Guidelines for training a backtesting machine learning model for financial time series data

Yuan-Long Peng

Abstract

Financial time series data belongs to sequential numerical data. There are several guidelines that practitioners should obey. N-fold cross-validation is invalid for financial time series data because it disrupts the order of data. Compared to categorical data, numeric data metrics are not restricted to a certain domain. For example, accuracy is restricted between 0 and 1; however, MSE is a positive value without restriction. Therefore, it is easy to overfit or underfit financial time series data. This article discussed preprocessing financial time series data and measuring the performance of time series data. This article introduces naïve prediction of previous values as baseline values for machine learning models on MSE, MAE, and MAPE metrics. These measures make the result more solid, sound, and close to real trading/investment.

1. Introduction

Backtesting is a method to verify and validate trading strategies. Because financial time series data cannot be disrupted, n-fold cross-validation cannot be applied to financial time series data. The training, validation, and testing data split needs to be sequential. The training data is used to train the machine learning model. The validation data is used to fine-tune hyper-parameters. The testing data is used to measure the performance.

There are two ways to prepare a split of backtesting. The first way, shown in Table 1, is to split all the time series data into training, validation, and testing data. The second way, shown in Table 2, divides all the time series data into training, validation, and testing data with a moving window. The first way has vast training data, which can include more patterns in the training data, but it cannot test different financial events in the early timeframe. The second way can learn patterns within the moving window, which allow testing results, including different period across all the

time series data and financial events (e.g., financial crisis). For example, the T5 and T6 can be the testing data if the moving window is adopted.

Table 1 Conducting time series data backtesting with the training, validation, and testing dataset split.

All the time series data											
T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**
Crisis 1		**Crisis 2**		**Crisis 3**	**Crisis 4**			**Crisis 5**		**Crisis 6**	
Training data				Validation data				Testing data			

Table 2 Conducting time series data backtesting with moving window.

All the time series data											
T1	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**	**T12**
Crisis 1		**Crisis 2**		**Crisis 3**	**Crisis 4**			**Crisis 5**		**Crisis 6**	
Training data		Validation data		Testing data							
		Training data		Validation data		Testing data					
				Training data		Validation data		Testing data			
						Training data		Validation data		Testing data	

The financial time series data is sequential numerical data. Therefore, improper training of machine learning models will lead to a biased result. The following guidelines are golden rules for training machine learning models on financial time series data.

Guideline 1: From Stationary to Overfitting and Underfitting

Classical time series prediction models, such as AR(autoregression), MA(moving averaging), and ARIMA (Autoregressive Integrated Moving Average model), require the time series data to conform to stationery. Stationary means the distribution of time series data is identical across time frames. Unit root test can be applied to test whether it is stationary. However, a trend in time series data is viewed as non-stationary.

Machine learning can learn patterns in time series data regardless of stationary or non-stationary. Therefore, machine learning models are considered state-of-the-art methods. With the superior capability for pattern learning, the machine learning

model tends to learn patterns along with noises. On the contrary, machine learning models can underfit the time series data with improper hyper-parameters. The method to measure over-fitting and under-fitting is to split the time series data into training, validation, and testing dataset and measure the performances of the training and validation dataset to make sure they are close. It is overfitting if the training data result outperforms the validation data result. Table 3 shows an example of overfitting. To avoid overfitting, machine learning models require fine-tuning hyper-parameter to ensure similar performances on training and validation data [5]. If the result of training data is inferior to the naïve baselines on training data, it is underfitting. Table 4 shows an example of underfitting. To avoid underfitting, machine learning models require fine-tuning hyper-parameter to ensure results outperform the naïve baselines on training data.

Table 3. Example of overfitting: MSE, MAE, and MAPE values on validation data are greater than the training data.

	Training data	**Validation data**
MSE	0.38	0.72
MAE	0.64	0.91
MAPE	0.54	0.82

Table 4. Example of underfitting: MSE, MAE, and MAPE values exceed the naïve baselines on training data.

	Training data	**Naïve baseline on training data**
MSE	0.38	0.26
MAE	0.64	0.56
MAPE	0.54	0.42

Guideline 2: Data balance of regression and classification models

The objective of machine learning is to minimize the loss. Therefore, when the data is imbalanced, the machine learning model tends to predict the majority class or value in the dataset. The method to fix the problem of categorical data imbalance is oversampling, downsampling, and SMOTE(Synthetic Minority Over-sampling Technique) [1]. Table 5 shows oversampling of class 2 to balance the training data, and Table 6 shows the downsampling of class 1 to balance the training data.

Table 5. Example of ***oversampling for categorical data balance***.

	Class 1 (e.g., Rise)	**Class 2 (e.g., Fall)**
The original number of Data	5,000	2,500
Oversample class 2 for two times	5,000	**5,000**

Table 6. Example of ***downsampling for categorical data balance***.

	Class 1 (e.g., Rise)	**Class 2 (e.g., Fall)**
The original number of Data	5,000	2,500
Downsample class 1 to half	**2,500**	2,500

Over-sampling or under-sampling cannot be applied to the numerical data because the numerical data is continuous. It means the range of numerical datasets can be divided into infinite portions. Therefore, over-sampling and under-sampling cannot balance infinite portions. Instead of fixing the data imbalance in the numerical dataset, we can monitor the similarity of training data distribution and validation (please note that the testing data cannot be used in the case of data balance; otherwise, it is data leakage.). Table 7 shows that the distribution of training and validation data is similar; however, Table 8 shows that the distribution of training and validation data is dissimilar.

Table 7. Example of monitoring numerical data balance - training data and validation ***share the same distribution***.

	1st quartile	**Median**	**3rd quartile**
Training data	15.2	19.3	24.1
Validation data	15.0	19.4	24.2

Table 8. Example of monitoring numerical data balance - training data and validation ***do not share the same distribution***.

	1st quartile	**Median**	**3rd quartile**
Training data	15.2	19.3	24.1
Validation data	11.0	12.4	17.2

Metrics of machine learning models

The classification model has accuracy, precision, and recall. The range of accuracy is between 0 and 1. Regarding data imbalance, accuracy is not an objective

metric because the model tends to predict the majority class. For example, the dataset contains two classes with a proportion of 1 to 99. The model will naïve predict the majority class to minimize loss.

Metrics of precision and recall can avoid the shortcoming of accuracy for a classification model. Precision refers to the proportion of true positive accounts for the summation of true positive and false positive. Recall refers to the proportion of true positive accounts for the summation of true positive and false negative. Therefore, precision and recall can avoid data imbalance because they emphasize the proportion of true positives over false positives and false negatives.

Guideline 3: Naïve prediction of the previous value is the baseline of financial numerical data metrics

There are three popular metrics to measure the performance of numerical data: MSE (Mean Squared Error), MAE (Mean Absolute Error), and MAPE (Mean Absolute Percentage Error). The mathematical expressions are shown as follows [3].

$$\text{MSE} = \sum_{t=1}^{n} \frac{(x_t - \bar{x}_t)^2}{n} \tag{1}$$

$$\text{MAE} = \sum_{t=1}^{n} \frac{|x_t - \bar{x}_t|}{n} \tag{2}$$

$$\text{MAPE} = \sum_{t=1}^{n} \frac{1}{n} \frac{|x_t - \bar{x}_t|}{\bar{x}_t} \tag{3}$$

For most practitioners, showing the performance of the regression model is to compare it with baseline models (e.g., neural networks). However, when the baseline is under-fitting, the improvement is unreliable. To find proper baselines, let's naïve predict the value of the last value x_{t-1}. Rewrite the MSE, MAE, and MAPE as naïve prediction baselines and the equations are shown as follows.

$$\text{Naive prediction MSE} = \sum_{t=1}^{n} \frac{(\bar{x}_{t-1} - \bar{x}_t)^2}{n} \tag{4}$$

$$\text{Naive prediction MAE} = \sum_{t=1}^{n} \frac{|\bar{x}_{t-1} - \bar{x}_t|}{n} \tag{5}$$

$$\text{Naive prediction MAPE} = \sum_{t=1}^{n} \frac{1}{n} \frac{|\bar{x}_{t-1} - \bar{x}_t|}{\bar{x}_t} \quad (6)$$

Naïve prediction baselines mean the model simply predicts the t-1 value. If the machine learning model underperforms naïve prediction baselines on the training dataset, the model is underfitting. Table 9 shows an example of underfitting. On the contrary, the model is overfitting if the MAPE of the machine learning model outperforms naïve prediction baselines on the training dataset but is inferior to naïve prediction baselines on the validation dataset.

Table 9. Example of comparing the performance of machine learning models with naïve prediction baselines. Naïve prediction outperforms machine learning models.

	MSE	**MAE**	**MAPE**
Neural Network	0.14	0.32	0.42
XGBoost	0.13	0.41	0.49
Random Forest	0.19	0.31	0.45
Naïve prediction	***0.12***	***0.30***	***0.41***

Guideline 4 Backtesting return should be greater than the opportunity cost

Opportunity cost is the baseline of your prediction of return. Fixed deposit interest rate and 10-year U.S. treasury yield can be viewed as the opportunity cost of capital because they are risk-free interest rates. The result is a failure when the backtesting return is greater than 0 but less than the opportunity cost.

Example of opportunity cost: The annualized return of backtesting is 2%, and the fixed deposit interest rate is 3%. It implies that depositing money in the bank can outperform your financial time series data prediction.

Guideline 5 Dealer market guarantees liquidity; however, the auction market does not

Backtesting assumes that the execution price is the same as the historical price, which means there is no slippage and abundant liquidity. Slippage refers to the difference between the execution price and the expectation price; Liquidity refers to an order that can be executed quickly without changing the price drastically.

The auction market refers to the executed price determined by matching the

highest bid and the lowest ask. The liquidity of the auction market is determined by the quantities of the bid and ask. Therefore, liquidity is difficult to measure, and the slippage will undermine the result in real trading/investment [4].

Dealer market refers to the dealer quoting a bid and an ask, which means the prices the dealer is willing to buy and sell, and the difference between bid and ask is the dealer's profit. The dealer is also considered a liquidity provider. Therefore, the backtesting on the dealer market is close to real trading/investment.

The stock market, future market, and option market are auction markets. The foreign exchange market is the dealer market. Therefore, the performance of the foreign exchange market in practical action has less slippage and abundant liquidity.

Guideline 6 Make the objective function learn the return instead of the price when the trading strategy is specified

Suppose the backtesting aims to measure the performance of trading strategies. In that case, the objective of machine learning should be the return of the trading strategy because the calculation of return requires at least two predictions of price (price of long position and price of short position), which means the error will double if the objective function is the prices. Therefore, return is a better metric when the trading strategy is specified compared to MSE, MAE, and MAPE [2].

Conclusion

Financial time series data is sequential and numerical data. If the backtesting does not tackle the following details properly, the result is biased and impractical. The training, validation, and testing data cannot be disrupted, and naïve prediction of previous value is the baseline of machine learning performance. As for the objective of machine learning, values of trading strategy return are better than price values. The dealer market provides abundant liquidity; however, the auction market depends on the amount of bid and ask orders.

References

[1] Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Advances in Knowledge Discovery and Data Mining: 13th Pacific-Asia Conference, PAKDD 2009 Bangkok, Thailand, April 27-30, 2009 Proceedings 13 (pp. 475-482). Springer Berlin Heidelberg.

[2] Hu, Z., Zhao, Y., & Khushi, M. (2021). A survey of forex and stock price prediction using deep learning. Applied System Innovation, 4(1), 9.

[3] Lu, W., Li, J., Wang, J., & Qin, L. (2021). A CNN-BiLSTM-AM method for stock price prediction. Neural Computing and Applications, 33, 4741-4753.

[4] Schwartz, R. A., Francioni, R., & Weber, P. (2020). Market Liquidity: An Elusive Variable. The Journal of Portfolio Management, 46(8), 7-26.

[5] Ying, X. (2019, February). An overview of overfitting and its solutions. In Journal of Physics: Conference series (Vol. 1168, p. 022022). IOP Publishing.