Forecasting Realized Volatility with Kernel Ridge Regression

Blake LeBaron *

International Business School

Brandeis University

October 2019

Abstract

This paper explores a common machine learning tool, the kernel ridge regression, as applied to financial volatility forecasting. It is shown that kernel ridge provides reliable forecast improvements to both a linear specification, and a fitted nonlinear specification which represent well known empirical features from volatility modeling. Therefore, the kernel ridge specification is still finding some nonlinear improvements that are not part of the usual volatility modeling toolkit. Various diagnostics show it to be a reliable and useful tool. Finally, the results are applied in several dynamic trading strategies to judge the value added over the current forecasting methods.

Keywords: Machine learning, realized volatility, kernel ridge regression

*415 South Street, Mailstop 32, Waltham, MA 02453 - 2728, blebaron@brandeis.edu, www.brandeis.edu/~blebaron.

1 Introduction

Machine learning (ML) tools have been dramatically changing the world of data analytics. They have been applied in almost all fields with enough data to make improved predictive modeling a possibility. Even though the basic ML toolbox has been around for over 50 years, modern computing, data availability, and improved algorithms have invigorated this area of research. Finance is obviously a major area for application of these tools.[1] It is obvious why primary interest is directed at forecasting returns and developing trading strategies. Economic gains in this area have the potential to be quite large. However, they run up against efficient market limits that imply predictability of market returns should be difficult if not impossible. This makes this modeling space difficult, and more importantly, difficult to perform clear model comparisons in a world where the signal to noise ratio is very low. This paper turns to modeling volatility, where predictability is higher, and selecting good predictive models may have a larger impact.[2]

The conditional variance of returns, or volatility, is very predictable. This feature has been known for a long time, and has led to the development of many useful models and procedures.[3] Early work concentrated on squared and absolute daily returns. In the early 1990's a revolution occurred with the use of high frequency data to estimate variances over a given day using intraday data. Known as realized volatility, it provided more accurate measures of volatility and its dynamics.[4]

The purpose of this paper is to develop and explore predictive models for daily realized volatility time series using a kernel ridge regression. This machine learning tool is capable of capturing a rich set of nonlinear features in the data. There are many methods in the standard machine learning tool set including, nearest neighbors, support vectors, ridge and lasso regressions, regression trees, and random forests. There are also deep neural networks which form much of the basis for modern image classification. A full comparison of all these tools is beyond the scope of this paper. Also, which tool is well suited for time series analysis is still an open question. Analysis here will concentrate on the kernel ridge regression. Recently, kernel ridge has been applied in macro forecasting in Exterkate, Groenen, Heij & van Dijk (2016). It looks like a promising tool for exploring a large rich set of nonlinear features while avoiding model over fitting.

[1] ML research in finance is not new. In the 1990's there was an earlier wave of interest. Several early examples include Diebold & Nason (1990), LeBaron (1992), Meese & Rose (1990), and Mizrach (1992). LeBaron (1998) brings together many tools considered important today. These include neural networks, evolutionary learning, and boostrap/cross-validation systems in the search for improved foreign exchange trading strategies.

[2] Examples of machine learning tools for volatility prediction are Andrada-Felix, Fernandez-Rodriguez & Fuertes (2016), Audrino & Knaus (2016), Chen, Hardle & Jeong (2010), Luo, Zhang, Xu & Wang (2017), and Lux, Hardle & Lessmann (2018).

[3] See Andersen, Bollerslev, Christoffersen & Diebold (2006) for one of many surveys in this large area of research.

[4] This area is now very large, and is a standard for volatility modeling. See Barndorff-Nielsen & Shephard (2010) and Andersen, Bollerslev, Christoffersen & Diebold (2013) for two of the many surveys available.

It is closely related to support vector machine regressions, but these use a slightly nonstandard objective function that makes comparisons to other time series results in econometrics more difficult. By utilizing a flexible functional form, kernalized methods in machine learning can theoretically represent any possible nonlinear relationship. This makes them a good approach in searching for possible volatility nonlinearities.

Volatility may be much more predictable than returns, but it provides its own set of challenges. First, the known predictable features mean that the bar is now higher in terms of forecast comparisons. Just being able to forecast volatility is not interesting. The question is whether a model is able to beat a well defined benchmark. This opens the question of exactly what this benchmark should be. This paper searches through some common examples from the recent financial time series literature. Second, volatility has some patterns that suggest either formal long memory processes, or potential regime shifts that might make building smaller parsimonious models difficult.

Section 2 will introduce the data and linear and nonlinear models that will be used. Section 3 estimates the various models and measures the comparisons across the standard models along with the kernel ridge target. Section 3.4 applies the forecasts to several dynamic trading strategies to measure the economic significance of the results, and section 4 concludes.

2 Data and methodology

2.1 Data sources

This paper uses intraday data from the Dow Jones Industrial index provided by Tick Data. The series begins on April 1, 1993, and continues through March 2019. The series is sampled at a 5 minute frequency within the trading day only.[5] This high frequency series is used to aggregate up to a daily series. The Dow is used for several reasons. First, it is a common, well monitored, index, of relatively large stocks, and is available on a long time series at the intraday frequency. Second, there are intraday series on the Dow that go back to 1933. In other research, LeBaron (2018), I have used the modern data and the older data to build very long volatility time series. This paper concentrates on the modern data only.

Estimating daily realised volatility follows the standard procedure of summing the 5 minute squared returns,

$$RV_t^2 = \sum_{h=1}^{H} r_{t,h}^2, \tag{1}$$

[5]See Liu, Patton & Sheppard (2013) for evidence on why 5 minute returns are a good solution to many problems with high frequency series.

where $r_{t,h}$ are 5 minute log price differences. A key test will be implementing a volatility control strategy which depends on estimates of the daily standard deviation. For most tests here, the sample standard deviation, $RV_t = \sqrt{RV_t^2}$, will be used rather than the daily variance.

A second series used will be the realized quarticity defined by,

$$RQ_t = \frac{H}{3}\sum_{h}^{H} r_{t,h}^4. \tag{2}$$

This measure of within day fourth moments is used in Bollerslev, Patton & Quaedvlieg (2016) to augment and improve traditional volatility forecasting models.

Finally, daily returns will be measured within the day using the open and close on the Dow. These will be important for the trading strategies which consider only trade within the day, and ignore overnight periods. Forecasting overnight volatility adds a extra element of forecasting complexity which will not be considered here. $R_{t,oc}$ represents daily arithmetic open to close returns on day t. Also, close to close returns will be used as an input into volatility forecasts, and are given by $R_{t,cc}$ which represents an estimate from the close on day $t-1$ through the close on day t. In both cases log returns are represented by $r_{t,oc}$, and $r_{t,cc}$ respectively. These are built into a daily series containing 6546 observations.

2.2 Predictive tools

Volatility is persistent, and predictable using many different models. For realized volatility standard linear time series models have been shown to work well in forecasting near horizon future volatility. One model, introduced in Corsi (2009), has proved particular useful, and is a standard for forecasting. Corsi's approach is to construct weekly and monthly volatility measures which will be used in forecasting. This done here using the standard deviations of daily volatility.

$$RV_{t|t-m} = \frac{1}{m}\sum_{j=0}^{m-1} RV_{t-j}. \tag{3}$$

for $m = 5$, and $m = 22$. Also, for returns a similar definition holds,

$$r_{t:t-m} = \frac{1}{m}\sum_{j=0}^{m-1} r_{t-j,cc}. \tag{4}$$

The standard model proposed in Corsi (2009) for volatility forecasting is then a modified autoregression based on the composite time series,

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \mu_{t+1}. \tag{5}$$

This model has connections to LeBaron (2001) which demonstrates the usefulness of a three factor approach to long memory correlation features in volatility time series.

It is also well known that returns themselves help to predict future volatility. Volatility is more likely to rise in a falling market. This is known as the leverage effect.[6] A symmetric augmentation of the previous model yields,

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \beta_6 r_{t|t-21} + \mu_{t+1}. \tag{6}$$

Recently, these linear forecasting rules have been modified using several nonlinear forms. Bollerslev et al. (2016) propose the Realized Quarticity (RQ), and interact it with the coefficient on the first lag of volatility,

$$RV_{t+1} = \beta_0 + (\beta_1 + \beta_{1Q} RQ_{t-1}^{1/2}) RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \beta_6 r_{t|t-21} + \mu_{t+1}. \tag{7}$$

The basic intuition for this is simple. As a fourth moment it is a proxy for the precision of the estimate of the second moment. With a less precise estimate of RV_t the coefficient should be reduced through β_{1Q}. This model often does not carry all these components. A much more parsimonious version will be used. A second nonlinear structure is proposed in Wang & Yang (2017). Their functional form also attenuates the first coefficient again according to,

$$RV_{t+1} = \beta_0 + (\beta_1 + \beta_{|r|} |r_t| + \beta_r r_t + \beta_{RQ} RQ_t^{1/2}) RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \beta_6 r_{t|t-21} + \mu_{t+1}. \tag{8}$$

In this case returns and absolute returns are allowed to impact the persistence of volatility.

[6] This feature was discovered in Black (1976), and named in Christie (1982) as being related to the amount of leverage at various firms. It has recently been reexamined in Hasanhodzic & Lo (2011). Glosten, Jagannathan & Runkle (1993) and Nelson (1991) are useful models, and a more modern approach is in Curci & Corsi (2012).

2.3 Kernel Ridge Regression

Kernel ridge regression is a function approximation system that can be applied in a regression context. However, it is most often seen as part of a support vector system used for classification.[7]

This nonlinear regression utilizes two important tools from machine learning. First, a ridge regression penalty allows the model to be fit in a potentially high dimensional nonlinear space while keeping overfitting problems under control. Second, a method known as the "kernel trick" makes the estimation and forecasting problem computationally tractable.

First, the overfitting issues is addressed. Start with a traditional linear regression,

$$y = X\beta + \epsilon$$

where y is a $Tx1$ vector, X is a TxN matrix, and β is a $Nx1$ vector of parameters. T represents the number of observations, and N the number of predictor variables. Standard linear regression objectives minimize,

$$\sum_{t=1}^{T}(y - X\hat{\beta})^2, \tag{9}$$

with $\hat{\beta}$ representing a traditional OLS estimate of the parameters. For N large relative to T, it is highly likely this regression will yield poor out of sample predictions, as it will overfit the training samples. Ridge regression can ameliorate this issue by adding the ridge penalty as in,

$$\sum_{t=1}^{T}(y - X\hat{\beta})^2 + \lambda\sum_{i=1}^{N}\beta_i^2. \tag{10}$$

The regularizing term shrinks the β estimates to zero and reduces overfitting in overly complex regression problems.

In the nonlinear space the regression becomes,

$$y = f(X) + \epsilon.$$

Build an approximation to $f(X)$ with a function, $\phi : \mathbb{R}^N \to \mathbb{R}^M$ and a transformed variable, $z_i = \phi(x_i)$ which is now an M length vector. Also, represented by Z, a TxM matrix. Assume that $f(x_i)$ is linear in

[7] Classic textbook references can be found in Hastie, Tibshirani & Friedman (2009), and Murphy (2012). Also, there are excellent descriptions in Exterkate et al. (2016) and Exterkate (2013).

z, or $f(x) = z_i'\gamma$, $\gamma \in \mathbb{R}^M$. Now we have a linear regression in the transformed space of z. At first, this sounds good, but in a general nonlinear space $M >> N$, and also it is possible that $M > T$. The regression would most likely overfit yielding relatively useless out of sample predictions. The ridge penalty can be used again to restrict the regression coefficients, γ, in the new space.

All should theoretically work fine with this approach, but the computational costs of this regression are large. The kernel trick maps the problem into another more computationally tractable space. First, the standard regression solutions would involve $Z'Z$, or a MxM matrix. For M large this could be problematic. This matrix can be replaced with ZZ', a TxT matrix. At first, this might not seem like much of an improvement, but for large nonlinear expansions, M might be much larger than T, so the TxT matrix is an improvement. The second part of this is to realize that ZZ' involves cross products across $\phi()$ as in $\phi(x_i)'\phi(x_j)$. The final step is to use special kernel functions designed to give $k(a,b) = \phi(a)'\phi(b)$ where $k(a,b)$ can be computed quickly. We now can get simpler estimates for ϕ, and therefore $f(x)$, which are functions of the kernel function. For example a forecast for a new point x takes the form,

$$f(x) = \sum_{t=1}^{T} \alpha_t k(x, x_t). \tag{11}$$

Although, still appearing computationally costly, this can be calculated relatively fast, and avoids problems with possibly large values for M.

There are many kernel functions in use. This paper will concentrate only on the radial basis, or gaussian kernel,

$$k(x, x_t) = \exp(-\theta||x - x_t||^2). \tag{12}$$

Intuitively, it measures the distance between two points, giving nearer points larger weight. It is normalized with a bandwidth parameter, θ. Nonlinear functions can now be efficiently estimated, and used for forecasting.

Two final issues remain. First, there are two hyper-parameters that need be tuned. These are θ and λ, the bandwidth and ridge penalty respectively. They are optimized using a grid search which implements a five fold randomized cross validation (250 runs) over a large sequence of parameter pairs. The pair with the smallest test sample MSE is used as the optimal tuned parameters for the kernel used in all later experiments.

Second, as in many nonlinear models there is an issue of data normalization. It is often standard procedure to preprocess all inputs by subtracting the mean, and dividing by the standard deviation. This

nontrivial transform can have a big impact as these inputs are sent into the nonlinear kernel. The standard kernel function often involves a single bandwidth parameter, θ. Technically, if this were following standard nonparametric procedures the bandwidth would vary across all the input dimensions adding a relatively high dimensional hyper-parameter. In this paper inputs will be normalized, but in a relatively careful fashion. First, there are no mean adjustments. Since the radial kernel is distance based, it is not affected by fixed shifts in the input data. The *RV* inputs are divided by their standard deviations estimated in a given training sample. Since returns are close to uncorrelated over time the N period mean returns scale at approximately $1/\sqrt{N}$, so they are multiplied by $\sqrt{N}$, and then all three lags (if all are used) are divided by the standard deviation of the single period return estimated in the training data. This procedure gives an extra set of 4 parameters estimated in training data.

2.4 Randomized cross validation

Most of the model testing will use a form of 5-fold randomized cross validation to both evaluate performance, and also as a model comparison tool. The time series will be combined into (target, predictor) pairs (y_{t+1}, X_t) where y_{t+1} is the target volatility, RV_{t+1}, and X_t represents a vector of information available at time t. This includes lags, and sums of lags, of realized volatility, and returns as described. These pairs are split into two distinct random subsets, training and testing with $4/5$ in training, and $1/5$ in testing. Training will be used for model estimation, and testing for out of sample model evaluation. The data splitting processes will be performed many times in a monte-carlo generating both means and standard errors for various measures of forecast performance.

3 Empirical results

3.1 Volatility features

Figure 1 shows the Dow realized volatility time series. It is adjusted to units of annualized standard deviation. Average market volatility is usually in a range of about $0.15 - 0.20$. The mean for this series, which is presented in table 1, is well below this level. This is due to the fact that the overnight variability is not accounted for in these estimates. The figure shows typical volatility features with some unusually large spikes in volatility along with periods of persistently high and low volatility. Figure 2 displays a histogram for the time series, and clearly shows a right skew which should be expected for a series of standard devi-

ations. Figure 3 repeats this for the log realized volatility values, and confirms the well known fact that RV is close to log normally distributed.

Table 1 summarizes all the values used for prediction. The three realized volatility series report means near 10 percent which are below the full daily volatility levels in the Dow. All three series show the positive skew from the last figure, as well as excess kurtosis. Turning to the returns, they show much less skewness, but it looks strong and negative. All three return series show excess kurtosis which is common for any relatively high frequency return series. The final row reports values for the RQ_t fourth moment, or realized quarticity. It shows extreme skew and kurtosis.

Figure 4 shows the autocorrelations for the daily realized volatility series. The are positive, large, and decay very slowly. They still are positive at a lag of 250 days (about 1 year). This highly persistent behavior in volatility is another well known feature, and will influence most of the benchmark models which are used in this study.

3.2 Benchmark model selection

Since volatility prediction is well known, the choice of an appropriate benchmark model to compare with kernel approaches is critical. Table 2 estimates various versions of the model developed in (Corsi 2009). The table reports parameter estimates along with standard errors (in parenthesis). The final three columns show overall goodness of fit measures with the Bayesian Information Criterion (BIC), the in sample R-squared, and the in sample mean squared error (MSE). The latter two of these along with a simple check of coefficient significance would suggest the most highly parameterised model as expected. The BIC measure reports a tie between the last two. Given this evidence the best model using standard identification tools would be the second to last line using three volatility lags, and two return lags.

This paper takes as given a target of the next day's standard deviation as the desired forecasting object. However, there are many ways that one could proceed to estimate this. Table 3 explores several different approaches in building volatility forecasts. These approaches are inspired by the visual results from figure 3 which suggests a near log normal distribution for realized volatility. It is therefore possible that estimation in log space might be the best way to go.[8] Table 3 also implements the cross validation procedure which will be used throughout the rest of this study. 250 randomized cross validations are performed, and the means are reported in the table. Randomized sets of the sample are used for training (4/5) and out of sample testing (1/5). Out of sample, or test MSE will be used at the criterion for finding the best model. In

[8]Much of the realized volatility literature operates with log volatility measures.

all cases the best model from table 2 will be used.

Row one (RV/RV) in tabel 3 estimates the based model with raw standard deviations on both sides. It is interesting that the basic linear model shows very little evidence of in sample overfitting in that MSE and R^2 estimates are very close. The second row, labeled $RV/\log(RV)$ keeps the standard deviation target on the left side of the regression, but logs all the three standard deviation measures on the right hand side. This leads to a large decrease in R^2 both in training and testing periods. The third row replaces the left side RV value with $\log(RV)$. The final standard deviation estimate is obtained by simply exponentiating the estimated value as in $\exp(E\log(RV))$. Performance in terms of test MSE is still poor relative to the initial benchmark. The final row uses the fact that this estimate is a biased estimate of the standard deviation. It adjusts the log volatility estimate using

$$E(RV) = e^{E(\log(RV))+(1/2)\sigma^2_{RV}}, \tag{13}$$

and is referred to as the bias adjusted estimate. The standard deviation for the $\log(RV)$ value is estimated using training data only. This bias adjustment, does not help the estimate very much at all. The drop in performance is probably due to the fact that the RV levels are not precisely log normal which is required for the adjustment to work.

Results from tables 2 and 3 suggest the optimal linear comparison model should be estimated in standard deviations on both target and predictors. Furthermore, table 2 also recommends the model with 3 lags in realized volatility, and 2 in the returns part. The next table explores some of the nonlinear modifications that were suggested earlier.

In table 4 the previously described nonlinear specifications are explored to see which, if any, improve on the base model in terms of test sample forecast performance. The first row, labeled Base, uses the basic linear models with 3 lags of volatility, and two in returns. The second row, labeled Base+$r_{t|t-21}$ adds the extra return lag to recheck if it provides any meaningful improvement in terms of test sample MSE. The table shows that it does not, and agrees with our previous decisions based on BIC. The next row adds the realized quarticity as in equation 7, and it shows an MSE improvement in both training and testing samples. The next row implements the second nonlinear form from equation 8, but keeping the coefficient on lags of RQ fixed at zero. It shows improvement over the base linear model in terms of test sample MSE, but it falls short of the RQ model. Finally, all terms are added as shown in equation 8. This highly parameterized model obviously reports the smallest training MSE, but falls short of the more parsimonious Base+RQ only

model from row 3.

These results suggest two useful comparisons for the kernel ridge experiments. First, the basic linear model will be used with three volatility, and two return lags. Second, this model will be augmented with the RQ interaction term as in equation 7.

3.3 Kernel comparisons

The next experiment is a quick test of the kernel ridge regression in terms of finding known nonlinear structure. RV_t will be projected on simple functions of lagged returns and a kernel which should incorporate all nonlinear structure. The results are shown in table 5. The forecast target is the one day head RV measure. In case (A) RV is regressed on simple lagged returns. Volatility is fundamentally a nonlinear function of returns, so not much predictability should be expected in this case. The first row shows that this is definitely the case. The test sample R^2 is nearly zero. The values in the MSE column present first the mean MSE across 250 cross validation draws. The value in parenthesis presents the estimated standard error for the MSE mean. Finally, the value in [] reports the fraction of these runs generating a MSE test forecast smaller than the kernel model estimated in (D).

Obviously, at least lagged squared returns need to be used to help in forecasting future realized volatility. These are added in model (B), reducing MSE, and bringing the test R^2 to 0.154. The value in brackets again shows that the MSE is still bettered by the kernel ridge model. The final standard specification adds the lagged return to capture the volatility asymmetries from rising and falling markets. Forecast performance again improves. The final row presents the estimates from the kernel ridge model using only a simple lagged return as input. It is then able to nonlinearly transform this value to develop an optimal nonlinear specification. It is clear in the table that this value is less than all three of the others by a significant amount. None of the standard models can achieve the test sample forecast performance of kernel model. This shows that it is finding some interesting nonlinear patterns beyond simply combining squared returns and returns in a linear structure. There must be some more complex nonlinear pattern at work in the data.

These results are an interesting experiment, but they are not the primary objective in this paper since lagged returns and squared returns are not optimal forecasts for realized volatility. Table 6 reports core comparisons for the paper. In this table the linear and nonlinear models from equations 6 and 7,

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \mu_{t+1},$$

and,

$$RV_{t+1} = \beta_0 + (\beta_1 + \beta_{1Q} RQ_{t-1}^{1/2}) RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \mu_{t+1},$$

respectively, are compared with the kernel regression using,

$$RV_{t+1} = f(RV_t, RV_{t|t-4}, RV_{t|t-21}, r_t, r_{t|t-4}) + \mu_{t+1}.$$

They are referred to as Linear, Nonlinear, and Kernel in table 6. The table again reports training and testing MSE's and R^2 estimates from 250 randomized cross validations. It further reports mean absolute error estimates (MAE) as well from the same runs. The first row reports estimates from the linear model. Comparing the MSE of $3.75e-6$ with the value from the last row of $3.45e-6$ shows an improvement going to to the kernel. This improvement is tested first by reporting the fraction of linear model estimates out of the 250 cross validation runs which give a value as small as the corresponding value from the kernel ridge. This value, in brackets, shows that this does not happen. This table also estimates the MSE and MAE differences between the linear model and the kernel model. These are given in the row labeled Linear difference. Both values are significantly positive (standard errors are in parenthesis) indicating larger forecast errors for the linear model.

The row labeled Nonlinear repeats the previous comparisons using the nonlinear framework. Of all the standard models used, this benchmark has shown the best performance in terms of test sample MSE. This is clear from table 6. It shows both smaller MSE and larger R^2 than the linear model, as well as smaller MAE as well. There is again improvement in MSE from moving to the Kernel model from this one. The change from $3.693e-6$ to $3.450e-6$ represents and improvement of about 6.5 percent. Again, none of the nonlinear models are able to beat the kernel in any of the 250 monte-carlos as shown by the value of 0 given in the brackets. Finally the difference in estimated MSE is shown in the line, Nonlinear difference, and is significantly positive as is the corresponding value for the MAE.

3.4 Dynamic strategies

The previous results have shown that forecasting improvements from a kernel ridge appear strong and significant against two standard volatility forecasting models. Are these results economically significant? This section will explore this using two different dynamic portfolio strategies. First, a volatility control strategy will be implemented using various comparison volatility forecasts to see how well the strategy

does in terms of hitting an exante target volatility level. Second, the estimated volatility will be used in a myopic optimal trading strategy which does not try to forecast returns, but uses the conditional variance estimate to adjust portfolio fractions.

Dynamic volatility control strategies are simple trading systems designed to hit a target volatility level.[9] Investors are assumed to move funds between a risky asset and a risk free investment giving a portfolio return of,

$$R_{p,t+1} = \alpha_t R_{t+1} + (1 - \alpha_t) R_{f,t+1}. \tag{14}$$

The standard deviation of this portfolio is

$$\sigma(R_{p,t+1}) = \alpha_t \sigma(R_{t+1}) \tag{15}$$

where $\sigma(R_{t+1})$ is the standard deviation of the risky asset. Targeting a level of volatility at σ_C gives a portfolio fraction of

$$\alpha_t = \frac{\sigma_C}{RV_{t+1|t}} \tag{16}$$

where $(RV)_{t+1|t}$ is the forecast for the one period ahead standard deviation from a given volatility model.

Table 7 presents these results. The strategy is applied to open to close returns for the target day for the volatility forecast. Portfolio returns and standard deviations are estimated in the testing period which is again clean from model estimation biases. The table again reports means from 250 cross validation runs. The target volatility is set to 0.10 per year, and numbers are reported on an annualized basis.

For the line labeled Naive, the forecast simply uses the unconditional standard deviation in the training sample. The mean value is close to 10 percent, and the MSE around the target is $2.2e-5$. Linear refers to the base linear volatility forecasting model, and it shows a dramatic improvement in the variability of the portfolio around target volatility. The two nonlinear models, Nonlinear and Kernel, show further improvements, but they are less dramatic. Moving from the nonlinear model to the Kernel ridge model shows a MSE improvement of only about 3.2 percent. As a comparison, a strategy is run using the actual realized volatility on day $t+1$. This counter factual experiment is done to test that there are dramatic gains possible from accurate volatility forecasting. Results are given in the row labeled True. The MSE about the target shows a reduction of well over 100 times, so good volatility timing is definitely something that would show up in these results.

[9] See Taylor (2017) for examples and performance comparisons.

The final column reports the Sharpe ratio for these strategies. Values are a little higher than one might expect since they are not including overnight variability. Also, this is not a strategy that is designed to improve Sharpe ratios. As a matter of fact, if risk and return are moving together dynamically as they should, then Sharpe ratios should not change much across the strategies. There is a moderate boost in the Sharpe ratio in moving to the dynamic strategies, but values across the three strategies are similar. That values are dramatically different from the experiments using the expost volatility levels. Obviously, using the future information has a dramatic impact on the performance of the strategy. Again, this experiment is just for comparison purposes, and is not implementable.

The last strategy looks at a dynamic trading system designed to fit an optimal mean/variance investor. The investor is assumed to be myopic with CRRA aversion preferences. It is well known (see Campbell & Viceira (2002)) that the optimal fraction of wealth to put in the risky asset in this case is,

$$\alpha_t = \frac{E(r_{t+1}) - r_{f,t+1} + (1/2)RV^2_{t+1|t}}{\gamma RV^2_{t+1|t}}, \tag{17}$$

where γ is the coefficient of relative risk aversion, and $RV_{t+1|t}$ is the volatility forecast for $t+1$ made at time t. $E(r_{t+1})$ uses the unconditional mean log return from the training period. Table 8 reports annualized certainty equivalent returns (CE) for values of $\gamma = [2,3,4]$ for the same set of volatility forecasting strategies from the last table. There are some improvements in shifting to a dynamic variance forecasting system (ie. moving from Naive to the other strategies), but in general the CE values across the three types of volatility forecasts are not all that different. Obviously, using a strategy with the realized expost variance, again reported in the row labeled True, shows that effective volatility forecasting is useful for these investors.

4 Conclusions

This paper has explored several nonlinear forecasting tools as applied to forecasting a daily realized volatility series for the Dow Industrials. The results show that a common machine learning tool, kernel ridge regression, was able to find nonlinear features that generated statistically significant improvements in out of sample forecasts. It was able to improve on a basic linear model, and also a nonlinear model. In volatility space, where there is a lot of structure, the kernel ridge model is able to detect and utilize some of this structure.

It should be noted that the kernel ridge approach was able to generate improved forecasting with no

complex model specification. There was no need to coax the model with constructed nonlinear feature variables. It was able to learn them itself. The nonlinear specifications used in this paper were the result of many years of model explorations.

The final results show that most of these improvements are significant to a trader using a dynamic volatility control strategy. However, the marginal gains of kernel ridge versus the nonlinear specification may not be economically important. On the other hand, kernel ridge did not require any careful model prespecification, and its usefulness may be large in areas where lots of nonlinear model exploration has not been done. Further analysis of simple trading systems, such as volatility control, is an important issue for future research.

For a realized volatility forecasting problem, kernel ridge regression is a reliable way to include nonlinear features in modeling. It also appears to be fitting nonlinear features which have not yet been discovered in standard econometric models.

References

Andersen, T. G., Bollerslev, T., Christoffersen, P. F. & Diebold, F. X. (2006), Volatility and correlation forecasting, *in* G. Elliott, C. W. J. Granger & A. Timmermann, eds, 'Handbook of Economic Forecasting', North Holland, pp. 778–878.

Andersen, T. G., Bollerslev, T., Christoffersen, P. F. & Diebold, F. X. (2013), Financial risk measurement for financial risk management, *in* G. Constantinedes, M. Harris & R. Stulz, eds, 'Handbook of the Economics of Finance', Vol. 2, Elsevier, pp. 1127–1220.

Andrada-Felix, J., Fernandez-Rodriguez, F. & Fuertes, A. M. (2016), 'Combining nearest neighbor predictions and model-based predictions of realized variance: Does it pay?', *International Journal of Forecasting* **32**(3), 695–715.

Audrino, F. & Knaus, S. D. (2016), 'Lassoing the HAR model: A model selection perspective on realized volatility dynamics', *Econometric Reviews* **35**(8-10), 1485–1521.

Barndorff-Nielsen, O. E. & Shephard, N. (2010), Measuring and modeling volatility, *in* R. Cont, ed., 'Encyclopedia of Quantitative Finance', John Wiley, pp. 1898–1901.

Black, F. (1976), 'Studies of stock price volatility changes', *Proceedings of the American Statistical Association, Business and Economics Statistics Section* pp. 177–181.

Bollerslev, T., Patton, A. J. & Quaedvlieg, R. (2016), 'Exploiting the errors: A simple approach for improved volatility forecasting', *Journal of Econometrics* **192**, 1–18.

Campbell, J. Y. & Viceira, L. M. (2002), *Strategic Asset Allocation*, Oxford University Press, Oxford, UK.

Chen, S., Hardle, W. K. & Jeong, K. (2010), 'Forecasting volatility with support vector machine-based garch model', *International Journal of Forecasting* **29**(4), 406–433.

Christie, A. A. (1982), 'The stochastic behavior of common stock variances: Value, leverage and interest rate effects', *Journal of Financial Economics* **10**, 407–432.

Corsi, F. (2009), 'A simple approximate long-memory model of realized volatility', *Journal of Financial Econometrics* **7**(2), 174–196.

Curci, G. & Corsi, F. (2012), 'Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling', *Journal of Business and Economic Statistics* **30**(3), 368–380.

Diebold, F. X. & Nason, J. M. (1990), 'Nonparametric exchange rate prediction?', *Journal of International Economics* **28**, 315–332.

Exterkate, P. (2013), 'Model selection in kernel ridge regression', *Computational Statistics and Data Analysis* **64**, 1–16.

Exterkate, P., Groenen, P. J. F., Heij, C. & van Dijk, D. (2016), 'Nonlinear forecasting with many predictors using kernel ridge regression', *International Journal of Forecasting* **32**, 736–753.

Glosten, L. R., Jagannathan, R. & Runkle, D. E. (1993), 'On the relation between the expected value and the volatility of the nominal excess return on stocks', *Journal of Finance* **48**, 1779–1801.

Hasanhodzic, J. & Lo, A. W. (2011), Black's leverage effect is not due to leverage, Technical report, Boston University.

Hastie, T., Tibshirani, R. & Friedman, J. (2009), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer.

LeBaron, B. (1992), 'Forecast improvements using a volatility index', *Journal of Applied Econometrics* **7**, S137–S150.

LeBaron, B. (1998), An evolutionary bootstrap method for selecting dynamic trading strategies, *in* A. P. N. Refenes, A. N. Burgess & J. Moody, eds, 'Advances in Computational Finance', Kluwer Academic Press, pp. 141–160.

LeBaron, B. (2001), 'Stochastic volatility as a simple generator of apparent financial power laws and long memory', *Quantitative Finance* **1**, 621–631.

LeBaron, B. (2018), A long history of realized volatility, Technical report, Brandeis International Business School.

Liu, L. Y., Patton, A. J. & Sheppard, K. (2013), Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes, Technical report, Duke University.

Luo, R., Zhang, W., Xu, X. & Wang, J. (2017), A neural stochastic volatility model, Technical report, University College London.

Lux, M., Hardle, W. K. & Lessmann, S. (2018), Data driven Value-at-Risk forecasting using a SVR-GARCH-KDE hybrid, Technical report, Humboldt University.

Meese, R. A. & Rose, A. K. (1990), 'Nonlinear, nonparametric, nonessential exchange rate estimation', *American Economic Review* **80**, 192–196.

Mizrach, B. (1992), 'Multivariate nearest-neighbor forecasts of EMS exchange rates', *Journal of Applied Econometrics* **7**, S151–63.

Murphy, K. P. (2012), *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA.

Nelson, D. B. (1991), 'Conditional heteroskedasticity in asset returns: A new approach', *Econometrica* **59**, 347–370.

Taylor, N. (2017), 'Risk control: Who cares?', *European Financial Management* **23**(1), 153–179.

Wang, J. & Yang, M. (2017), Conditional volatility persistence, Technical report, University of Technology Sydney.

Table 1: Summary Statistics

	Mean	Std	Skewness	Kurtosis
RV_t	0.113	0.0038	1.44	2.22
$RV_{t\|t-4}$	0.116	0.0042	2.82	14.1
$RV_{t\|t-21}$	0.116	0.0039	2.58	11.5
r_t	0.077	0.1732	-0.19	8.10
$r_{t\|t-4}$	0.078	0.0725	-0.74	6.01
$r_{t\|t-21}$	0.076	0.0325	-0.94	3.57
RQ_t	$1e-6$	$4e-6$	13.82	299

Summary statistics for realized volatility measures. *RV* represent daily realised volatility estimates, and are presented in units of annual standard deviations. *r* represent log returns over the different lagged horizons. For consistency the returns are also scaled up to annualized units by multiplying by 250. The final row presents the realized quarticity estimates.

$$RV_{t+1} = \beta_0 + \beta_1 RV_t + \beta_2 RV_{t|t-4} + \beta_3 RV_{t|t-21} + \beta_4 r_t + \beta_5 r_{t|t-4} + \beta_6 r_{t|t-21} + \mu_{t+1}$$

Table 2: Benchmark model fitting

β_1	β_2	β_3	β_4	β_5	β_6	BIC	R^2	MSE
0.830						$-5.932e4$	0.688	$4.65e-6$
(0.007)								
0.526	0.322					$-5.997e4$	0.719	$4.18e-6$
(0.013)	(0.012)							
0.520	0.191	0.166				$-6.009e4$	0.725	$4.10e-6$
(0.013)	(0.017)	(0.015)						
0.0469	0.224	0.174	-0.034			$-6.029e4$	0.734	$3.97e-6$
(0.013)	(0.017)	(0.015)	(0.002)					
0.447	0.190	0.221	-0.026	-0.055		$-6.034e4$	0.736	$3.92e-6$
(0.014)	(0.017)	(0.016)	(0.003)	(0.007)				
0.448	0.178	0.226	-0.0250	-0.050	-0.035	$-6.034e4$	0.737	$3.92e-6$
(0.014)	(0.018)	(0.016)	(0.003)	(0.007)	(0.015)			

Model estimation: Fitted parameters, full sample. Estimation is by OLS, and numbers in parenthesis are standard errors on the parameter estimates. For this table only, R^2 is the traditional, in sample, R^2 measure. BIC refers to the Bayesian information criterion, or Schwarz criterion used for in sample model selection. MSE is the in sample Mean Squared Error. *RV* values are in units of daily standard deviations, and *r* are logged daily returns.

Table 3: Target selection

Target/Predictors	Train MSE	Train R^2	Test MSE	Test R^2
RV/RV	3.742e-6	0.731	3.751e-6	0.731
$RV/\log(RV)$	4.019e-6	0.712	4.037e-6	0.711
$\log(RV)/\log(RV)$	3.967e-6	0.715	4.005e-6	0.712
$\log(RV)/\log(RV)$ bias adjust	4.810e-6	0.655	4.836e-6	0.653

Target model comparisons: Target refers to left hand forecast target in use, both the RV, or standard deviation target, and a log(RV), logged standard deviation. Predictors refer to right hand side variables which are either raw standard deviations or logged. The final row adjusts for bias in the log volatility predictor. Training and testing data are means across 250 randomized cross validations using $4/5$ in the training sample and $1/5$ in the testing sample.

Table 4: Model selection

Model	Train MSE	Train R^2	Test MSE	Test R^2
Base	3.742e-6	0.731	3.751e-6	0.731
Base $+r_{t\|t-21}$	3.731e-6	0.732	3.780e-6	0.729
Base+RQ_t	3.673e-6	0.737	3.678e-6	0.736
Base+Nonlin	3.690e-6	0.735	3.746e-6	0.731
All+Nonlin+RQ_t	3.641e-6	0.739	3.695e-6	0.735

Model comparisons: Base is standard linear framework. $+r_{t|t-21}$ adds extra one month lag on returns. RQ adds the interaction term with realized quarticity. Nonlin adds the general nonlinear interaction components. The final line adds all of these. All results are reported means from 250 randomized cross validations using a randomized 5 fold train/test split.

$$(A) \quad RV_{t+1} = \beta_0 + \beta_1 r_t$$

$$(B) \quad RV_{t+1} = \beta_0 + \beta_1 r_t^2$$

$$(C) \quad RV_{t+1} = \beta_0 + \beta_1 r_t^2 + \beta_2 r_t$$

$$(D) \quad RV_{t+1} = f(r_t)$$

Table 5: Primitive Model Comparisons

Model	Train MSE	Train R^2	Test MSE	Test R^2
(A)	1.354e-5	0.028	1.357e-5 (4.19e-8) [0.00]	0.026
(B)	1.174e-5	0.158	1.180e-5 (4.59e-8) [0.00]	0.154
(C)	1.141e-5	0.181	1.153e-5 (4.73e-8) [0.00]	0.173
(D)	1.014e-5	0.273	1.014e-5 (3.21e-8)	0.273

Model comparisons: Simple models compared to kernel ridge regression (D). Specifications (A), (B), and (C) represent simple modeling approaches to forecast daily realized volatility. All results are reported means from 250 randomized cross validations using a randomized 5 fold train/test split.

Table 6: Kernel Ridge Forecast Comparisons

Model	Train MSE	Train R^2	Test MSE	Test R^2	Test MAE
Linear	3.740e-6	0.732	3.758e-6 (1.50e-8) [0.00]	0.731	1.407e-3 (1.40e-6) [0.00]
Linear difference			3.120e-7 (4.37e-9)		6.016e-5 (7.56e-7)
Nonlinear	3.670e-6	0.737	3.693e-6 (1.43e-8) [0.00]	0.736	1.393e-3 (2.09-6) [0.00]
Nonlinear difference			2.427e-7 (5.06e-9)		4.586e-5 (8.33e-7)
Kernel	3.343e-6	0.760	3.450e-6 (1.33e-8)	0.753	1.347e-3 (2.03e-6)

Model comparisons: Linear represents the basic linear comparison model with three lags of RV, and 2 of returns. Nonlinear represents the addition of the realized quarticity interacted with the first lag in volatility. Differences shows the MSE and MAE differences between the appropriate model and the Kernel estimated error. Values in parenthesis are standard errors. Values in brackets represent the fraction of models able to beat the kernel in the 250 monte-carlo runs. All results are reported means from 250 randomized cross validations using a randomized 5 fold train/test split.

Table 7: Volatility Control

Volatility Model	mean(vol)	MSE(target - vol)	Sharpe ratio
Naive	0.0997	$2.21e-5$	0.52
Linear	0.0998	$8.19e-6$	0.62
Nonlinear	0.0998	$7.91e-6$	0.59
Kernel	0.1000	$7.78e-6$	0.59
True	0.1000	$1.19e-8$	1.91

Volatility control strategy results. The first column is the mean volatility for the strategy whose target is set to 0.10. The second column is the MSE of the distance to the target volatility for the simulated portfolios across 250 cross validation runs. The last column is the Sharpe ratio for the dynamic volatility control portfolios in annualized units. The strategy is implemented only within each day (open to close). Naive is a strategy using only unconditional estimates of volatility. Linear implements the linear model framework. Nonlinear adds the estimated quarticity in a nonlinear interaction, and Kernel uses the kernel ridge forecast. True substitutes in the expost observed volatility on day $t+1$ for comparison.

Table 8: Dynamic strategy: Annualized CE

Volatility Model	$\gamma = 2$	$\gamma = 3$	$\gamma = 4$
Naive	0.044	0.035	0.032
Linear	0.052	0.044	0.038
Nonlinear	0.047	0.047	0.036
Kernel	0.052	0.044	0.038
True	0.184	0.176	0.171

Dynamic portfolio strategies. Optimal portfolio holdings for different volatility forecasts, and different levels of relative risk aversion. All values are reported as annualized certainty equivalent returns. Naive is a strategy using only unconditional estimates of volatility. Linear implements the linear model framework. Nonlinear adds the estimated quarticity in a nonlinear interaction, and Kernel uses the kernel ridge forecast. True substitutes in the expost observed volatility on day $t + 1$ for comparison.

Figure 1: Dow daily realized volatility (annualized Std.)

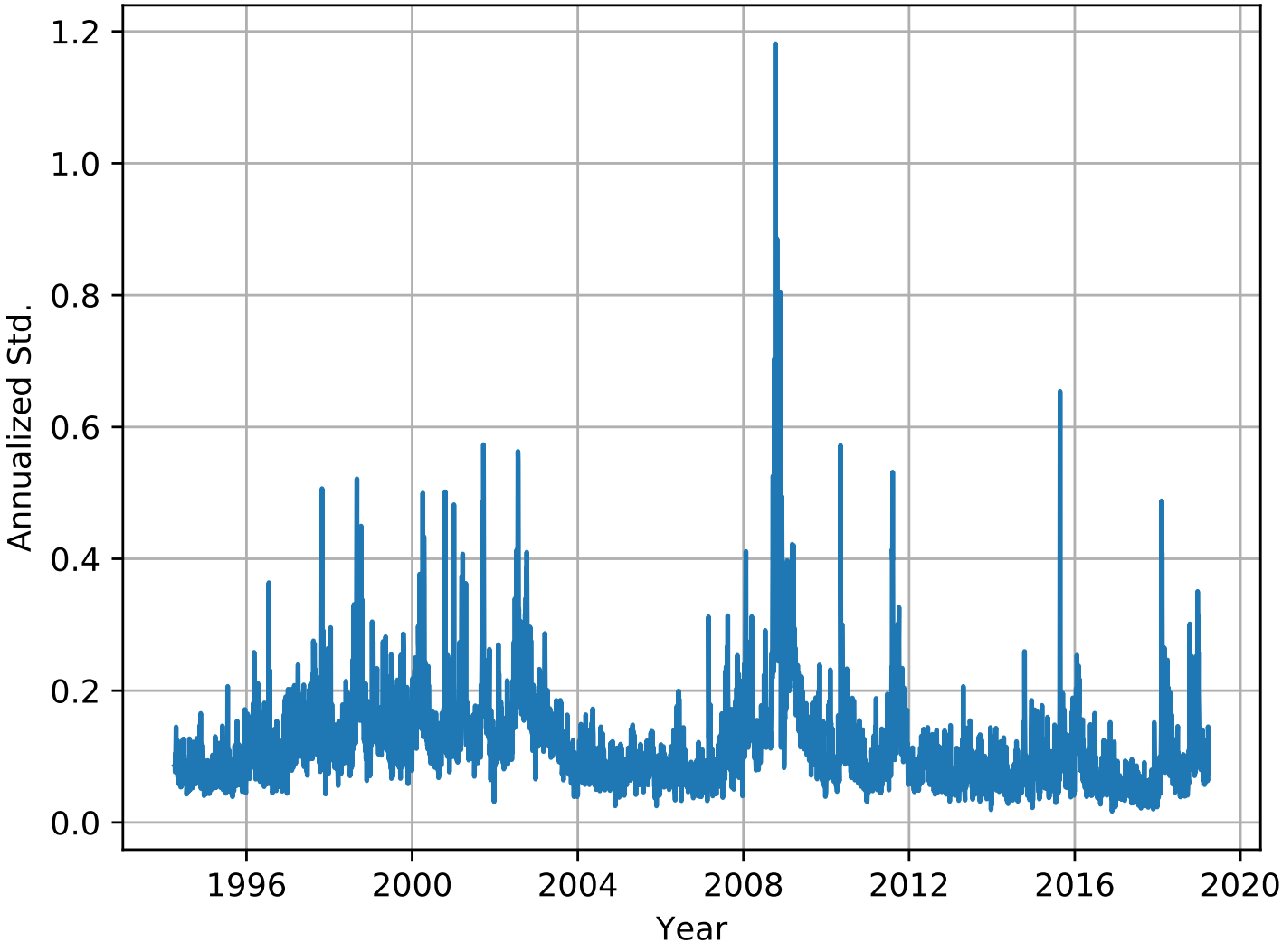

Figure 2: Dow daily realized volatility density

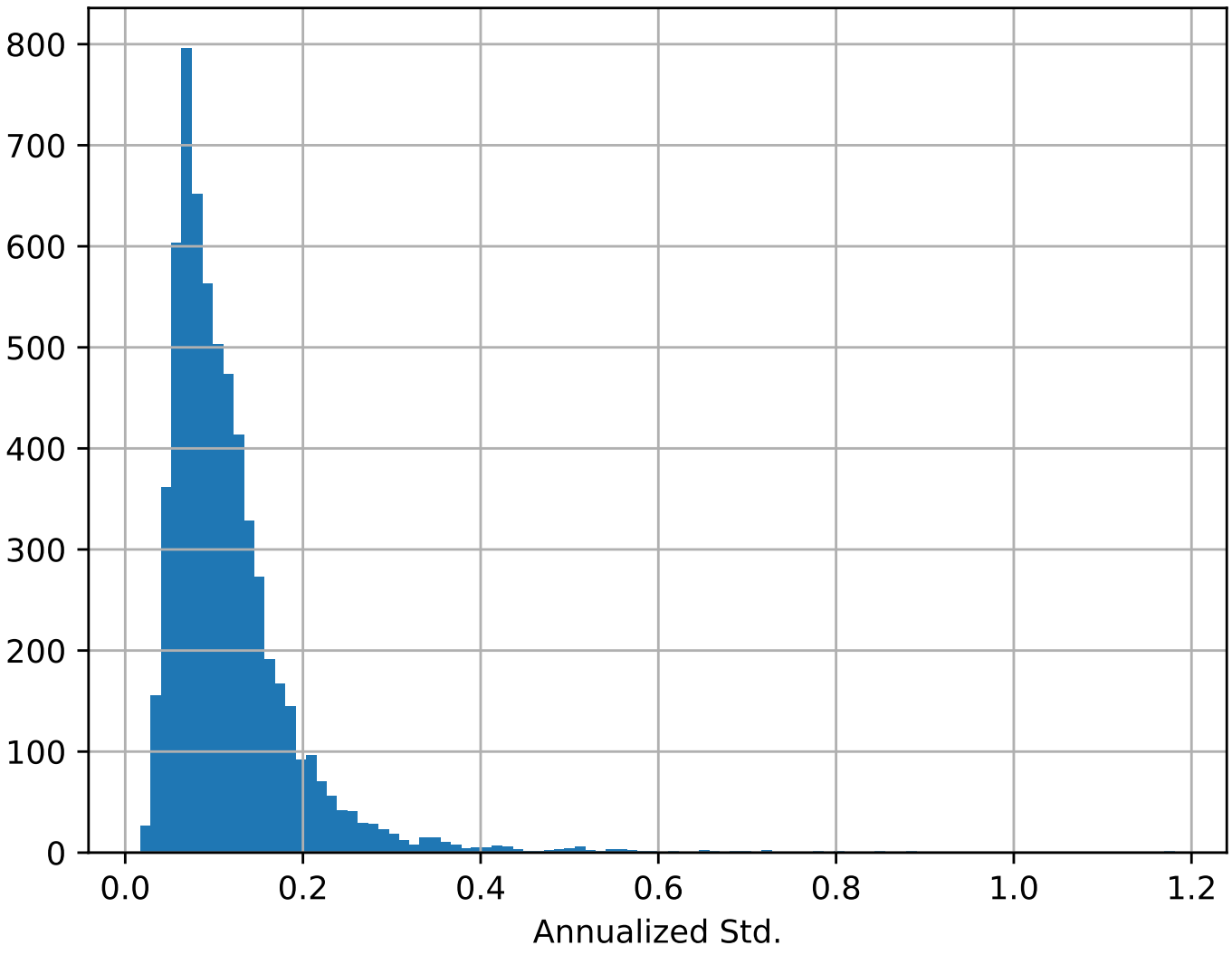

Figure 3: Dow daily realized volatility density

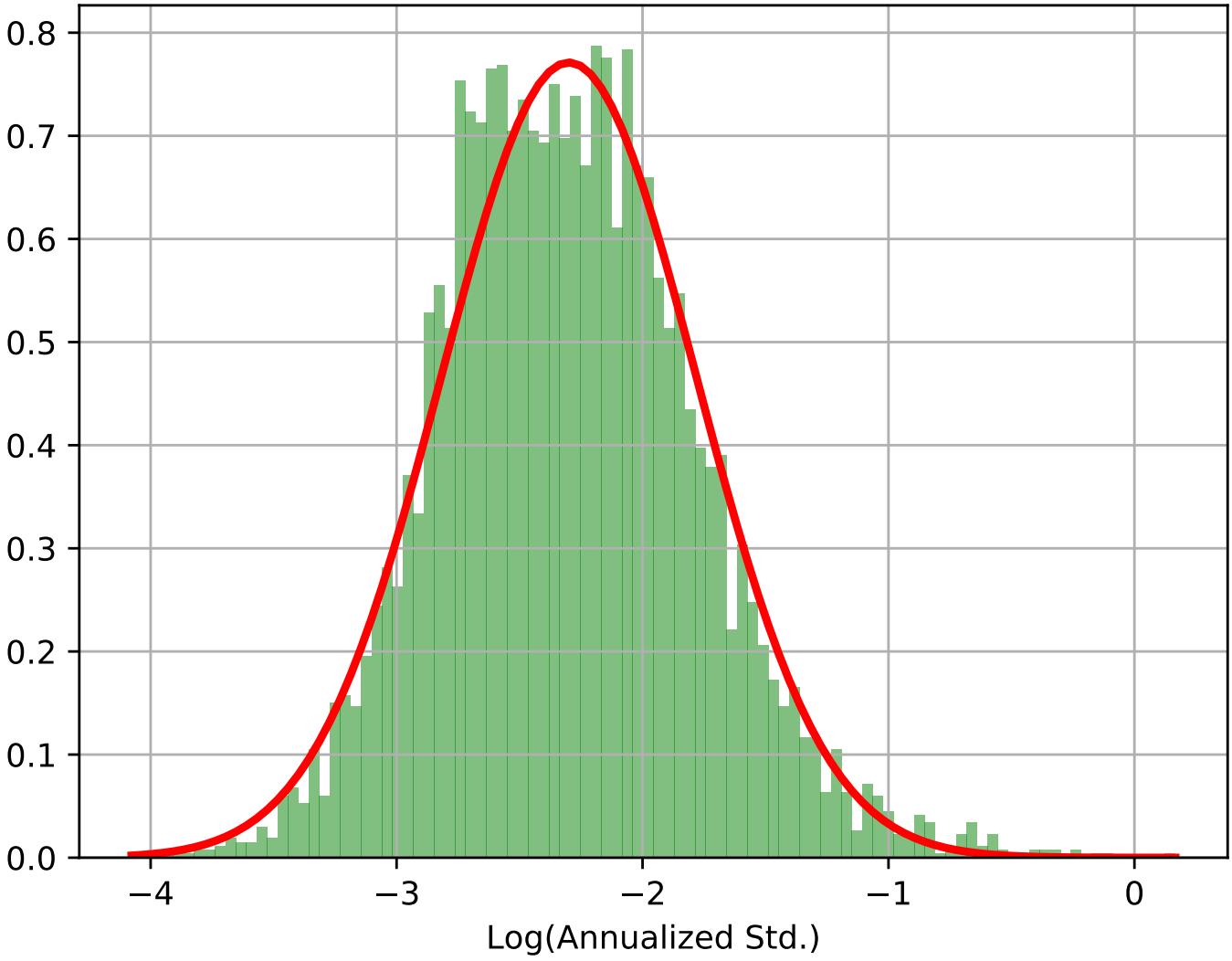

Figure 4: Dow daily realized volatility autocorrelations