investment trust (REIT) focused on logistics and warehouse properties worldwide.
SHW	Sherwin-Williams Company	One of the largest producers of paints, coatings, and related products globally.
APD	Air Products and Chemicals, Inc.	A multinational industrial gases and chemicals company serving various industries.
SYY	Sysco Corporation	The world's largest food distribution company, serving restaurants, healthcare, and educational facilities.
AFL	Aflac Incorporated	A Fortune 500 insurer known for supplemental health and life insurance policies.
EFX	Equifax Inc.	A major consumer credit reporting agency providing data analytics and risk assessment services.
VMC	Vulcan Materials Company	The largest U.S. producer of construction aggregates (crushed stone, sand, and gravel).
EMR	Emerson Electric Co.	A diversified technology and engineering company specialising in automation and industrial solutions.
HWM	Howmet Aerospace Inc.	A manufacturer of engineered metal products for aerospace, defence, and transportation industries.

Source: own research based on Yahoo Finance repository.

Five data preparation strategies were compared (note that the abbreviations are used in Tables 2 to 7):

- **(RAW)** raw data,
- **(NORMALISED)** min-max normalised data according to formula,

$$x_t^* = \frac{x_t - min(x)}{max(x) - min(x)},$$

where

- $\boldsymbol{x_t}$ – the original value of the time series at time t,
- $\mathbf{min}(\boldsymbol{x})$ – the minimum value of the entire time series (or the chosen window),
- $\mathbf{max}(\boldsymbol{x})$ – the maximum value of the entire time series (or the chosen window),
- $\boldsymbol{x_t^*}$ – the normalised value scaled to the range [0, 1],
- **(RATES_OF_RETURN)** integrated (rates of return instead of values) data calculated by the following formula

$$r_t = ln(P_t) - ln(P_{t-1}),$$

where

- $\boldsymbol{r_t}$ – *logarithmic rate of return* (log-return) between periods $t-1$ and t,
- $\boldsymbol{P_t}$ – *asset price* (or index value) at time t,
- $\boldsymbol{P_{t-1}}$ – *asset price* at the previous time period,
- **(RATES_PERCENT)** integrated and scaled to percents (rates of return multiplied by 100%) data,
- **(NORMALISED RATES_OF_RETURN)** integrated and then min-max normalised data.

For each enterprise, the following methods were deployed: ARIMA model(Box & Jenkins, 1970), GARCH Model (Engle, 1982; Bollerslev, 1986), Brownian motion (Einstein, 1905; Wiener, 1923), Exponential Smoothing (Holt, 1957; Winters, 1960), and Long Short-Term Memory neural network (Hochreiter & Schmidhuber, 1997). In addition, the Gated Recurrent Unit (GRU) neural network (Cho et al., 2014) was used for forecasting seven-session values. The learning period was established to 50, 100, 200, 300, 500, and 1000 days and seven consecutive stock values were used for estimating the quality of forecasting.

To evaluate the quality of forecasting in each data preprocessing scenario, forecasts for the subsequent seven days, based on 50, 100, 200, 300, 500, and 1000 days, were estimated for each method and each preprocessing scenario. The usual measures, Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error, were used for evaluation.

4. Results

Table 2 presents the results of a simulation study for the ARIMA Model (Box & Jenkins, 1970). The lowest values of Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error were obtained for the scenario in which data were transformed by calculating return ratios (which may be treated as a form of differentiation).

Table 2. The simulation study results for the ARIMA model

	STRATEGY	RMSE	MAE	MAPE
ARIMA	RAW	1.703619	1.466516	1.240773
	RATES_PERCENT	1.413970	1.168053	0.989860
	RATES_OF_RETURN	**1.325411**	**1.066128**	**0.904349**
	NORMALISED	1.515548	1.258586	1.074684
	NORMALISED RATES_OF_RETURN	1.987031	1.651763	1.396822

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras, sci-kit learn* python packages.

Table 3 presents the simulation study result for the Brownian motion (Einstein, 1905; Wiener, 1923) method. The best results were for raw data, although this may not come as a surprise, since the overall results were worse than those for other methods/models.

Table 3. The simulation study results for the Brownian motion method

	STRATEGY	RMSE	MAE	MAPE
BROWNIAN	RAW	**6.9271061**	**6.4963058**	**5.5047914**
	RATES_PERCENT	11.7456434	11.7447571	9.9846868
	RATES_OF_RETURN	11.7633319	11.7625180	9.9998469
	NORMALISED	11.7346792	11.7337449	9.9752872
	NORMALISED RATES_OF_RETURN	11.7423766	11.7414762	9.9818863

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras, sci-kit learn* python packages.

Table 4 presents the results for five data preparation scenarios of the GARCH model (Engle, 1982; Bollerslev, 1986). The lowest error coefficients were obtained in procedures that included integration. The rates of return calculation and rates of return multiplied by 100 percent may be declared the best methods, while the difference between them was insignificant.

Table 4. The simulation study results for the GARCH model

	STRATEGY	RMSE	MAE	MAPE
GARCH	RAW	1.987084	1.700064	1.437339
	RATES_PERCENT	**1.588790**	**1.342292**	**1.136150**
	RATES_OF_RETURN	**1.588994**	**1.342695**	**1.136499**
	NORMALISED	1.810632	1.347512	1.165044
	NORMALISED RATES_OF_RETURN	2.072017	1.720957	1.455080

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras, sci-kit learn* python packages.

Table 5 presents the results for the gated recurrent unit (GRU) neural network (Cho et al., 2014). The lowest values of Root Mean Square Error, Mean Absolute Error, and Mean Absolute Percentage Error were found for the scenario in which the data were min-max normalised, showing a significant difference compared to other data preparation scenarios.

Table 5. The simulation study results for the Gated Recurrent Unit model

	STRATEGY	RMSE	MAE	MAPE
GRU	RAW	2.201235	1.914494	1.622920
	RATES_PERCENT	3.120753	2.399227	2.026249
	RATES_OF_RETURN	2.408659	2.041601	1.725056
	NORMALISED	**1.493877**	**1.198476**	**1.024842**
	NORMALISED RATES_OF_RETURN	2.131069	1.778118	1.503224

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras, sci-kit learn* python packages.

For Holt-Winters exponential smoothing (Holt, 1957; Winters, 1960), the best strategy was to leave raw data unchanged, however the error differences in Table 6 were relatively bigger than those for other models/methods.

Table 6. The simulation study results for Holt Winters exponential smoothing

	STRATEGY	RMSE	MAE	MAPE
HOLT_ WINTERS	RAW	**1.657260**	**1.413981**	1.196456
	RATES_PERCENT	1.765113	1.535244	1.301922
	RATES_OF_RETURN	1.766970	1.537145	1.303544
	NORMALISED	1.724796	1.472040	**1.186088**
	NORMALISED RATES_OF_RETURN	1.856842	1.552035	1.312818

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras*, *sci-kit learn* python packages.

For the Long Short-Term Memory (LSTM) network (Hochreiter & Schmidhuber, 1997), the second type of recurrent neural network analysed, the results (see Table 7) indicated a slight superiority of the min-max normalised approach. However, its advantage over the rates of return ratios strategy was marginal.

Table 7. The simulation study results for the Long Short-Term Memory model

	STRATEGY	RMSE	MAE	MAPE
LSTM	RAW	2.363669	2.013477	1.703297
	RATES_PERCENT	3.420169	2.909002	2.460272
	RATES_OF_RETURN	1.533231	1.286535	1.091071
	NORMALISED	**1.493877**	**1.198476**	**1.024842**
	NORMALISED RATES_OF_RETURN	1.978351	1.639916	1.386805

Source: own calculations with *yfinance, numpy, pandas, statsmodel, keras, sci-kit learn* python packages.

The author would like to emphasize at this point that the study aimed to compare data preparation strategies, and not the models' forecasting abilities as such (already examined in many studies).

The research results were sufficient to offer some suggestions for the proper design of benchmark and real forecasting studies (namely, to confirm the theoretical recommendations from the subject literature).

5. Discussion and Conclusions

The results obtained across the five preprocessing strategies and six forecasting models highlight the critical role of model-specific data transformation in determining predictive accuracy. In particular, the study confirms long-established theoretical expectations regarding the necessity of differencing for ARIMA and GARCH processes, the importance of normalisation for recurrent neural networks such as LSTM and GRU, and the relative robustness of exponential smoothing and Brownian motion methods when applied to raw input series.

Nevertheless, several limitations should be acknowledged. The analysis focused on ten large, well-established NYSE companies, which may reduce generalisability to small-cap firms or high-volatility markets. Only univariate forecasting models were examined, despite the growing relevance of multivariate methods in financial prediction. Additionally, the study relied on a fixed forecast horizon of seven sessions, and alternative horizons may yield different comparative performance outcomes. Finally, while min–max normalisation was shown to be effective for neural networks, other scaling techniques such as z-score normalisation or robust scaling were not tested, yet may provide useful comparative insights.

Future research could therefore extend the present work in several directions. First, applying the same preprocessing comparison framework to other asset classes – including commodities, cryptocurrencies, and energy demand–would allow for broader validation. Second, incorporating additional transformations (Box–Cox, Yeo–Johnson, detrending) could illuminate how alternative approaches interact with linear and nonlinear models. Third, multivariate forecasting frameworks should be evaluated to reflect more complex real-world decision environments. Lastly, testing multiple forecast horizons and alternative evaluation metrics may reveal new dimensions of model sensitivity to preprocessing choices.

It is also important to emphasize that the choice of a data transformation strategy was influenced by the forecasting method and also by its internal architecture and parameterisation. For example, in LSTM networks the preferred input scaling depends on the activation function used in the recurrent units: sigmoid-based architectures typically benefit from inputs scaled to the range [0.2, 0.8], whereas tanh activations operate most effectively when values fall within approximately [-0.8, 0.8]. These activation-dependent constraints influence how efficiently gradients propagate during training, thereby affecting convergence speed and forecast accuracy but do not contradict the fact that for this type of model, normalisation is the optimal strategy. Consequently, the relationship between preprocessing and forecasting performance must be interpreted in light of both model structure and hyperparameter choices.

- Data preparation for GARCH and ARIMA requires the integration step.
- The best strategy for LSTM and GRU is to min-max normalise raw data.
- Brownian motion and exponential smoothing do not require the transformation step.

It is worth noting that benchmark analyses comparing forecast measures with the same data preparation step are not entirely factually correct as every method needs a different preparation, and only the result after reverse transformation should be compared. Similarly as with real forecasting procedures, data preparation and transformation are crucial for the final quality.

References

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, *31*(3), 307-327.

Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis: Forecasting and control*. Holden-Day.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2016). *Time series analysis: Forecasting and control* (5th ed.). Wiley.

Brownlee, J. (2019). *Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

Cho, K., van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation* (pp. 103-111). Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-4012

Chollet, F., & Allaire, J. J. (2018). *Deep learning with R*. Manning.

Einstein, A. (1905). On the movement of small particles suspended in stationary liquids required by the molecular-kinetic theory of heat. *Annalen der Physik*, *17*, 549-560.

Enders, W. (2015). *Applied econometric time series* (4th ed.). Wiley.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of UK inflation. *Econometrica*, *50*(4), 987-1007.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research, 270*(2), 654-669. https://doi.org/10.1016/j.ejor.2017.11.054

Francq, C., & Zakoïan, J.-M. (2019). GARCH models: Structure, statistical inference and financial applications (2nd ed.). Wiley.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Hamilton, J. D. (1994). *Time series analysis.* Princeton University Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holt, C. C. (1957). *Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages. ONR Memorandum*, Vol. 52. Carnegie Institute of Technology.

Tsay, R. S. (2010). *Analysis of financial time series* (3rd ed.). Wiley.

Wiener, N. (1923). Differential space. *Journal of Mathematics and Physics*, *2*(1), 131-174.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, *6*(3), 324-342.

Zhang, D., Chen, S., & He, Z. (2020). A deep learning framework for financial time series using stacked autoencoders and long–short term memory. *PLOS ONE, 15*(1), e0226997. https://doi.org/10.1371/journal.pone.0226997

Poprawa trafności prognoz: rola transformacji danych w modelach szeregów czasowych

Streszczenie

Cel: Sformułowanie sugestii, która metoda transformacji danych jest preferowana dla różnych metod prognozowania, w tym podejść uczenia maszynowego, szczególnie w prognozowaniu wartości akcji.

Metodyka: Badanie dotyczące predykcji rzeczywistych wartości akcji na przykładzie dziesięciu średniej wielkości przedsiębiorstw notowanych na NYSE, porównujące pięć scenariuszy przygotowania danych.

Wyniki: Potwierdzenie założeń teoretycznych oraz rekomendacje dotyczące właściwego projektowania studiów porównawczych (benchmark) i rzeczywistych modeli prognostycznych.

Implikacje i rekomendacje: Zgodnie z literaturą przedmiotu, dla modeli opartych na procesach stochastycznych, takich jak ARIMA i GARCH, pożądaną transformacją jest przejście na stopy zwrotu. W przypadku podejść uczenia maszynowego, w szczególności rekurencyjnych sieci neuronowych typu Long Short Term Memory (LSTM) i Gated Recurrent Unit (GRU), zalecana jest transformacja danych przy użyciu skalowania min-max. Dla metod wygładzania wykładniczego i ruchu Browna najlepsze wyniki osiągnięto dla danych nieprzekształconych (surowych). Artykuł potwierdził te teoretyczne rekomendacje na podstawie badaniach rzeczywistych cen akcji. W końcowej części artykułu zostały przedstawione sugestie dla badaczy przygotowujących studia porównawcze i rzeczywiste modele prognostyczne.

Oryginalność/wartość: Temat przygotowania i transformacji danych, choć powszechnie obecny w literaturze przedmiotu, rzadko jest potwierdzany przez badania na rzeczywistych zbiorach danych. Według wiedzy autorów, tego typu analiza nie była dotąd przeprowadzona na podstawie danych giełdowych.

Słowa kluczowe: prognozowanie, modele procesów stochastycznych, rekurencyjne sieci neuronowe