Contents lists available at GrowingScience

International Journal of Data and Network Science

homepage: www.GrowingScience.com/ijds

The comparison stateless and stateful LSTM architectures for short-term stock price forecasting

Anna Chadidjah[a], I Gede Nyoman Mindra Jaya[a*] and Farah Kristiani[c]

[a]*Department of Statistics, Universitas Padjadjaran, Jl. Raya Bandung Sumedang km 21 Jatinangor, Sumedang 45363, Indonesia*
[b]*Department of Mathematics, Parahyangan University, Jl. Ciumbuleuit No. 94, Hegarmanah, Kec. Cidadap, Kota Bandung 40141, Indonesia*

C H R O N I C L E

Article history:
Received: November 1, 2023
Received in revised format: November 25, 2023
Accepted: January 16, 2024
Available online: January 16, 2024

Keywords:
Time series
Forecasting
RNN
LSTM
Stateless
Stateful
Apple stock price

A B S T R A C T

Deep learning techniques are making significant contributions to the rapid advancements in forecasting. A standout algorithm known for its ability to produce accurate forecasts by recognizing temporal autocorrelation within the data is the Long Short-Term Memory (LSTM) algorithm, a component of Recurrent Neural Networks (RNN). The LSTM method employs both stateless and stateful architecture approaches, providing versatility in its application. This research aims to compare stateful and stateless algorithms in LSTM models, focusing on forecasting stock prices, such as those of Apple Inc. This comparative analysis is crucial, taking into account various characteristics of time series data, including the benefits and drawbacks of temporal autocorrelation. The comparison results reveal that, despite the stateful algorithm requiring more computational time, it achieves greater accuracy than the stateless approach. The forecast indicates a potential upward trend in share prices for the period of January to December 2024, according to the projected outlook for Apple's stock value. However, it is essential to exercise prudence in interpreting these results, considering that share price fluctuations are influenced by a significant number of variables.

1. Introduction

Forecasting is a complex field of statistical analysis that involves predicting future events, which is challenging due to the presence of uncertainties. The complexity arises from the intrinsic unpredictability of forthcoming occurrences. Forecasting methods frequently employ historical data to construct models that aim to produce precise predictions spanning numerous future time periods (Petropoulos, et al., 2022). The accuracy of forecasts depends on the quality of the data and the appropriateness of the methodology (Brownlee, 2019). Forecasting finds application in diverse fields, including forecasting of the stock prices (Lee & Kim, 2020). Accurate stock price forecasting is crucial for effective business planning, making the selection of an appropriate forecasting method a critical aspect of financial analysis. Traditional forecasting methods, such as the autoregressive (AR) model, moving average (MA) model, autoregressive moving average (ARMA) model, and autoregressive conditional heteroscedasticity (GARCH) method, have been widely employed to predict stock prices (Ho, Darman, & Musa, 2021). Nevertheless, the classical approach encounters limitations when forecasting data with irregular patterns. Stock price movements, influenced by a multitude of factors, pose a challenge for classical methods in effectively capturing complex temporal patterns (Wu, Xu, Chen, Li , & Zhao, 2022). The utilization of deep learning (DL) techniques, such as Recurrent Neural Networks (RNNs), became widely embraced for capturing complex temporal patterns (Moghar & Hamiche, 2020). RNNs stand as highly acclaimed deep learning algorithms that have gained immense popularity in recent years, finding broad applications across diverse fields, with a notable presence in forecasting. Rendered specifically for sequential processing, RNNs demonstrate exceptional performance by capturing the interconnections among elements in a sequence. When considering forecasting, the Long-Term Short-Term Memory (LSTM) technique becomes an increasingly common algorithm

* Corresponding author.
E-mail address: mindra@unpad.ac.id (I G. N. M. Jaya)

ISSN 2561-8156 (Online) - ISSN 2561-8148 (Print)

doi: 10.5267/j.ijdns.2024.1.009

(Moghar & Hamiche, 2020). LSTM models exhibit notable benefits in tasks related to predicting time series, especially in the presence of autocorrelation—the connection between a time series and its delayed versions. The effectiveness of these techniques stems from their capability to retain state information and identify patterns throughout the duration of the time series (Gülmez, 2023). The recurrent architecture facilitates the persistence and communication of states across weight updates as the epoch progresses.

Keras stands out as one of the most extensively used open-source packages, applicable to both Python and R environments. Keras supports both "stateful" and "stateless" operation modes for LSTMs. 'Stateful' architecture designates the final state of each sample in a batch as the initial state of the sample in the following batch. In contrast, during the normal or "stateless" mode, Keras performs a sample shuffle, resulting in the loss of dependencies between the lagged version of the time series and itself. Nevertheless, by leveraging the autocorrelations that are intrinsic to the time series, it is frequently possible to attain more precise outcomes when operating in "stateful" architecture (business-science.io, 2018); (Bismi, 2023). The decision to utilize an LSTM methodology is supported by two fundamental factors. To commence, the machine is provided with feedback on errors identified during the initial training phase to enhance the model. In addition, throughout the process, faults are continuously incorporated into the machine's gates. Additionally, LSTM networks demonstrate a significant lack of sensitivity to inter-event delays in time series data, which sets them apart from alternative artificial neural networks and forecasting methods (e.g., ARIMA) in the pursuit of an efficient unknown forecast model (Polyzo, Samitas, & Spyridou, 2021).

This research paper utilizes monthly data spanning from 1981 to 2023 to forecast Apple's stock prices from January to December 2024 through the application of machine learning. In addition to their primary function of financial forecasting, precise stock price predictions assist investors in mitigating potential losses in financial markets and provide governmental financial institutions with an essential financial early warning system. The study presents four fundamental contributions. In the first place, a comparison is made between stateful and stateless architectures in order to ascertain which framework is more efficient at forecasting stock prices, with a specific focus on those of Apple Inc. Additionally, it showcases the capability of LSTM networks to efficiently process arrival time series by utilizing their resilient autoregressive characteristics that have been calibrated using training data derived from analogous past occurrences. Furthermore, an illustration of backtesting, a widely utilized method in financial time series analysis, is provided in the paper as a means of validating the accuracy of stock price predictions. In conclusion, the research paper presents a projected valuation of Apple's stock for the period from January to December 2024.

The structure of the sections that follow is maintained throughout this paper. Section 2 examines pertinent scholarly works pertaining to deep learning, placing particular emphasis on LSTM. A comparative analysis of stateful and stateless LSTM architectures is presented in Section 3, utilizing Apple Inc. stock price data. A discussion and conclusions are presented in Section 4.

2. Deep Learning (DL)

Deep learning, a potent branch of machine learning, tackles the issue of predicting nonlinear outputs. The prediction output denoted as $\hat{Y}(\boldsymbol{X})$, from a high-dimensional input matrix $\boldsymbol{X} = [X_1, \dots, X_P]$, where P represents the number of inputs, and an output Y. Deep learning entails the investigation and development of a complex input-output mapping function (Dixon, Polson, & Sokolov, 2018):

$$\boldsymbol{Y} = F(\boldsymbol{X}), \text{ for } \boldsymbol{X} = [X_1, \dots, X_P] \tag{1}$$

The output $\boldsymbol{Y}$, can be continuous, discrete or mixed. The predictor consist of a multivariate functions $F(X)$ generated by superpositions of univariate semiaffine functions. A semiaffine function, denoted by $f^l_{W^l,b^l}$ is defined as (Dixon, Polson, & Sokolov, 2018)

$$f^l_{W^l,b^l}(X) \coloneqq f^l(W^l X + b^l) \tag{2}$$

where f^l is univariate and continuous. A nonlinear predictor is contracted using a sequence of layers L through a composite mapping:

$$\hat{Y}(X) \coloneqq F_{W,b}(X) = \left(f^L_{W^L,b^L} \circ \dots \circ f^1_{W^1,b^1}\right)(X) \tag{3}$$

where $W = (W^1, \dots, W^L)$ represents weight matrices and $b = (b^1, \dots, b^L)$ denotes offsets (Dixon et al. 2018). Let Z^L denote the l −th layer with $Z^0 = X$. The configuration of DL rule can then be expressed as a hierarchy of $L - 1$ unobserved layers, denoted as Z^l (Dixon, Polson, & Sokolov, 2018)

$$\begin{aligned} \hat{Y}(X) &= f^L(Z^{L-1}), \\ Z^1 &= f^1(W^1 Z^0 + b^1) \\ Z^2 &= f^1(W^2 Z^1 + b^2) \\ &\dots\dots\dots\dots \\ Z^{L-1} &= f^{L-1}(W^{L-1} Z^{L-2} + b^{L-1}) \end{aligned} \tag{4}$$

When Y is numeric, the output function $f^L(X)$ is given by the semiaffine function $f^L(X) \coloneqq f^L_{W^L,b^L}(\mathrm{X})$. When Y is categorical, $f^L(X)$ is a softmax function. The activation (or link) functions, i.e., $f^l, 1 \leq l < L$, are prespecified, whereas the weight matrices $W^l \in \mathbf{R}^{N_l \times N_{l-1}}$ and offset vectors $b^l \in \mathbf{R}^{N_l}$ have to be learned from a training data set $\left(\mathrm{X}^{(i)}, \mathrm{Y}^{(i)}\right)_{i=1}^{T}$. Common choices of f^l are hinge or rectified linear units, i.e, $\max(x, 0)$ and sigmoidal $(\cosh(x), \tanh(x))$ activation functions.

2.2.1 Recurrent Neural Networks (RNNs)

A recurrent neural network (RNN) is an artificial neural network type that can process sequence or time series data. RNNs work by passing data from one sequence step to the next via a loop. At each time step, the network takes an input vector and combines it with the hidden state vector from the previous time step to produce a new hidden state vector. This procedure is repeated until the entire sequence has been analyzed, at which point the final hidden state vector is obtained which can be utilized to make a prediction (Levin, 1990). RNNs employ recurrent layers to capture temporal dependencies with a modest number of parameters (West & Harrison, 1997). Using a recurrent layer and a feed-forward layer, they learn temporal dynamics by mapping an input sequence to a hidden state sequence and outputs (Dixon, Polson, & Sokolov, 2018).

Let Y_t represent the observed response and Z_t the hidden states, then the RNN model is given by:

Response: $$\hat{Y}_t = f^2(W_z^2 Z_t + b^2) \tag{5}$$

Hidden state: $$Z_t = f^1(W^1[Z_{t-1}, X_t] + b^1)$$

where f^1 denotes an activation function like $tanh(x)$, and f^2 is either a softmax function or an identity map. The choice of the activation function depends on whether the response is categorical or continuous. The weight matrices $W^1 = [W_z^1, W_x^1]$ and $W_z^2$are established during the network's training process and remain constant over time. X_t represents the inputs that are at the extremes up to k time lags. Z_{t-1}, where t is equal to 2, 3, and so on, denotes the hidden states from the previous time step. The hidden state is initialized at zero, meaning that Z_{t-k} is equal to 0 (Dixon, Polson, & Sokolov, 2018).

The primary differentiating factor between recurrent neural networks (RNNs) and feed-forward deep learning (DL) is the incorporation of a hidden layer that includes an autoregressive component, specifically denoted as $W_z^1 Z_{t-1}$. The outcome is a network configuration in which each layer corresponds to a specific time interval, denoted as t, highlighting the temporal aspect (Fig. 2).

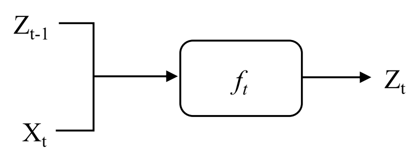


Fig. 2. Hidden layer of an RRN *Source:* (Dixon, Polson, & Sokolov, 2018).

2.2.2 Long Term Short Memory Neural Networks (LSTMNs)

Recurrent Neural Networks (RNNs) face challenges in acquiring knowledge of long-term patterns primarily because of the occurrence of vanishing and exploding gradients. These issues arise when gradients are backpropagated through the numerous unfolded layers of the network. In order to tackle the problem of gradients that either vanish or explode, (Hochreiter & Schmidhuber, 1997) introduced a particular variant of recurrent neural network called LSTMN. An LSTM (Long Short-Term Memory) network incorporates a memory unit that allows the network to identify and disregard irrelevant previous states. To achieve this objective, the hidden state is produced by a hidden cell state C_t, which facilitates the retention of long-term dependencies from the preceding sequence data. (Dixon, Polson, & Sokolov (2018) suggests the following architecture for the purpose of forecasting:

Output:
$$Z_t = O_t \cdot \tanh(C_t)$$
$$K_t = \tanh(W_c^T[Z_{t-1}, X_t] + b_c) \tag{6}$$
$$C_t = F_t \cdot C_{t-1} + I_t \cdot K_t$$

State equations:
$$\begin{pmatrix} I_t \\ F_t \\ O_t \end{pmatrix} = \sigma(W^T[Z_{t-1}, X_t] + b)$$

where $\cdot$ denotes the dot product. and I_t, F_t, O_t. are the input, forget, and output states $F_t \cdot C_{t-1}$ defines the long-range dependence with the state (I_t, F_t, O_t) Fig. 3 is the graphical represntation of architecture (6)

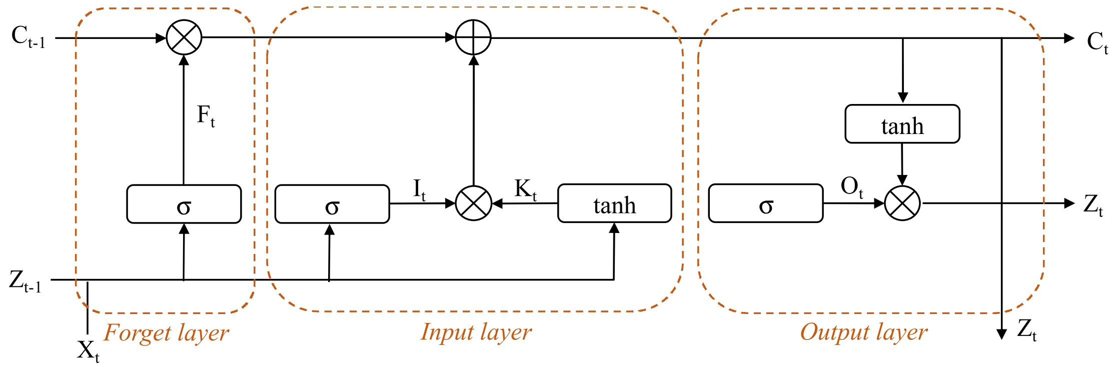


Fig. 3. Hidden layer of a LSTM. Input (Z_{t-1}, X_t) and state output (Z_t, C_t)

Source: Adapted from (Dixon, Polson, & Sokolov, 2018).

The cell state C_t is the characteristic that sets an LSTMN apart from an RNN. Gates, characterized by the activation function $\sigma(x)$ and dot product ·, are employed to modify the memory state by either adding or removing data. The initial gate $F_t \cdot C_{t-1}$, commonly referred to as the "forget gate," selectively ignores certain portions of the preceding cell state's information. The gate$I_t \cdot K_t$, referred to as the "input gate," is responsible for determining the values that need to be updated. Subsequently, the updated cell state is obtained by adding the previous cell state to the components of the vector $[Z_{t-1}, X_t]$ that were selected by the forget gate (Ruiz-Cárdenas et al., 2012). Therefore, the vector C_t serves as a means to eliminate irrelevant data from previous time steps and incorporate relevant data from the current time step. The output is determined by the output gate $Z_t = O_t \cdot \tanh(C_t)$, which is the result of multiplying the output gate O_t with the hyperbolic tangent function applied to the state of the cell C_t, with certain entries removed. Hence, the forget gate plays a crucial role in addressing the issue of vanishing and exploding gradients. There are two algorithms within LSTM, namely stateless and stateful. The primary distinction between these two algorithms revolves around the management of hidden states and cell states across sequences during training and prediction. In a stateful LSTM, the hidden state and cell state persist throughout the entire sequence, maintaining continuity across multiple data points. Conversely, in a stateless LSTM, both the hidden state and cell state are reset at the commencement of each sequence, leading to a fresh start for processing each new set of input data (Bernico, 2018).

3. Application

3.1 Apple Stock Price Data

The stock price data for Apple Inc. has been obtained from finance.yahoo.com, covering monthly records from 1981 to 2023. We utilized the adjusted closing prices for the analysis. The R-code was modified from business-science.io, (2018). Table 1 provides descriptive statistics for Apple Inc.'s stock price data, and Fig. 4 shows the monthly trend of Apple Inc.'s stock price from January 1981 to December 2023.

Table 1
Descriptive statistics of the Apple Inc’s stock price from January 1981 to December 2023

Statistics	Date	Stock Price (USD)
Min	Jul-82	0.044
Q1	Jan-96	0.242
Median	Apr-92	0.417
Q3	Mar-14	16.795
Max	Dec-23	194.309

Source: finance.yahoo.com

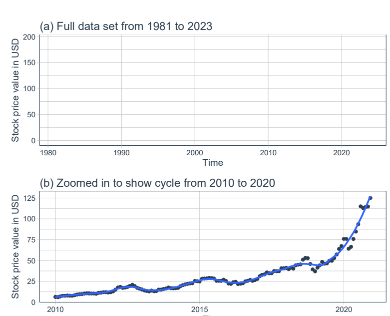


Fig. 4. (a) Monthly Temporal Trend of Apple Inc. Stock Price from 1981 to 2023, and (b) zoomed in to show cycle from 2010 to 2020

Fig. 4 (a) depicts a significant and consistent increase in the stock price of Apple Inc., with a particularly strong upward trend starting in 2020 and following an exponential growth pattern. To illustrate a more distinct cyclic pattern, data from 2010 to 2020 is presented in Fig. 4(b). The visualization depicts a recurring circular pattern over a specific time interval.

3.2 Spatiotemporal autocorrelation

Next, we will perform an autocorrelation test, which will serve as the basis for using time series models, such as LSTM. Time series models utilize autocorrelation to produce accurate predictions for multiple future time periods. Our goal is to produce short term forecasts for a 1-year or twelve-month timeframe using batch forecasting, a method that entails generating a consolidated set of forecasts for a specific period, rather than repeatedly making a single prediction for one or more future time periods. Batch predictions are only effective if the autocorrelation extends beyond a period of twelve months. Firstly, it is essential to review the Autocorrelation Function (ACF), which measures the correlation between a time series and its lagged versions.

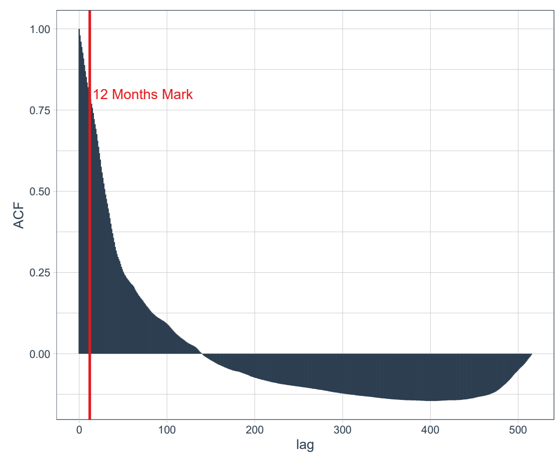


Fig. 5. Autocorrelation function

We computed 516 autocorrelation values (one for the time series and its 516 lags), and the results appear promising. A substantial number of lags show statistically significant autocorrelations. The news is indeed encouraging. Autocorrelation remains greater than 0.5 even after lag 12, which is equivalent to one year. Theoretically, one of the high-autocorrelation lags could be utilized to construct a Bayesian dynamic model in addition to an LSTM model.

However, there's a possibility that autocorrelations at lags 2 and beyond are primarily influenced by the propagation of autocorrelation at lag 1. This suspicion is corroborated by the Partial Autocorrelation Function (PACF) plot. The PACF plot clearly shows a prominent peak at lag 1, suggesting that the autocorrelations beyond this lag can be well explained by the autocorrelation at lag 1 (Fig. 5).

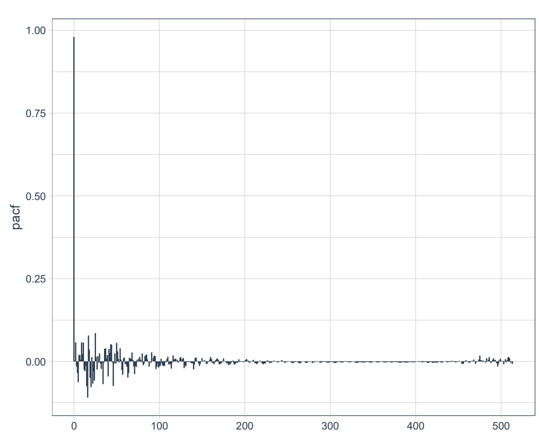


Fig. 6. Partial autocorrelation function

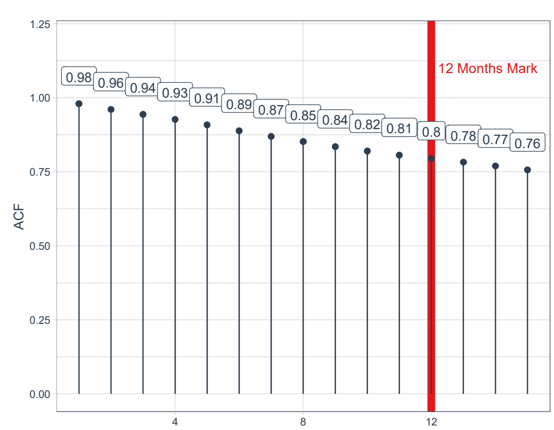


Fig. 7. ACF for 12 Months

After careful examination, the optimal lag appears to be at lag 1. However, this may not be the definitive choice, as there are additional considerations for batch forecasting with a LSTM and Bayesian dynamic model.

3.3 Backtesting: Time Series Cross Validation

Cross-validation involves constructing models on subsets of data and assessing them against a validation set to determine an anticipated accuracy level and error range. In the context of time series, the approach differs from non-sequential data, as it necessitates preserving the time dependency on previous samples when devising a sampling plan. To address this, a cross-validation sampling plan can be established by adjusting the window used to select sequential sub-samples. In finance, a similar analysis is often referred to as “Backtesting”, where a time series is divided into multiple uninterrupted sequences offset at various windows, allowing testing of strategies on both current and past observations (business-science.io, 2018). In order to facilitate cross-validation, the proposed sampling scheme employs a training set consisting of 12×1 samples, which corresponds to a year of study. The validation set, on the other hand, covers ten years and is assessed at 12×1. In order to ensure that the samples are distributed uniformly across four sets that encompass the complete 43-year history of sunspots, a twenty-year skip span (skip = 12×1) is selected.

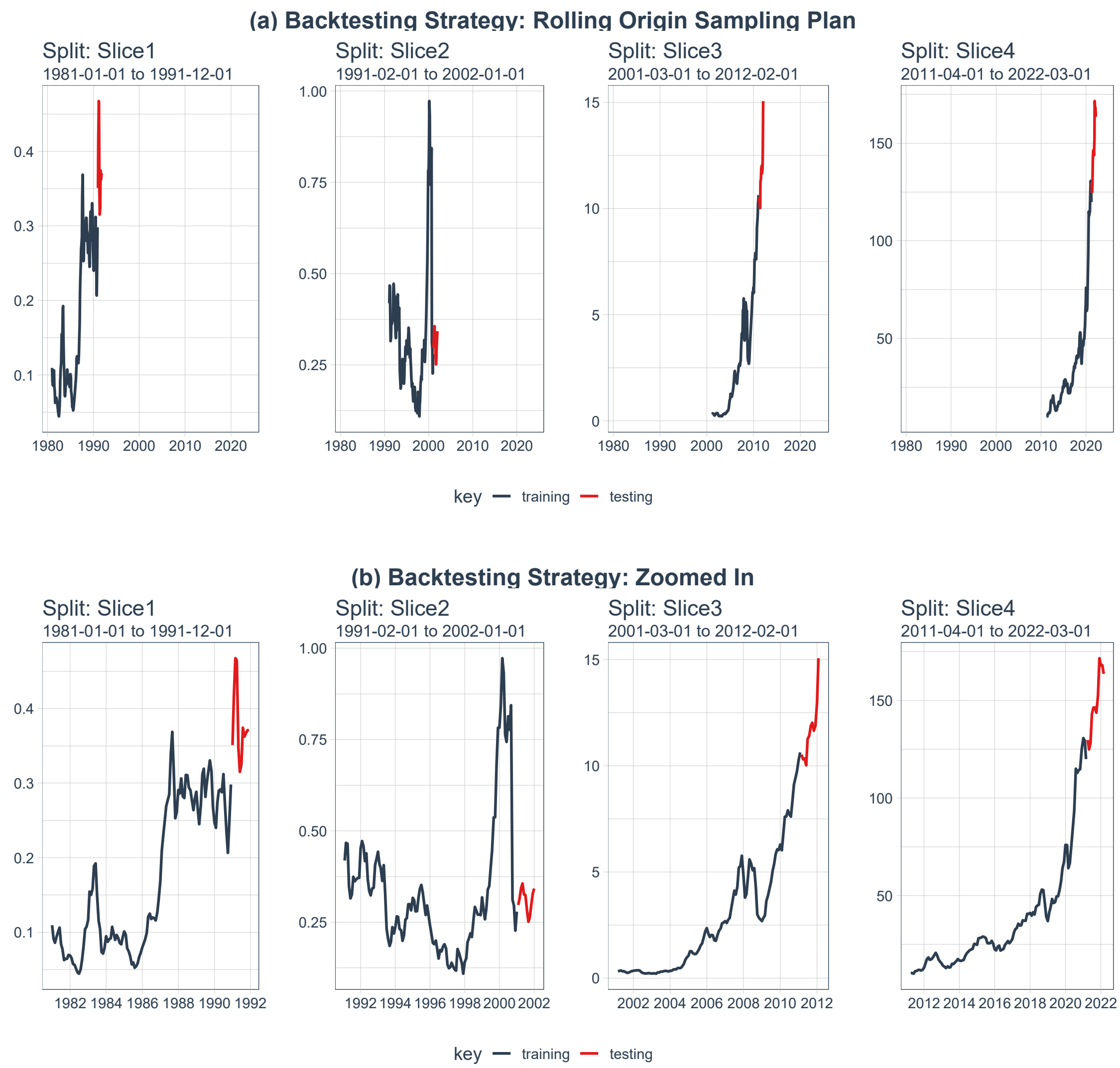
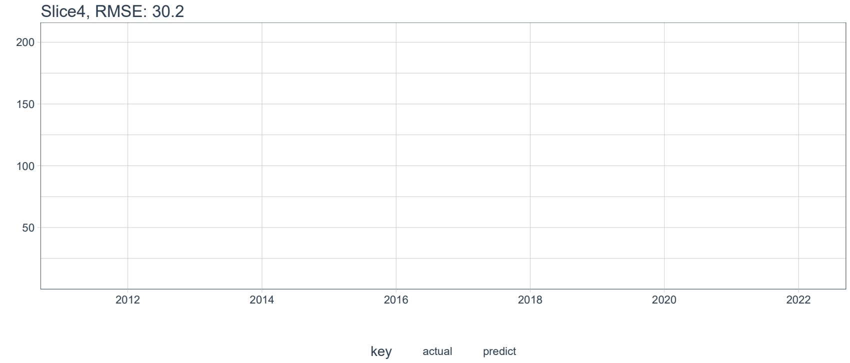


Fig. 8. Backtesting strategy

The four slices rolling the original sampling plan are depicted in Fig. 8. By employing multiple train-test splits, a greater number of models will be trained, leading to a more precise estimation of the models' performance on unobserved data.



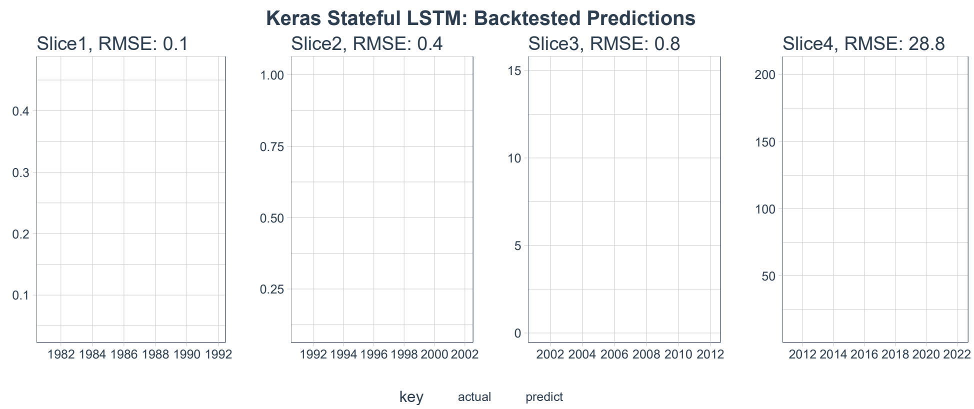


Fig. 9. Keras Stateful LSTM Backtested Prediction

3.4 Stateless and stateful architectures comparison

To assess the accuracy of both stateless and stateful LSTM models in predicting Apple's stock prices, we partitioned the dataset into training and testing sets. The testing data comprises only the twelve months of data from 2023, accounting for approximately 2.3% of the entire dataset. This selection aligns with the common practice of focusing stock price forecasting models on the short term. Fig. 10 provides a visual comparison of the forecasting results between the stateless and stateful LSTM methods.

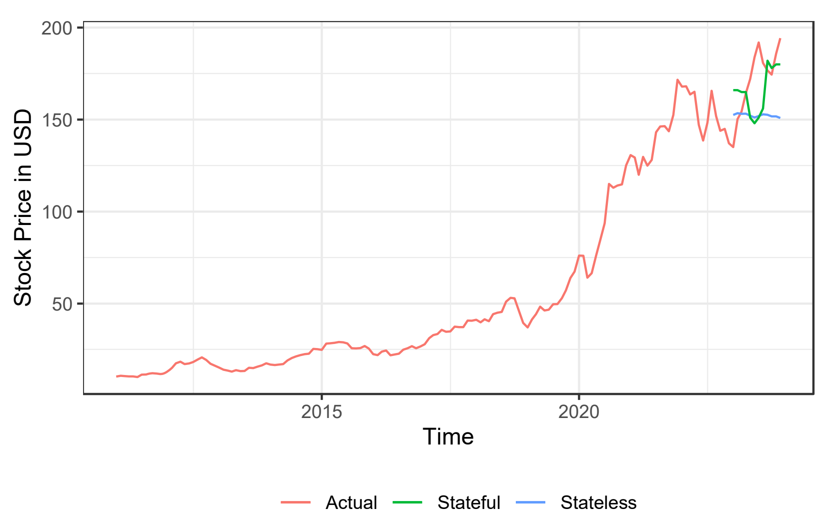


Fig. 10. Comparison between Stateless and Stateful LSTM Models

Fig. 10 clearly demonstrates that the stateful algorithm produces more precise predictions for the testing data in comparison to the stateless algorithm. The results obtained from the stateful algorithm's predictions demonstrate a noticeable trend pattern that closely resembles the actual data, indicating a positive trend. Conversely, the stateless predictions exhibit a more consistent pattern. In order to obtain a more accurate assessment of the precision of the two methods, we compare them using the mean absolute error (MAE) and mean absolute prediction error (MAPE) criteria, which are explained in detail in Table 2.

Table 2
Comparative Criteria for Stateless and Stateful LSTM Models

Month	Actual	Prediction		MAE		MAPE	
		Stateless	Stateful	Stateless	Stateful	Stateless	Stateful
Jan	135.02	152.52	166.00	17.49	30.98	12.95	8.84
Feb	150.27	153.42	166.00	3.15	15.73	2.09	8.20
Mar	154.34	153.12	165.00	1.22	10.66	0.79	7.76
Apr	164.38	153.16	165.00	11.22	0.62	6.83	7.73
May	172.07	152.07	151.00	20.00	21.07	11.62	0.70
Jun	183.79	151.10	148.00	32.70	35.79	17.79	2.05
Jul	191.90	151.89	151.00	40.01	40.90	20.85	0.59
Aug	180.76	152.82	156.00	27.94	24.76	15.46	2.08
Sep	176.77	152.57	182.00	24.20	5.23	13.69	19.29
Oct	174.44	151.70	178.00	22.74	3.56	13.04	17.34
Nov	185.80	151.69	180.00	34.11	5.80	18.36	18.66
Dec	194.31	150.87	180.00	43.44	14.31	22.36	19.31
Overall mean				23.18	17.45	12.99	9.38

Table 2 shows the real values, forecasts produced by stateless and stateful algorithms, along with the calculations of MAE and MAPE values. The stateful algorithm demonstrates significantly lower MAE and MAPE values in comparison to the stateless algorithm. This observation implies that, when using the same set of Apple stock price data, the stateful algorithm produces more precise outcomes. This difference can be partially attributed to the inherent characteristics of stock prices, which are characterized by strong temporal autocorrelation.

3.5 Forecasting Apple Inc's stock price

Based on the comparison between stateless and stateful LSTM algorithms, the application of the stateful LSTM algorithm resulted in a forecast for Apple's stock prices spanning January to December 2024, as depicted in Figure 8. The forecast was trained for more than 50 epochs using a stateful architecture with the following parameters: lag = 12, batch size = 12, train length = 12, time steps = 1, and train length = 12. The forecasted results indicated a positive trend in Apple's stock prices throughout 2024. Nevertheless, it is crucial to emphasize, particularly for investors, that these forecast outcomes are solely derived from historical data. Real estate stock prices are intricately influenced by various factors, underscoring the need for caution in interpreting the insights gleaned from this forecast.

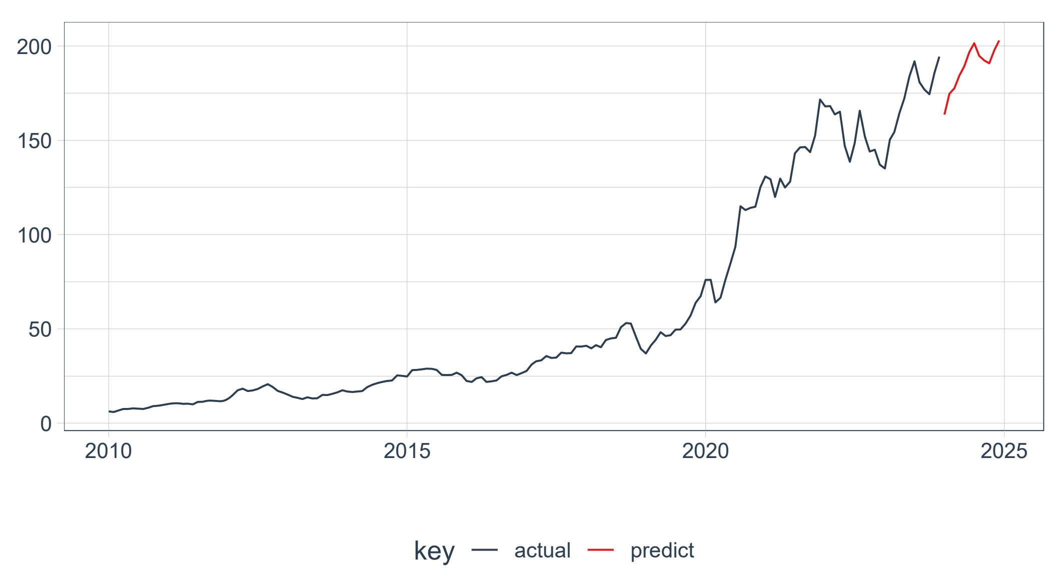


Fig. 11. Forecast of Apple Stock Prices for January to December 2024

4. Conclusion

The objective of this study is to perform a comparative analysis of stateless and stateful algorithms in LSTM models. This specific topic is often neglected when it comes to forecasting, especially in the domain of predicting stock prices. Although LSTM models are commonly used for forecasting, there is limited discussion on the subtle distinctions between stateless and stateful algorithms. This comparative analysis is essential for determining the most appropriate LSTM algorithm, considering the distinct attributes of time series data, such as the advantages and disadvantages of temporal autocorrelation. To accomplish this, we gathered empirical data from https://finance.yahoo.com/ that includes the closing stock prices of Apple Inc. for a monthly period of 43 years, starting from January 1981 and ending in December 2023. Exploratory data analysis uncovered a strong temporal autocorrelation among observation periods, indicating the potential effectiveness of the stateful algorithm approach. This hypothesis was supported by empirical comparisons. The evaluation entailed utilizing data from January to December 2023 as the testing dataset, while the remaining dataset was employed as the training dataset. The choice to exclusively employ a 12-month testing period is in accordance with the prevailing convention of predicting short-term fluctuations in stock prices. The analytical findings revealed that, although the stateful algorithm approach required more time for computation, it surpassed the stateless approach in terms of accuracy. Expanding upon the algorithm that maintains a memory of previous states, predictions for the value of Apple's stocks were generated for the time frame spanning from January to December 2024. Although the predictions suggest a possible rise in Apple's stock price in 2024, it is important to acknowledge that these projections rely on past data and that unforeseen changes can greatly affect the actual value of the shares. Therefore, investors should remain vigilant and monitor the situation closely.

Acknowledgement

Thanks to the Rector and the Directorate of Re-search, Community Service, and Innovation of Universitas Padjadjaran for providing the research grant program.

References

Bernico, M. (2018). Deep Learning Quick Reference . Mumbai: Packt Publishing.

Bismi, I. (2023, May 29). Difference between Stateful and Stateless RNNs. [Accessed 06 January 2024]. From Medium: https://medium.com/@iqra.bismi/difference-between-stateful-and-stateless-rnns-2b397184e759

Brownlee, J. (2019, August 28). How To Backtest Machine Learning Models for Time Series Forecasting. Retrieved January 2024, from Machine Learning Mastery: https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting/

business-science.io. (2018, April 17). Time Series Deep Learning: Forecasting Sunspots With Keras Stateful LSTM In R. [Accessed 06 January 2024]. From R-Bloggers: https://www.r-bloggers.com/2018/04/time-series-deep-learning-forecasting-sunspots-with-keras-stateful-lstm-in-r/

Dixon, M. F., Polson, N. G., & Sokolov, V. O. (2019). Deep learning for spatio-temporal modeling: dynamic traffic flows and high frequency trading. *Applied Stochastic Models in Business and Industry*, *35*(3), 788-807.

Gülmez, B. (2023). Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm. *Expert Systems with Applications, 227*, 120346.

Ho, M., Darman, H., & Musa, S. (2021). Stock Price Prediction Using ARIMA, Neural Network and LSTM Models. *Journal of Physics: Conference Series, 1988,* 012041.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Lee, S., & Kim, H. (2020). Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation Author links open overlay panel. *Expert Systems with Applications, 161*(15), 113704.

Levin, E. (1990). A Recurrent Neural Network: Limitations and Training. *Neural Networks, 3*, 641-650.

Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. *Procedia Computer Science, 170*, 1168-1173.

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M., Barrow, D., Taieb, S., . . . Bessa, R. (2022). Forecasting: theory and practice. *International Journal of Forecasting, 38*, 705-871.

Polyzos, S., Samitas, A., & Spyridou, A. E. (2021). Tourism demand and the COVID-19 pandemic: An LSTM approach. *Tourism Recreation Research*, *46*(2), 175-187.

Ruiz-Cárdenas, R., Krainski, E., & Rue, H. (2012). Direct fitting of dynamic models using integrated nested Laplace approximations — INLA. *Computational Statistics and Data Analysis, 56*, 1808–1828.

West, M., & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. New York: Springer.

Wu, J., Xu, K., Chen, X., Li , S., & Zhao, J. (2022). Price graphs: Utilizing the structural information of financial time series for stock prediction. *Information Sciences, 588,* 405-424.