

From Price to Returns: The Performance Evolution of Machine Learning Models in Stock Prediction Tasks

Zirui Zeng *

School of Management, Shenzhen University, Shenzhen, China

* Corresponding Author Email: 2022041013@email.szu.edu.cn

Abstract. More and more machine learning models have been used in financial forecasting. However, its application in real life has encountered many difficulties, mainly due to the non-stationary nature of the stock price data. This paper systematically examines the important effect that problem formulation has on predictive quality by looking at two different modeling approaches. The first paradigm makes direct end-to-end prediction on raw non-stationary price sequences. On the other hand, the second is theoretically informed, and focused on forecasting stationary log returns of carefully engineered features. A suite of models (LSTM, Random Forest, Ridge) were compared to a Naive Forecast in both frameworks. Empirical tests resulted in very poor performance from models of the first paradigm: the model performed abysmally and the predictions were a lot worse than the baselines. On the other hand, they applied to stationary returns are extremely predictive. The paper's main idea is that making data stationary is more than something people do to improve the model, it is essential to obtain any form of significant financial forecast. The selection of a well founded methodology over the selection of a particular complex algorithm is much more significant.

Keywords: Stock Price Prediction, Machine Learning, Time Series Forecasting, Data Stationarity.

1. Introduction

To predict the stock prices in the financial market has been an eternal problem, no matter for investment professionals or academic people. People apply machine learning model to do prediction, it not only could make us lots of economic profits, and, researchers can see the complicated financial system by using those machine learning models as benchmarks. Compute is more now due to more data and because of the more data and more compute, machine learning/deep learning are a must. LSTM such as is a kind of more complicated model, it has shown much potential, since nonlinear relationship can be handled by it and long-term dependency information can be obtained as well [1].

However, in the process of researching and applying, there is one problem that has always been an issue. And some of these big and complex models actually do much worse at forecasting real stock prices. Sometimes even worse than the simplest benchmark model. Because of the nature of Stock Price Data, that's called "non-stationary" data set [2]. That is, its statistical properties such as mean and variance change over time. This is a violation of one of the core assumptions underlying many machine learning models, that the data distribution remains stable. Therefore, it is difficult for models to learn patterns that can predict the future stably, and they are easy to be affected by short-term trends or noise, resulting in severe overfitting problems [3].

In the persistent difficulty in this field suggests the main problem is not about how complex the model is, but the problem is that people are trying to predict directly the stock prices. Therefore, this research is no longer about having a more complicated model architecture. Rather, by way of a number of controlled studies, this paper looks into the inherent flaws in the traditional way. The research then introduces and validates a more effective alternative methodology that overcomes the limitations of the widely used but ineffective one. The main contribution of this work is to supply an empirical argument for this important change in the way people think, so that the community can find a better research direction.

The main contributions of this paper can be summarized as:

In the first place, the forecasting paradigms are compared in detail. It creates a managed setup to assess not only a few models but also the wider methodologies those models are used for. The study makes explicit comparison between the end to end price prediction and one where forecast is made for stabilized log returns.

Second, it specifies conditions for the successful application of complex sequential models. The study explores the frequent assertion of unstable performance for advanced models like LSTMs in financial settings. It explores the hypothesis that these problems are due not to the model but the use of them with non-stationary price data and thereby defines what problem formulation is needed for them to succeed.

Lastly, it gives a thorough, quantifiable argument for the fundamental notion that defining the problem comes first before choosing a model. Forecasting returns is a known best practice, but it is often treated like a heuristic. This study attempts to quantify the performance cost of making data stationary; finding that the paradigm choice is far more important to success than the specific model architecture choice that follows.

2. Related Work

The basic difficulty of stock price prediction is the non-stationary of financial time series [4, 5], which violates the assumption of most machine learning models. This is not just an empirical observation; theory shows that the generalization error of a standard model on non-stationary data can be undefined, which explains why it often fails [6]. As a result, the field has split into two main streams: direct price forecasting and forecasting of stationary returns.

Direct Price Forecasting Paradigm is full of problems. The paper has proven over and over that a really complicated model of just plain price will not do well, and it will over-fit. Take random forests for instance, which have been seen to produce much larger errors than simple linear models [7], or the sequence-aware architectures like LSTM that can easily blow up [8, 9]. Some complex hybrids have succeeded, but they usually require large, non-generalizable, hand-crafted features rather than learning key price dynamics directly, which suggests that the direct methods are hard [10].

To address these issues, a standard strategy is to change the forecasting goal into a more stable metric, generally logarithmic returns. This shift is also theoretically justified as it is much easier for the machine learning models to learn return series rather than price series [9], and is also empirically true. For example, researchers have shown that changing the target from price to log-returns can vastly increase the accuracy of a model. In fact, doing so increased the R^2 from 0.77 to 0.92 [4]. For those context, in this case, the more advance models like LSTMs are better at finding complex patterns than the older ones [5, 11].

But the paradigm does have its problems. A large replication study by Li and Wang found that many published high performance results had issues with their methods, and simple robust models such as ridge often beat complex LSTM even at return prediction [12]. This collection of conflicting pieces of evidence reveals a gaping hole, one in which there is no systematic, side by side comparison of many different kinds of models, ranging from very simple to very complex, inside a single, controlled framework and between the two different paradigms.

This study aims to fill this gap. To empirically establish the decisive effect of our chosen paradigm on the performance of ML models by performing rigorous experiments on a set of representative models, to resolve the contradictions in the existing literature, and to give practitioners an unambiguous and dependable guide.

3. Methodology and Experimental Design

This study carefully checks a big experiment plan for two ways of predicting things. First, this chapter presents the datasets and some basic preprocessing shared by both methods. Next is an examination of the heart of the experiment, where the author presents two contending approaches to

feature engineering, one based on end-to-end learning and the other on explicit, domain-driven feature creation. Then the paper introduces a set of selected prediction models, which cover various models including classic benchmarks and advanced neural network models. Finally, this chapter introduces the particular experiments and metrics used to compare models under the two paradigms.

3.1. Data and Feature Engineering Paradigms

Empirical research is based on daily stock data of Tesla, Inc. (NASDAQ: TSLA) from June 29, 2010 to December 31, 2023. Data from Kaggle. The high volatility and clear trends seen in TSLA's price history serve as a tough, representative test case for modern forecasting models. To face head-on the main debate within financial time-series analysis, the paper takes the two competing approaches and turn them into different experimental paradigms.

Very strict preprocessing schedule is used to protect analytical soundness and keep clear of look-ahead bias. The dataset is first split chronologically into a training set (first 80% of timeline) and a test set (last 20% of timeline). Then all input features are standardized with a StandardScaler that is fit only on the training data. The statistical parameters learned from the training set are used to transform both the training and test set, which is an important step to avoid information about the future from being used to train the model.

3.1.1. End-to-End Learning from Raw Price Sequences

The first paradigm is on whether models can learn from raw non-stationary prices end-to-end. In this approach, the model is queried for the absolute value of the closing price of the next time step denoted by. To do this, they are given the unprocessed OHLCV data of the previous 30 days as inputs. This method purposefully avoids manual feature engineering, to test if the models have any raw pattern recognition skill at all on a hard, trendy time series.

3.1.2. Explicit Feature Engineering from Stationary Returns

In contrast, the second paradigm adopts a theoretically-grounded methodology that begins by transforming the data to achieve stationarity, a critical prerequisite for many time series models. Instead of predicting the price itself, the models here are trained to forecast the logarithmic return, calculated as $\text{Return}_t = \log(\text{Close}_t / \text{Close}_{t-1})$. Correspondingly, all input features are carefully engineered from historical data available up to time step $t-1$ and are expressed in the return or change domain. These engineered features include metrics such as lagged intraday volatility, lagged intraday trend, and the sequence of the past 30 logarithmic returns, providing the models with a more stable statistical foundation for learning.

3.2. Model Selection

The current work makes use of the type of models which are targeted at some specific kind of models rather than all the possible types of models and these models represent different tiers of logical complexity.

The basic standard is the Naive forecast. This persistence model is the basic reference. It is a "no-change" forecast, where the forecast for the next time step is equal to the value at the current time step. Model is:

$$\hat{Y}_{t+1} = Y_t \quad (1)$$

where $\hat{Y}_{t+1}$ is the observed value at the current time step. It provides a baseline that the paper can use to measure how much better more advanced models are.

To cover the class of simple but robust linear models, this study includes ridge regression. This model adds an L2 regularization penalty to the regular linear regression loss function, which can help prevent overfitting by penalizing large coefficients. Its objective function is to find the coefficients ($\mathbf{w}$) such that:

$$\min_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \alpha \|\mathbf{w}\|_2^2 \quad (2)$$

Where $\mathbf{X}$ is the matrix of input features, $\mathbf{y}$ is the target variable, $\mathbf{w}$ is the vector of model coefficients, and the hyperparameter α Controls the strength of regularization. It is to test the hypothesis that a well-regularized, simple model would do surprisingly well, especially on stationary data.

A random forest is also included as a representative of the traditional, non-sequential machine learning ensembles. A Random Forest works by creating lots and lots of decision trees during the training process. For a regression task, the final prediction will be the average of the predictions of all the individual trees in the forest. If the forest contains B trees $\{\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_B\}$, the prediction ($\hat{\mathbf{y}}$) for a new input instance $\mathbf{x}$ is given by:

$$\hat{\mathbf{y}} = \frac{1}{B} \sum_{b=1}^B \mathbf{T}_b(\mathbf{x}) \quad (3)$$

where $\mathbf{T}_b(\mathbf{x})$ is the prediction of the b -th tree. To evaluate the performance of powerful but inherently not designed for sequence data non-stationary methods on both forecasting paradigms.

Lastly, the paper used LSTM, which is a special kind of RNN, to model the long range sequential dependency. The LSTM's key part is its cell state, this is like a conveyor belt for memory that lets information stay around for a long time. There are 3 gates for this memory:

- Forget Gate: It decides what information to forget about the cell state.
- Input Gate: Decides what new info to store.
- Output Gate: filters the cell state to output the prediction for the current time step.

This gated mechanism allows LSTM to learn which part of the history information to keep and which one to forget. It is very suitable for capturing complex, long-term patterns in financial time series.

3.3. Experimental Procedure and Evaluation

A standardized experimental procedure is used to make the comparison more rigorous and fair. All of them are trained just on the preprocessed training set. For the LSTM model, Adam optimizer and Mean Squared Error loss function is used and also used early stopping callback to avoid overfitting.

After the training is done, each of the models will generate out-of-sample predictions on the test set. For Paradigm 2, a critical reconstruction step is made. As such, their model outputs are on the log return scale ($\hat{\mathbf{r}}_t$), and are converted to the price domain with the previous day's actual closing price: $\hat{\mathbf{P}}_t = \mathbf{P}_{t-1} \cdot \exp(\hat{\mathbf{r}}_t)$. This means the paper knows that every single model on each side has generated the last estimation on the same thing (price) so the paper can see which one did a better job at predicting it directly and justly.

The two standard regression metrics are applied to evaluate the final price predictions.

Root Mean Squared Error (RMSE): this measure calculates the average amount of prediction error in the original price units.

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\mathbf{P}_i - \hat{\mathbf{P}}_i)^2} \quad (4)$$

where $\mathbf{P}_i$ is the actual price, $\hat{\mathbf{P}}_i$ is the predicted price, and n is the number of samples in the test set.

R-Squared (R^2): This coefficient of determination represents the percentage of variation in actual prices that is explained by the model's forecasts.

$$R^2 = 1 - \frac{\sum_{i=1}^n (\mathbf{P}_i - \hat{\mathbf{P}}_i)^2}{\sum_{i=1}^n (\mathbf{P}_i - \bar{\mathbf{P}})^2} \quad (5)$$

where $\bar{\mathbf{P}}$ is the mean of the actual prices in the test set.

4. Results

This study evaluated four different forecasting models under two different paradigms: forecasting prices directly, or forecasting log returns. The results indicate that the choice of forecasting paradigm is the one factor that has the greatest impact on a model's performance.

4.1. Paradigm 1: Ineffectiveness of Direct Price Prediction

The first paradigm applies models directly to the non-stationary price series. The results, shown in Fig. 1, show clearly the failure of the advanced models. The Naive Forecast set a good base line (RMSE: 42.49) but both the LSTM and random forest models had terrible performance, RMSE: 187.83 and 212.53 respectively. Interestingly, the Random Forest yielded a negative R-squared value of -2.5950, suggesting that it performed much worse than a naive mean forecast. This result indicates that even these complex architectures find it hard to model raw, trending financial data, and are frequently bested by the most basic persistence benchmark. Figure 1: Model Performance on Direct Price Prediction (Paradigm 1).



Fig. 1 Model Performance on Direct Price Prediction (Paradigm 1).

4.2. Paradigm 2: Success of Prediction via Logarithmic Returns

Second paradigm changed the forecasting target from price level to logarithmic return, which is more stable for forecasting. As shown in Fig 2, this results in an improvement in the efficacy of the model that is both dramatic and universal.

The performance of the LSTM and Random Forest models showed a reversal. Its RMSE fell like a stone to 42.98 and 45.08 correspondingly, in line with Naive Forecast. Their R-squared values skyrocketed into a high performance area (LSTM: 0.8530, Random Forest: 0.8382) indicating that the models were now able to explain a significant amount of the variance in the data. The predictions from all the models now visually matched the actual price movements closely, the statistical improvements were confirmed.

Summarizing, the results unambiguously show that transforming the non-stationary price series into a more stationary return series is a requirement for successfully applying the advanced machine learning models in question.



Fig. 2 Model Performance on Price Prediction via Logarithmic Returns (Paradigm 2).

5. Discussion

The chapter intends to explore the experimental results contained in Chapter 4, looking at these in terms of the macroeconomic environment of financial prediction, and discussing any limitations associated with this study.

The main finding of this study— the large difference in the predictive performance of the two experiments — supports the key role that data stationarity plays in time series modeling. Stock prices are classic non-stationary series, and their characteristics are often described as a "random walk," meaning that the statistical properties like mean and variance change constantly. This feature violates the basic assumption of most machine learning models, making it hard for machines to learn the general pattern. Our results in Paradigm 1 make clear how researchers see this dilemma playing out. Both simple models and complex models, most notably the random forest, fail to capture the simplicity and persistence of price data, generating large, noisy predictions.

On the contrary, logarithmic returns are made to measure the change, and their time series are much closer to stationary. The paper changes the job researches asks the model to do, moving it away from predicting prices themselves (paradigm 1) and towards using the statistically more stable returns (paradigm 2). So that the machine learning algorithms could really do their thing – finding complicated nonlinear stuff in a constant stream of numbers. Thus, the superior performance of LSTM networks and random forests in this context is not accidental, but an expected result of applying powerful tools on properly pre-processed data.

These results are practically very useful for the finance workers. They warned about taking ML models, “putting them in the box” (meaning a black box) and then pushing them through raw financial data. The key step of feature engineering – turning price into yield – isn’t some bonus that the paper gets to add in, it’s a necessity if the paper want to make any kind of prediction that means anything. Moreover, the naive forecasting shows its own unique value in both of the paradigms; it serves not just as a benchmark for the model performance, but also as a diagnostic tool. and when a model is complex and a simple model (paradigm 1) is performing better, which means that the modeling is failing.

But there are also limitations to this study. Firstly, all the experiments use one financial instrument during a certain time period. Whether these research conclusions can be extended to other assets, other markets, and different macroeconomic environments remains to be seen. Second, although the

paper uses the standard architecture for all models, detailed hyperparameter tuning is outside the scope of this study. But our results indicate that the model performance is much more affected by selecting the proper modeling paradigm compared to adjusting parameters.

6. Conclusion

The purpose of this research was to find different models that are used for the stock price prediction. But the main conclusion is not that any one model was superior – it's that the method was so vastly more important. The paper has shown that if researchers want to use ML for correct prediction of stock prices the paper needs to transform non stationary price to another kind of non stationary such as stationary log return. This is completed, more advanced models such as LSTM and Random Forest can generate quite accurate predictions. But they fail terribly when they do not. So to sum up, for the researchers and practitioners in the area of financial forecast, their main task is to make sure the data processing and problems setting is strict enough for them to apply some complicated prediction algorithms.

Based on the results of these studies, it is possible to move on to the next stage of the research in several ways. Firstly, it will be good to validate this conclusion on a broad array of asset portfolios and market regimes such as bull/bear/sideways markets. Second, after having established a successful paradigm, future work could include other predictive features such as trading volume, market sentiment from news headlines, or macroeconomic indicators to beat the naive benchmark. Finally, the paper can explore more modern architectures like transformer-based models, keeping the proven returns-based paradigm, and achieve better forecast accuracy.

References

- [1] Fischer T, & Krauss C. Deep learning with long short-term memory networks for financial market predictions. *European journal of operational research*, 270(2), 654-669. 2018.
- [2] Gu S, Kelly B, & Xiu, D. Empirical asset pricing via machine learning. *The Review of Financial Studies*, 33(5), 2223-2273. 2020.
- [3] Sezer O B, Gudelek M U, & Ozbayoglu A M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90, 106181. 2020.
- [4] Itri B, Mohamed Y, Omar B, Latifa E M, Lahcen M, & Adil O. Hybrid machine learning for stock price prediction in the Moroccan banking sector. *International Journal of Electrical & Computer Engineering* (2088-8708), 14(3). 2024.
- [5] Cao H, Lin T, Li Y, & Zhang H. Stock price pattern prediction based on complex network and machine learning. *Complexity*, 2019(1), 4132485.
- [6] Behera R K, Das S, Rath S K, Misra S & Damasevicius R. Comparative Study of Real Time Machine Learning Models for Stock Prediction through Streaming Data. *J. Univers. Comput. Sci.*, 26(9), 1128-1147. 2020.
- [7] Vuong P H, Dat T T, Mai T K, & Uyen P H. Stock-price forecasting based on XGBoost and LSTM. *Computer Systems Science & Engineering*, 40(1). 2022.
- [8] Kamalov F, Gurrib I, & Rajab K. Financial Forecasting with Machine Learning: Price Vs Return. *Journal of Computer Science*, 17(3), 251-264. 2021.
- [9] Guennioui O, Chiadmi D, & Amghar M. Machine learning-driven stock price prediction for enhanced investment strategy. *International Journal of Electrical & Computer Engineering* (2088-8708), 14(5). 2024.
- [10] Zhu Z, Chen H, Qu Q, & Chung V. FinCast: A Foundation Model for Financial Time-Series Forecasting. *arXiv preprint arXiv:2508.19609*. 2025.
- [11] Chaudhary T, & Minirani S. Machine Learning and Time Series Approaches for Stock Price Prediction: A Case Study of Indian Banks. *Procedia Computer Science*, 260, 626-634. 2025.
- [12] Li C, Gan Z, Yang Z, Yang J, Li L, Wang L, & Gao J. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2), 1-214. 2024.