

Evaluating Combined Forecasts for Realized Volatility Using Asymmetric Loss Functions

Giovanni De Luca♣, Giampiero M. Gallo♠, and Danilo Carità *♣

♣Università degli Studi di Napoli "Parthenope"

♠Università degli Studi di Firenze

Submitted: August 1, 2017 • Accepted: December 9, 2017

ABSTRACT: In this work we provide the findings of a forecast combination analysis carried out on the realized volatility series of three market indexes (DAX, CAC, and AEX). Two volatility types (5 minutes, kernel) have been considered. Different loss functions suggest that forecasts computed through combining models are generally more accurate than those provided by single models. However, the choice of the latter can significantly affect the goodness of the results.

JEL classification: C22, C53, C58

Keywords: realized volatility, forecast combinations, loss functions

Introduction

Volatility is a central parameter in many financial decisions, including the pricing and hedging of derivative products as well as the development of efficient risk management methods. Most of the volatility models presented in the literature are based on the empirical finding that volatility is time-varying and that periods of high volatility tend to cluster (Ané, 2006). The forecasting process of such an important measure represents a major issue.

*Corresponding Author. Email: danilo.carita@uniparthenope.it

In the literature there exists a wide variety of models to forecast volatility forecasts, but these are, almost by definition, simple and incomplete (Raviv, 2016). An improvement in the forecast accuracy can be achieved by combining forecasts originating from different types of models. Forecast combinations have been used successfully in empirical works in different areas such as forecasting Gross National Product, currency market volatility, inflation, money supply, stock prices, meteorological data, city populations, outcomes of football games, wilderness area use, check volume and political risks (Timmermann, 2006).

The aim of this paper is to use both single and combined models to forecast the daily realized volatility one step ahead for a one-year period. Thereafter we will compare predicted values with actual data using a number of loss functions. To carry out our analysis, we have used data on realized volatility of three market indexes (DAX, CAC, and AEX) for the period 2008 to 2016.

The remainder of the paper is organized as follows. Section 1 describes the data, the models adopted, and the loss functions used for evaluating the different forecasts. Section 2 presents the results of the analysis while Section 3 concludes the paper.

1 Data and Methodology

This study focuses on the realized volatility of three European market indexes:

- DAX 30 (*Deutsche Aktienindex 30*) is a blue chip stock market index consisting of the 30 largest German companies trading on the Frankfurt Stock Exchange;
- CAC 40 (*Cotation Assistée en Continu*) represents a capitalization-weighted measure of the 40 largest among the 100 companies with the highest market capitalizations on the Euronext Paris;
- AEX (*Amsterdam Exchange Index*) is a stock market index composed of Dutch companies that trade on the Euronext Amsterdam; it includes 25 most frequently traded securities on the exchange.

The time series of the indexes are provided by the Oxford-Man Institute of Quantitative Finance by means of its own website (<http://realized.oxford-man.ox.ac.uk/data>). For each asset, the dataset contains the realized volatility collected every 5 minutes, the realized kernel volatility (in both cases denoted by rv_t), and the daily returns (denoted by r_t), covering the period from January 1, 2008 to Dezember 31, 2016.

Three different models have been chosen to create the single forecasts:

1. *Asymmetric Multiplicative Error Model* (AMEM) (Engle, 2002; Engle and Gallo, 2006), which for a basic (1,1) order has the following structure:

$$\begin{aligned}rv_t &= \mu_t \xi_t \\ \mu_t &= \omega + \alpha_1 rv_{t-1} + \beta_1 \mu_{t-1} + \gamma D_{t-1} rv_{t-1}\end{aligned}\quad (1)$$

with $\omega > 0$, $\alpha_1 \geq 0$, $\beta_1 \geq 0$, $\alpha_1 + \beta_1 < 1$. D_t as a dummy variable that takes the value of 1 if the $r_t < 0$ and 0 otherwise;

2. *Asymmetric Power Multiplicative Error Model* (APMEM), which for the usual (1,1) order is given by:

$$\begin{aligned}rv_t &= \mu_t \xi_t \\ \mu_t^\delta &= \omega + \alpha_1 rv_{t-1}^\delta + \beta_1 \mu_{t-1}^\delta + \gamma D_{t-1} rv_{t-1}^\delta\end{aligned}\quad (2)$$

with $\omega > 0$, $\alpha_1 \geq 0$, $\beta_1 \geq 0$, $\alpha_1 + \beta_1 < 1$, $\delta > 0$. This model is a generalization of the basic MEM and is strictly related to the Asymmetric Power ACD model (cf., Fernandes and Grammig, 2006);

3. *Asymmetric Heterogeneous AutoRegressive Model* (AHAR), that is the HAR model with a leverage effect term (Corsi, 2009):

$$rv_t = c + \beta^{(d)} rv_{t-1} + \beta^{(w)} rv_{t-1}^{(w)} + \beta^{(m)} rv_{t-1}^{(m)} + \gamma D_{t-1} rv_{t-1} + \epsilon_t^{(d)}\quad (3)$$

where (d) stands for the time horizons of one day, $[rv_{t-1}^{(w)}]$ is the weekly realized volatility, which at time t is given by the average:

$$rv_t^{(w)} = \frac{1}{5} \left(rv_t^{(d)} + rv_{t-1d}^{(d)} + \dots + rv_{t-4d}^{(d)} \right)\quad (4)$$

and $rv_{t-1}^{(m)}$ is the monthly realized volatility which at time t is given by the average

$$rv_t^{(m)} = \frac{1}{22} \left(rv_t^{(d)} + rv_{t-1d}^{(d)} + \dots + rv_{t-21d}^{(d)} \right)\quad (5)$$

As a preliminary analysis, in Figure 1 we compare the forecasts obtained using the three models mentioned above for the year 2016 (colored lines) with the actual values of the volatility (dashed black line) for the DAX 5-minute series. The chart shows that all models react satisfactorily to positive peaks of volatility, whereas they are not able to achieve a suitable degree of accuracy when volatility reaches a local minimum. This issue, which is common also to the other observed time series, can be overcome by combining the forecasts of two

models, as we will see later.

The combined methods are based on the following two combination models:

- *comb1* model, based on a simple unconstrained ordinary least squares estimation of the weights. The one-step-ahead forecast is given by

$$rv_T(1) = \alpha + \beta_1 f_T^{(1)}(1) + \beta_2 f_T^{(2)}(1) \quad (6)$$

with $f_T^{(1)}(1)$ and $f_T^{(2)}(1)$ denoting, respectively, the first and second model forecasts;

- *comb2* model, with the combination given by

$$rv_T(1) = \alpha + (\beta_1 + \delta_1 D_{t-1}) f_T^{(1)}(1) + (\beta_2 + \delta_2 D_{t-1}) f_T^{(2)}(1) \quad (7)$$

which includes a dummy variable, D_t , that takes the value 1 if rv_t is lower than rv_{t-1} and 0 otherwise. The ratio of this choice is given by considering that, as we have mentioned before, the forecast of volatility is often far from the actual realized volatility while the latter is decreasing.

1.1 Loss Functions

To compare the results of the combined schemes with those that can be obtained by exclusively relying on a single model, we have computed three loss functions:

1. *Mean Square Error*(MSE), given by

$$MSE = \frac{\sum_{i=1}^n (rv_{T+i} - rv_{T+i-1}(1))^2}{n} \quad (8)$$

with rv_{T+i} being the observed value of the realized volatility and $rv_{T+i-1}(1)$ as the one step ahead forecast for time $T + i$, $i = 1, \dots, n$;

2. *Quasi-Likelihood*(QLIKE), defined as

$$QLIKE = \frac{1}{n} \sum_{i=1}^n \left[\frac{rv_{T+i}}{rv_{T+i-1}(1)} - \ln \left(\frac{rv_{T+i}}{rv_{T+i-1}(1)} \right) - 1 \right] ; \quad (9)$$

3. A new measure called *Asymmetric MSE*(AMSE), given by

$$AMSE = \frac{1}{n} \sum_{i=1}^n \left(1 + \left(\frac{\epsilon_{T+i}^2}{rv_{T+i}} \right)^m I(\epsilon_{T+i} > 0) \right) \epsilon_{T+i}^2 \quad (10)$$

where $\epsilon_{T+i} = rv_{T+i} - rv_{T+i-1}(1)$. This measure is an extension of the MSE: each term of the sum reduces to ϵ_{T+i}^2 when the indicator function is 0 (overestimation of the volatility) and is given by $\left(1 + \left(\frac{\epsilon_{T+i}^2}{rv_{T+i}}\right)^m\right) \epsilon_{T+i}^2$ when the indicator function is 1 (underestimation of the volatility).

We decided to build up a new loss function for the evaluation of forecasts for two reasons. First, as we have already said, AMSE is more suitable than MSE when it comes to assessing forecasts that underestimate the volatility as it penalizes underestimation to a greater extent. Second, it can be shown that AMSE is able to perform more reasonably than QLIKE, one of the most widely used loss functions in the volatility forecasting literature (Patton, 2011).

Figure 2 displays a graphic comparison of QLIKE with MSE and two versions of its asymmetric modification, the first with power term $m = 1$ and the second with $m = 2$. On the x -axis we have depicted the relative deviations of the forecasts from the true value (which amounts to 2 in this case), whereas on the y -axis we have represented the relative difference of the loss functions between the cases of underestimation and overestimation of the same size. As expected, MSE appears as a flat line because it is a symmetric loss function. In contrast, QLIKE and AMSE start to rise almost immediately, particularly QLIKE which, as evidenced by the sharp slope of the red line, is able to reach very high values. However, the AMSE loss function appears distinctly smoother than QLIKE (especially when $m = 1$), indicating that AMSE is well-balanced and also more regular than the QLIKE loss function.

For computing the forecast combinations, we start by splitting the data into an estimation and training set and a test set. The former is again split into two parts, the first being used to estimate the parameters of the model, the second (the training period) to estimate the weights to be attributed to the single forecasts. The test set is used to evaluate the different models. We have chosen to take into account two different training periods in our analysis: a four-year training period and a three-year training period. For instance, with a four-year training period, we estimate the parameters of the models using observations from January 2, 2008 to December 31, 2011, then compute one step ahead forecasts from January 2, 2012 to December 31, 2015; these forecasts are used to estimate the weights of the combinations; finally, the one step ahead forecast for January 2, 2016 is produced. Then, we estimate the parameters of the models using observations from January 3, 2008 to January 2, 2012, compute one step ahead forecasts from January 3, 2012 to January 2, 2016 to estimate the weights of the combinations, and the one step ahead forecast for January 3, 2016 is produced.

2 Comparisons among forecasting models

In this section we will show the results of our analysis. For each model we display the values of the three loss functions mentioned above for both the forecasts and the observed values.

Because two of the three single models we have used (AMEM and APMEM) are very similar to each other, we present first a comparison between AMEM and AHAR, then between APMEM and AHAR, along with the combined schemes we described in Section 1.

2.1 AMEM vs. AHAR

The order of the two single models is defined using the Ljung-Box test on the residuals of the in-sample analysis of the two models. We have selected an AMEM (1,1) for the DAX dataset, an AMEM (1,2) for CAC and AEX, and an AHAR with a second lag term (rv_{t-2}) for all datasets.

Table 1 shows the results for the first comparison, i.e., AMEM (1,1) and AHAR models, along with combined forecasts, using DAX data. We can see that the *comb2* model performs very well for almost all indicators; only QLIKE prefers the AMEM (1,1) model.

The findings provided in Table 2 for the CAC dataset are very similar to those for the DAX dataset. Indeed, there are only two differences: QLIKE prefers *comb2* for *rv* kernel with a training period of four years instead of AMEM (1,2); and AMSE with $m = 1$ prefers the AMEM model for *rv* 5 minutes with a training period of three years, instead of *comb2*.

The results for the AEX dataset (Table 3) are not so different from the others. The *comb2* model predominates, but the AMEM (1,2) also performs well, particularly according to QLIKE (in three cases out of four).

2.2 APMEM vs. AHAR

In this subsection we assess if a generalization of the AMEM basic model is able to improve the accuracy of the combined forecasts. According to the Ljung-Box test, we use an APMEM (1,1) for DAX and an APMEM (1,2) for CAC and AEX.

As shown in Table 4, we have observed an actual improvement in the combined forecasts. Compared to the findings shown in Table 1, in 14 cases out of 16 the loss functions appoint the smallest value to a combination. Even according to QLIKE, *comb2* is preferred over APMEM half the time.

As we expected, the improvement that occurred for the DAX dataset moving from AMEM to APMEM holds for CAC as well, even if it is less significant. Indeed, the results shown in Table 5 are almost the same as those that we see in Table 2 in terms of loss functions choices. However, this time AMSE ($m = 1$) selects the *comb2* model for all volatility measures and training periods.

Observing the Table 6, we can gladly see that, compared to the Table 3, the transition from AMEM to APMEM has caused a consistency of the loss functions. Almost all loss functions

suggest choosing *comb2*. The single model APMEM (1,2) has only been chosen with the use of the by QLIKE statistic. Overall, these are the same findings that are seen in Table 5.

2.3 Accuracy of Forecasts

So far we have evaluated the available forecasts by means of the numerical values provided by the loss functions. Before doing so, however, we need to assess if the forecast series are different from a statistical point of view. To this end, we have used a conditional predictive ability (CPA) test of Giacomini and White (2006) to make pair-wise comparisons among all forecasting models ($\alpha = 0.05$). The null hypothesis is that the two models under comparison have the same predictive accuracy. Because *comb2* has proved to be the best combination scheme in most cases, we tested if it is more accurate than the other models as our alternative hypothesis.

Tables 7-9 provide the findings of the analysis according to the three datasets. At first glance, we observe some similarities among the comparisons. In more detail, we find that the two combination schemes, *comb1* and *comb2*, show the same equal conditional predictive ability for all models, market indexes, types of realized volatility, or the training periods, except for kernel estimates with a training period of four years using the CAC dataset. Regarding the other comparisons, the alternative hypothesis was rejected for *comb2* and AMEM twice in DAX, once in CAC, and three times out of four in AEX. The same applies for *comb2* and APMEM. Finally, AEX data depict an equal forecast accuracy also between *comb2* and AHAR when *rv* 5 minutes with a three-year training period is involved. In all other cases, the CPA test provides evidence that *comb2* has a better predictive ability than the other models. Tables 7-9 especially make it clear that *comb2* outperforms AHAR in almost all situations, except for the AEX case mentioned before.

3 Conclusions

In this paper, we demonstrate that an improvement in the accuracy of forecasts of a measure of volatility, namely realized volatility, can be achieved by combining predictions originating from several models. We forecast the daily realized volatility one step ahead for a one-year period with three single models (AMEM, APMEM, AHAR) and two combinations (*comb1*, *comb2*). Subsequently, we compare predicted values and actual data using a number of loss functions. We find that combining the AHAR model with APMEM instead of AMEM causes an enhancement in the quality of the forecasts computed using combination schemes, especially the *comb2* model, which proves to be the best model in most situations. This finding holds for the DAX, AEX, and (to a lesser extent) CAC datasets, and for all training

periods. However, after carrying out a CPA test to assess the forecasting accuracy, we found that the *comb2* model, in almost all cases, was as accurate as *comb1* and, in half the cases, as accurate as AMEM and APMEM. Furthermore, *comb2* outperforms the AHAR model in almost all situations, allowing us to reject the null hypothesis of equal unconditional predictive ability.

References

- Ané, T. (2006). An analysis of the flexibility of asymmetric power GARCH models. *Computational Statistics and Data Analysis*, 51(2):1293–1311.
- Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2):174–196.
- Engle, R. (2002). Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business and Economic Statistics*, 20(3):339–350.
- Engle, R. and Gallo, G. (2006). A multiple indicators model for volatility using intra-daily data. *Journal of Econometrics*, 131(1):3–27.
- Fernandes, M. and Grammig, J. (2006). A family of autoregressive conditional duration models. *Journal of Econometrics*, 130(1):1–23.
- Giacomini, R. and White, H. (2006). Tests of conditional predictive ability. *Econometrica*, 74(6):1545–1578.
- Oxford-Man Institute of Quantitative Finance (2017). Realized Library. <http://realized.oxford-man.ox.ac.uk/data>.
- Patton, A. (2011). Volatility forecast comparison using imperfect volatility proxies. *Journal of Econometrics*, 160(1):246–256.
- Raviv, E. (2016). Forecast combinations in R using the ForecastCombinations package. A Manual.
- Timmermann, A. (2006). Forecast combinations. *Handbook of Economic Forecasting*, 1:135–196.

Figure 1: Comparison among observed realized volatility (5 minutes) for year 2016 and AMEM (1,1), APMEM (1,1), and AHAR forecasts – DAX dataset.

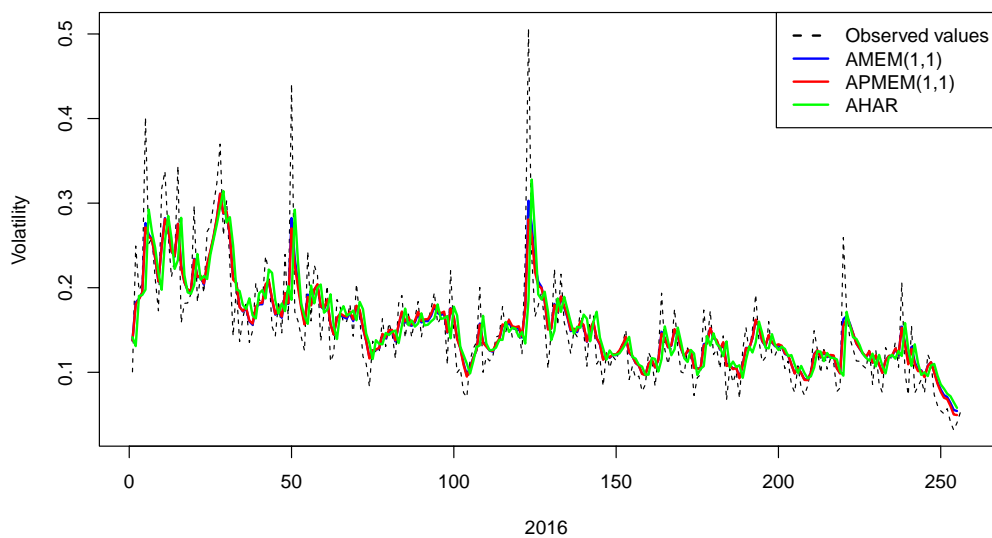


Figure 2: Comparison of MSE, QLIKE, AMSE (m=1,2) loss functions computed on a series h of evenly spaced forecasts from 0 to 2.

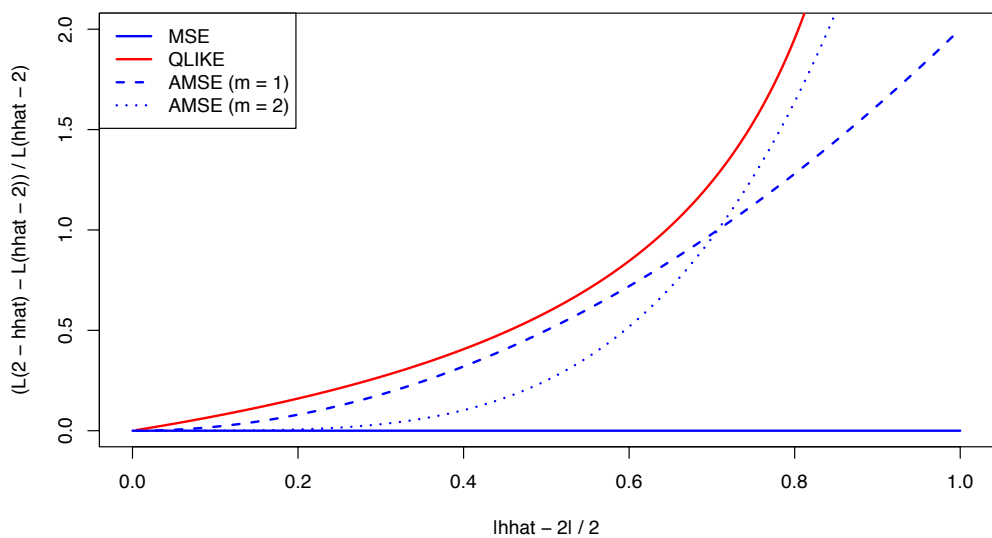


Table 1: Comparison of AMEM (1,1), AHAR, and combination schemes (smallest values in bold) – DAX dataset.

		MSE				QLIKE			
Series	Training period	AMEM (1,1)	AHAR	comb1	comb2	AMEM (1,1)	AHAR	comb1	comb2
rv 5min	4 years	0.254	0.293	0.252	0.248	4.231	4.979	4.285	4.246
	3 years	0.254	0.293	0.254	0.250	4.231	4.979	4.393	4.338
rv kernel	4 years	0.206	0.251	0.204	0.200	3.505	4.312	3.552	3.508
	3 years	0.206	0.251	0.206	0.203	3.505	4.312	3.639	3.594
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	AMEM (1,1)	AHAR	comb1	comb2	AMEM (1,1)	AHAR	comb1	comb2
rv 5min	4 years	0.271	0.316	0.271	0.268	0.257	0.298	0.255	0.251
	3 years	0.271	0.316	0.273	0.270	0.257	0.298	0.257	0.254
rv kernel	4 years	0.217	0.267	0.216	0.212	0.207	0.254	0.206	0.201
	3 years	0.217	0.267	0.218	0.215	0.207	0.254	0.208	0.204

Table 2: Comparison of AMEM (1,2), AHAR, and combination schemes (smallest values in bold) – CAC dataset.

		MSE				QLIKE			
Series	Training period	AMEM (1,2)	AHAR	comb1	comb2	AMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.243	0.292	0.241	0.239	3.809	4.660	3.845	3.836
	3 years	0.243	0.292	0.243	0.242	3.809	4.660	3.884	3.891
rv kernel	4 years	0.240	0.295	0.238	0.233	3.855	4.810	3.889	3.834
	3 years	0.240	0.295	0.239	0.236	3.855	4.810	3.925	3.895
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	AMEM (1,2)	AHAR	comb1	comb2	AMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.268	0.325	0.269	0.267	0.249	0.301	0.248	0.246
	3 years	0.268	0.325	0.270	0.270	0.249	0.301	0.249	0.249
rv kernel	4 years	0.264	0.325	0.264	0.260	0.245	0.303	0.244	0.240
	3 years	0.264	0.325	0.265	0.263	0.245	0.303	0.245	0.242

Table 3: Comparison among AMEM(1,2), AHAR, and combination schemes (smallest values in bold) – AEX dataset.

		MSE				QLIKE			
Series	Training period	AMEM (1,2)	AHAR	comb1	comb2	AMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.251	0.293	0.250	0.247	3.925	4.706	3.969	3.960
	3 years	0.251	0.293	0.251	0.251	3.925	4.706	4.007	4.023
rv kernel	4 years	0.219	0.257	0.218	0.213	3.854	4.677	3.898	3.834
	3 years	0.219	0.257	0.219	0.216	3.854	4.677	3.933	3.886
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	AMEM (1,2)	AHAR	comb1	comb2	AMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.288	0.343	0.289	0.287	0.263	0.313	0.263	0.261
	3 years	0.288	0.343	0.290	0.291	0.263	0.313	0.264	0.265
rv kernel	4 years	0.241	0.286	0.241	0.237	0.224	0.266	0.224	0.218
	3 years	0.241	0.286	0.242	0.239	0.224	0.266	0.224	0.221

Table 4: Comparison among APMEM (1,1), AHAR, and combination schemes (smallest values in bold) – DAX dataset.

		MSE				QLIKE			
Series	Training period	APMEM (1,1)	AHAR	comb1	comb2	APMEM (1,1)	AHAR	comb1	comb2
rv 5min	4 years	0.249	0.293	0.248	0.245	4.165	4.979	4.177	4.156
	3 years	0.249	0.293	0.250	0.247	4.165	4.979	4.267	4.232
rv kernel	4 years	0.203	0.251	0.202	0.198	3.464	4.312	3.469	3.429
	3 years	0.203	0.251	0.203	0.200	3.464	4.312	3.546	3.506
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	APMEM (1,1)	AHAR	comb1	comb2	APMEM (1,1)	AHAR	comb1	comb2
rv 5min	4 years	0.267	0.316	0.266	0.265	0.252	0.298	0.251	0.249
	3 years	0.267	0.316	0.268	0.266	0.252	0.298	0.253	0.251
rv kernel	4 years	0.214	0.267	0.213	0.210	0.205	0.254	0.203	0.200
	3 years	0.214	0.267	0.215	0.213	0.205	0.254	0.205	0.202

Table 5: Comparison among APMEM (1,2), AHAR, and combination schemes (smallest values in bold) – CAC dataset.

		MSE				QLIKE			
Series	Training period	APMEM (1,2)	AHAR	comb1	comb2	APMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.246	0.292	0.243	0.240	3.833	4.660	3.880	3.855
	3 years	0.246	0.292	0.245	0.242	3.833	4.660	3.931	3.913
rv kernel	4 years	0.241	0.295	0.239	0.234	3.865	4.810	3.903	3.840
	3 years	0.241	0.295	0.241	0.236	3.865	4.810	3.948	3.903
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	APMEM (1,2)	AHAR	comb1	comb2	APMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.272	0.325	0.271	0.268	0.252	0.301	0.250	0.247
	3 years	0.272	0.325	0.273	0.271	0.252	0.301	0.252	0.249
rv kernel	4 years	0.266	0.325	0.265	0.261	0.247	0.303	0.245	0.240
	3 years	0.266	0.325	0.267	0.263	0.247	0.303	0.247	0.243

Table 6: Comparison among APMEM (1,2), AHAR, and combination schemes (smallest values in bold) – AEX dataset.

		MSE				QLIKE			
Series	Training period	APMEM (1,2)	AHAR	comb1	comb2	APMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.256	0.293	0.252	0.247	3.946	4.706	4.001	3.975
	3 years	0.256	0.293	0.253	0.250	3.946	4.706	4.049	4.039
rv kernel	4 years	0.219	0.257	0.218	0.213	3.855	4.677	3.902	3.831
	3 years	0.219	0.257	0.219	0.215	3.855	4.677	3.942	3.886
		AMSE ($m = 1$)				AMSE ($m = 2$)			
Series	Training period	APMEM (1,2)	AHAR	comb1	comb2	APMEM (1,2)	AHAR	comb1	comb2
rv 5min	4 years	0.292	0.343	0.293	0.288	0.267	0.313	0.266	0.261
	3 years	0.292	0.343	0.294	0.292	0.267	0.313	0.267	0.264
rv kernel	4 years	0.241	0.286	0.242	0.237	0.224	0.266	0.224	0.219
	3 years	0.241	0.286	0.242	0.239	0.224	0.266	0.225	0.221

Table 7: Percentage of superiority in forecasting accuracy of *comb2* using CPA test (pair-wise comparisons against AMEM, APMEM, AHAR and *comb1*) – DAX dataset.

f_1	f_2	DAX rv 5 min. 4 years t.p.	DAX rv 5 min. 3 years t.p.	DAX rv kernel 4 years t.p.	DAX rv kernel 3 years t.p.
comb2	AMEM	0.016	0.178	0.007	0.129
comb2	AHAR	0.004	0.015	0.002	0.007
comb2	comb1	0.119	0.170	0.076	0.127
comb2	APMEM	0.015	0.156	0.005	0.096
comb2	AHAR	0.003	0.009	0.002	0.005
comb2	comb1	0.130	0.193	0.076	0.113

Table 8: Percentage of superiority in forecasting accuracy of *comb2* using CPA test (pair-wise comparisons against AMEM, APMEM, AHAR and *comb1*) – CAC dataset.

f_1	f_2	CAC rv 5 min. 4 years t.p.	CAC rv 5 min. 3 years t.p.	CAC rv kernel 4 years t.p.	CAC rv kernel 3 years t.p.
comb2	AMEM	0.036	0.142	0.006	0.034
comb2	AHAR	0.000	0.001	0.000	0.000
comb2	comb1	0.187	0.397	0.046	0.161
comb2	APMEM	0.044	0.107	0.008	0.036
comb2	AHAR	0.000	0.001	0.000	0.000
comb2	comb1	0.132	0.212	0.041	0.120

Table 9: Percentage of superiority in forecasting accuracy of *comb2* using CPA test (pair-wise comparisons against AMEM, APMEM, AHAR and *comb1*) – AEX dataset.

f_1	f_2	AEX rv 5 min. 4 years t.p.	AEX rv 5 min. 3 years t.p.	AEX rv kernel 4 years t.p.	AEX rv kernel 3 years t.p.
comb2	AMEM	0.074	0.538	0.024	0.153
comb2	AHAR	0.035	0.076	0.007	0.017
comb2	comb1	0.217	0.764	0.065	0.104
comb2	APMEM	0.051	0.233	0.024	0.135
comb2	AHAR	0.024	0.052	0.006	0.014
comb2	comb1	0.171	0.484	0.078	0.144