To appear in *Quantitative Finance*, Vol. 00, No. 00, Month 20XX, 1–5

A Note on Spurious Model Selection

Weiguan Wang† and Johannes Ruf*‡

†School of Economics, Shanghai University. Email: weiguanwang@shu.edu.cn
‡Department of Mathematics, London School of Economics and Political Science. Email: j.ruf@lse.ac.uk

(*Submitted June 18 2022*)

Breaking time structure leads to overestimating model performance, even if the model concerns only a single time period.

1. Motivation

Backtesting is concerned with studying the performance of a model on historical data. The data usually are subject to a time series structure. Handling such data incorrectly can introduce a strong bias in the evaluation of a model's performance. This is true even if the model of interest concerns only one period and hence is a priori not exposed to the time series' auto correlation.

Information leakage is introduced if the in- and out-of-sample set split does not take into account the intrinsic time series structure, for example, if the dataset is randomly split. In this case, the in-sample set may contain some information from the out-of-sample set that would not be available if the data split was done in pseudo real-time, which puts the earlier part of the data into the in-sample and the later part into the out-of-sample set. A random data split can lead to an overestimated model performance on the out-of-sample set, as its samples are not anymore independent from the in-sample set, and even more so for complex models. This issue is often obscured when studying complex models without a relevant time structure (e.g., studying models concerned with only a single time period).

We decided to write this short comment after noticing a few published papers that split time series data randomly. An extended version of this comment with more details and pointers to the relevant literature is available on SSRN (Wang and Ruf (2022))[1].

We believe the wrong split into in-sample and out-of-sample sets is usually done with the best intentions, sometimes with statistical cross validation in mind and sometimes by carelessly using standard software. It is known that improper cross-validation potentially leads to overfitted models (Opsomer *et al.* (2001), Bergmeir and Benítez (2012)). In several research works, blocked forms of cross-validation are argued to perform favourably in out-of-sample testing that preserves pseudo real-time (Burman *et al.* (1994), Racine (2000), Hall *et al.* (2004), Bergmeir *et al.* (2014), Bergmeir *et al.* (2018)). In this brief comment, we do not discuss block cross-validation but rather illustrate the information leakage of random data splits with real and simulated panel data.

*Corresponding author. Email: j.ruf@lse.ac.uk
[1]The code to reproduce the results in this paper can be found at https://github.com/weiguanwang/Information_Leakage_in_Backtesting.git.

2. Experimental setup

2.1. *Backtesting the hedging of options*

To illustrate how information may leak into the out-of-sample set we consider the following setup. The goal is to find the best one-period hedging strategy δ from a large class of functions by backtesting. This strategy is supposed to minimise the squared hedging error

$$\left(\delta S_1 + \left(1 + r\frac{1}{252}\right)(C_0 - \delta S_0) - C_1\right)^2. \tag{1}$$

This corresponds to the squared end-of-day wealth of an institution that sold a call, bought δ shares of the underlying on the previous trading day, and used the money market account for the remainder. In our survey paper Ruf and Wang (2020), we point to the large research body on this statistical hedging problem.

We shall compare two statistical models (function classes for the hedging strategy δ). These two models are among the best performing models studied in Ruf and Wang (2021). The first model assumes that the hedging strategy δ is a linear function (LR, for 'linear regression') of option characteristics, while the second one represents δ as an artificial neural network (ANN):

$$\delta_{\mathrm{LR}} = f_{\mathrm{LR}}(\delta_{\mathrm{BS}}, \mathcal{V}_{\mathrm{BS}}, \mathrm{Vanna}_{\mathrm{BS}}); \qquad \delta_{\mathrm{ANN}} = f_{\mathrm{ANN}}(\delta_{\mathrm{BS}}, \mathcal{V}_{\mathrm{BS}}, \sigma_{\mathrm{impl}}\sqrt{\tau}).$$

Here, δ_{BS}, $\mathcal{V}_{\mathrm{BS}}$, and $\mathrm{Vanna}_{\mathrm{BS}}$ represent the Black-Scholes (BS) greeks Delta, Vega, and Vanna, respectively, calculated under the BS model with the option's implied volatility, and $\sigma_{\mathrm{impl}}\sqrt{\tau}$ represents the square root of total implied variance.

2.2. *The data*

The experiment below is repeated on simulated and real-world data. The simulated data are generated from the BS model (according to the CBOE rules). The real-world data are end-of-day S&P 500 option data (SPX) obtained from OptionMetrics (2010-2019). Each of these datasets constitute panel data, i.e. cross sections of time series. There exist, at any point of time, several options that are different in strike and/or maturity.

Let us next describe a specific data point. Each point describes one out-of-the-money option over one period (1 day). The data point contains the option price at the beginning and end of the period and the underlying's price at the end of the period. Moreover, the data point includes a flag indicating whether the option is a call or a put, the risk-free rate, the strike and time-to-maturity of the option, its implied volatility, and its BS sensitivities.

We shall see that information leakage caused by the wrong data split becomes even more significant if the data are additionally 'tagged.' To explain what we mean, consider the situation that on any trading day we have an additional observation, say the daily value of the VIX (Volatility Index). We will argue that such additional features might lead to spurious model performance under random data splits. For the sake of this experiment, we want to ensure that this additional feature has nothing to do with the rest of the data. Hence, for both the simulated and real datasets, we shall use an independently (from all other data) sampled Ornstein-Uhlenbeck process as the additional feature. We call this feature 'fake VIX' to remind ourselves that it has nothing to do with any real-world observations.

2.3. *Separation into in-sample and out-of-sample sets*

For panel data as used here, different kinds of splits can be employed, e.g. chronological or random, as shown in Figure 1. When performing a chronological split ('pseudo real-time'), first a critical date

is determined. Samples from days before this point constitute the in-sample set and the remaining ones the out-of-sample set. Alternatively, the data could be split at random into in-sample and out-of-sample sets. In this approach, the in-sample and out-of-sample sets are also disjoint. However, we shall argue that such an approach introduces significant information leakage. Indeed, on each day several options are traded. Hence samples from the same day might show up in both the in-sample and out-of-sample sets simultaneously.

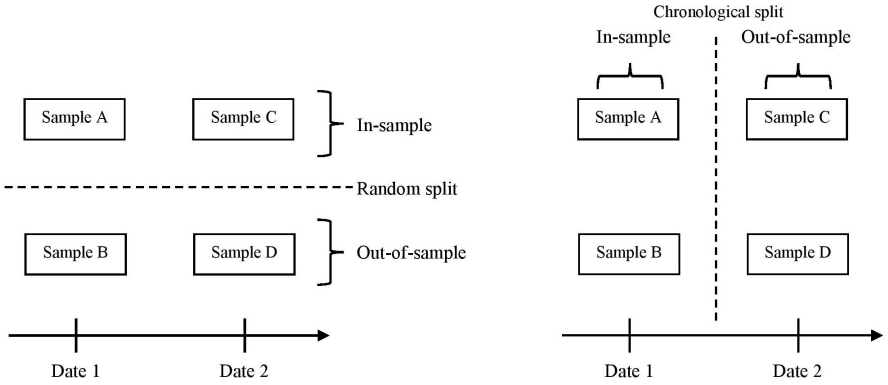

Figure 1.: Illustration of the random (left) and chronological (right) data splits. Each day in this illustration has two one-period samples.

2.4. *Four experimental configurations*

Each dataset will be pre-processed in four different ways ('configurations') as follows. The first two configurations involve a chronological split. The remaining two rely on a random split.

(i) The '*Baseline*' configuration corresponds to the standard setup. The dataset is separated chronologically into an in-sample and an out-of-sample set.
(ii) The '*VIX*' configuration takes the 'Baseline' configuration, but adds the simulated 'fake VIX' variable as an additional feature in the linear regression and the ANN.
(iii) The '*Permute*' configuration corresponds to the random split into in-sample and out-of-sample sets.
(iv) The '*Permute + VIX*' configuration is as the '*Permute*' configuration, but now with the 'fake VIX' variable as an additional feature.

3. Presence of information leakage

We now present and interpret the out-of-sample performance of the two models under the four configurations. For $\delta = \delta_{\mathrm{LR}}$ and $\delta = \delta_{\mathrm{ANN}}$ we compute the average of the values in (1) across all out-of-sample data points. We focus on the reduction in out-of-sample mean-squared hedging error relative to the hedging error when using the BS delta δ_{BS}. The two panels in Figure 2 summarise the results on the BS and the S&P 500 datasets, respectively.

For the simulation data, in the 'Baseline' configuration, neither of the statistical models has a better mean-squared hedging error than the BS delta. For the S&P 500 options, both statistical models lead to a hedging performance improvement of about 19% relative to using the BS delta. The following points summarise the outcomes of the experiment for the three other configurations.

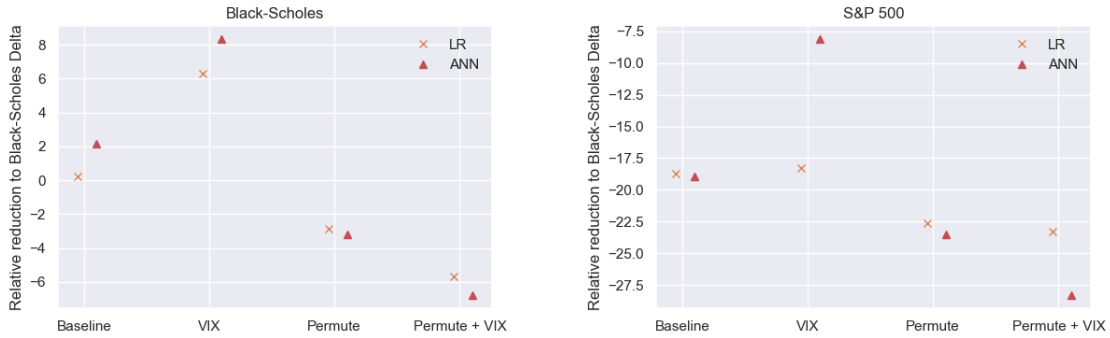

Figure 2.: Illustration of information leakage when failing to take into account the time series structure of a simulated dataset.

Comment 1: Adding an additional noisy feature leads to a worse out-of-sample performance when time structure is preserved.

In the 'VIX' configuration, both the linear regression and the ANN perform worse than in the 'Baseline' configuration, with the change for the ANN more pronounced. Indeed, the additional feature is simulated completely independently from the rest of the data. Hence, it has no predictive power for the hedging ratio at all.

Comment 2: A random data split leads to an improved performance of the statistical models, and more so for the more complex model.

This is also true when the data are generated by a time-homogeneous BS model. Since the discrete time steps are small, we know, a priori, that BS hedging is close to optimal. Nevertheless, instead of underperforming by about 0.2% for the linear regression and 2% for the ANN, the linear regression and ANN reduce the BS Delta benchmark in the BS data by about 3% after data permutation, with a larger relative improvement for the ANN. For the S&P 500 dataset, permuting samples allows the linear regression and ANN to reduce the BS Delta benchmark by about 23%, instead of about 19% in the 'Baseline' configuration.

Comment 3: Noisy features may increase information leakage if data are randomly split

If including 'fake VIX' as an additional feature when permuting samples, both statistical models improve, but most dramatically the ANN, which now outperforms the BS Delta benchmark by about 7% in the BS simulated data and by about 29% in the S&P 500 data. What is going on? By construction, each day has several options (corresponding to different strikes) but only one 'fake VIX' value. The random permutation now allows samples from the same day to appear both in the in-sample and out-of-sample sets. The presence of the additional feature makes it possible for the ANN (and partially also for the linear regression model) to understand from which day a sample is. In other words, the 'fake VIX' tags the different days and the models are able to pick up on it. This is relevant since on any specific day the underlying's price goes up or down (or, in case of the S&P 500 data, there is a certain shift in the implied volatility surface). This leaked information improves the models' hedging performance in backtesting but of course would not be available in real time.

4. Conclusion

Even for a linear regression model with few parameters, a faulty data split may lead to remarkably overconfident estimates of the model's performance. In addition, a more complex model (such as an ANN) may be more prone to information leakage. This might lead to the wrong conclusion that such a model outperforms the simpler one in a direct comparison.

Information leakage is further reinforced when data are 'tagged' by the presence of an additional

feature that takes the same value for different samples from the same date. Then random permutations make this feature informative (despite it having nothing to do with the data-generating mechanism). This yields a further seemingly important improvement for a model's performance, not achievable when applying the model in real time.

References

Bergmeir, C. and Benítez, J.M., On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 2012, **191**, 192–213.

Bergmeir, C., Costantini, M. and Benítez, J.M., On the usefulness of cross-validation for directional forecast evaluation. *Computational Statistics & Data Analysis*, 2014, **76**, 132–143.

Bergmeir, C., Hyndman, R.J. and Koo, B., A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 2018, **120**, 70–83.

Burman, P., Chow, E. and Nolan, D., A cross-validatory method for dependent data. *Biometrika*, 1994, **81**, 351–358.

Hall, P., Racine, J. and Li, Q., Cross-validation and the estimation of conditional probability densities. *Journal of the American Statistical Association*, 2004, **99**, 1015–1026.

Opsomer, J., Wang, Y. and Yang, Y., Nonparametric regression with correlated errors. *Statistical Science*, 2001, pp. 134–153.

Racine, J., Consistent cross-validatory model-selection for dependent data: hv-block cross-validation. *Journal of Econometrics*, 2000, **99**, 39–61.

Ruf, J. and Wang, W., Neural networks for option pricing and hedging: a literature review. *Journal of Computational Finance*, 2020, **24**, 1–46.

Ruf, J. and Wang, W., Hedging with Linear Regressions and Neural Networks. *Journal of Business & Economic Statistics*, 2021, **SSRN 3580132** Forthcoming.

Wang, W. and Ruf, J., Information Leakage in Backtesting. *SSRN 3836631*, 2022.