Journal of Econometrics 253 (2026) 106123

Contents lists available at ScienceDirect

Journal of Econometrics

journal homepage: www.elsevier.com/locate/jeconom

Large-scale model comparison with fast model confidence sets

Sylvain Barde

School of Economics, Politics and International Relations, University of Kent, Sibson Building, Canterbury, CT2 7FS, UK

ARTICLE INFO

JEL classification:
C12
C18
C52
C55

Keywords:
Model selection
Model confidence set
Bootstrapped statistics

ABSTRACT

The paper proposes a new algorithm for finding the confidence set of a collection of forecasts or prediction models. Existing numerical implementations use an elimination approach, where one starts with the full collection of models and successively eliminates the worst performing until the null of equal predictive ability is no longer rejected at a given confidence level. The intuition behind the proposed implementation lies in reversing the process, i.e. starting with a collection of two models and updating both the model rankings and p-values as models are successively added to the collection. The first benefit of this approach is a reduction of one polynomial order in both the time complexity and memory cost of finding the confidence set of a collection of M models using the R rule, falling respectively from $\mathcal{O}(M^3)$ to $\mathcal{O}(M^2)$ and from $\mathcal{O}(M^2)$ to $\mathcal{O}(M)$. The second key benefit is that it allows for further models to be added at a later point in time, thus enabling collaborative efforts using the model confidence set procedure. The paper proves that this implementation is equivalent to the elimination approach, demonstrates the improved performance on a multivariate GARCH collection consisting of 4800 models, and discusses possible use-cases where this improved performance could prove useful.

1. Introduction and literature review

A common problem when several models, forecasts or prediction methods are available for the same phenomenon of interest is the reliable identification of those that systematically perform better than the rest. As pointed out by Corradi and Swanson (2013), Diebold and Mariano (1995) offer a starting point via a test for the null hypothesis that two prediction models have equal accuracy. However, naïvely extending this test to larger collections of predictors is complicated by the fact that as the number of prediction models in the collection increases, the probability of accidentally selecting a model with no real predictive power increases, a problem known as data snooping. The Reality Check (RC) test of White (2000), which sequentially tests a collection of prediction models against a benchmark model in order to test the null that no model in the collection outperforms the benchmark, addresses this problem through the use of a bootstrap implementation, thus avoiding this data snooping bias. This in turn was improved by Hansen (2005) with the test for Superior Predictive Accuracy (SPA), which offers better protection against the inclusion of irrelevant models into the collection.

A further generalisation is proposed by Hansen et al. (2011), in the form of the Model Confidence Set (MCS), which unlike the RC and SPA tests does not require specifying an *a priori* benchmark, therefore providing greater flexibility. Given a collection of model losses the MCS procedure sequentially tests the null hypothesis that all models in the collection have equal predictive power on the data and eliminates the worst performing model if the hypothesis is rejected. This testing process continues until the null hypothesis can no longer be rejected on the worst-performing model. Several versions of the procedure exist, depending on the equivalence rule used to identify candidate models to be eliminated, however in all cases the surviving subset of models forms the MCS.

E-mail address: s.barde@kent.ac.uk.

https://doi.org/10.1016/j.jeconom.2025.106123
Received 22 July 2024; Received in revised form 20 June 2025; Accepted 25 October 2025

Available online 14 November 2025

The MCS has become a popular method of assessing the relative accuracy of forecasts and prediction models, particularly following the initial release in 2014 of the R package of Bernardi and Catania (2018), leading to a volume of research too large to present in detail.[1] Initial applications were in volatility forecasting, for example Patton et al. (2009), Laurent et al. (2012), Boudt et al. (2013), Iltuzer and Tas (2013), Neumann and Skiadopoulos (2013), Wilhelmsson (2013), Amado and Teräsvirta (2014), Hamid (2014), Caporin and McAleer (2014), Hansen et al. (2014), Liu et al. (2015). More recent applications of the MCS to volatility forecasts include Ma et al. (2019), Audrino et al. (2020), Conrad and Kleen (2020), with Dumitrescu et al. (2022), Masini et al. (2023) providing recent examples of model comparison of machine-learning forecasting models.

Use of the MCS has extended beyond forecasts of stock market volatility, and finds wide application in several other areas. A significant and early area of application is forecasts of crude oil prices and futures (for example Kristjanpoller and Minutolo, 2016; Degiannakis and Filis, 2017; Wei et al., 2017; Baumeister et al., 2022). More recently, the MCS has found use in energy and environmental forecasting, such as Huang et al. (2021), Liu et al. (2021), Liang et al. (2022) for carbon pricing forecasts, Bessec and Fouquau (2018), Rendon-Sanchez and de Menezes (2019) for electricity load forecasts, or Newell et al. (2021) for GDP-based forecasts of temperature trends. Finally, cryptocurrency volatility is a recent and fast-growing field of application. Corbet et al. (2019) provide a recent literature review, with Caporale and Zekokh (2019), Walther et al. (2019), Ma et al. (2020) providing notable applications, and Alonso-Monsalve et al. (2020), Akyildirim et al. (2021) again comparing forecasts generated with machine-learning methods.

Despite this popularity, the MCS approach suffers from two drawbacks, pointed out by Ferrari and Yang (2015, p. 3): the MCS "is meant to handle only a fixed and small number of models to begin with". The first is the high computational requirement and poor scaling characteristics of the approach, which limit the size of the collections that can be examined in practice. The second is that the MCS approach loses one of the attractive features of the RC test, which is the ability to perform incremental testing. As explained by White (2000, p. 1110), a collaborative effort can be carried out by posting the bootstrap indices, the values of the RC test statistic and bootstrapped test statistics, allowing "researchers at different locations or at different times to further understanding of the phenomenon modelled without needing to know the specifications tested by their collaborators or competitors". The iterative elimination approach used by the MCS procedure precludes adding further models *ex post*, and thus this type of collaborative and incremental testing.

A review of the 200 most cited papers using the MCS shows indications that both drawbacks are present.[2] Regarding the first, only a small number of cases examine collections larger than 100 models. In the field of stock market volatility forecasts, Laurent et al. (2012) compare 125 multivariate specifications while Liu et al. (2015) examine a collection of 625 different realised measures of asset price volatility. The relatively more recent work of Newell et al. (2021) manages 800 specifications, while the largest MCS application to date is the 1176-strong collection examined by Caporale and Zekokh (2019). The latter three large collections are the by-product of the curse of dimensionality, as they systematically combine multiple design dimensions for the models involved. This could easily have produced much larger collections, as is the case in the empirical application below, yet these sizes are close to the computationally feasible maximum for a desktop computer of corresponding vintage.

The average collection size in the 200 studies considered is much lower, at 38.64 models, with a median of 11. In the large majority of cases, this is because the focus of the study is not a sweeping review, instead the authors propose a novel forecasting methodology, model or predictor, which is then tested against a set of standard literature benchmarks. This results in a large number of small studies, each showing that their proposed approach improves on the benchmarks, but with little information on relative performance. Even if such designs are reasonable given their research questions, it is clear that it would be desirable to use a methodology that allows researchers to compare their findings to the current state of the art, in the interest of better understanding the determinants of improved predictive performance across an entire field. The cost of coordinating such an effort has fallen since the initial proposal of White (2000), due to the emergence of code repositories and related online platforms. A notable example of this is the www.kaggle.com platform, which serves as a machine learning repository and hosts prediction competitions with thousands of submissions.[3]

This paper contributes an alternative algorithmic approach for obtaining the MCS, based on the range (R) elimination rule, which addresses both problems: it provides both a dramatic improvement in performance on large collections while allowing the possibility of updating the MCS *ex post* with further models. The intuition underpinning the approach is that rather than starting with the full collection of models and shrinking it down to the subset of models that form the MCS, the collection is initially made up of two models and MCS is gradually updated as models are added to the collection. In this manner, only deviations of the new model with respect to the existing collection need to be calculated and stored, rather than the full set of pairwise deviations, thus reducing the computational complexity of the R-rule MCS procedure by one polynomial order. Similarly, growing the collection of models rather than shrinking it intuitively allows for further models to be added to the collection at a later point in time.

The remainder of the paper is organised as follows: Section 2 describes the existing R-rule elimination algorithm for the MCS and identifies the conditions required for vector updating to be feasible. The updating algorithm is presented in Section 3, which also establishes its theoretical equivalence with the elimination algorithm. Section 4 presents the results of the Monte Carlo exercise carried out to confirm the theoretical results, while Section 5 presents a large-scale application to multivariate GARCH forecasting models of stock volatility. Finally, Section 6 discusses the main use cases where the fast updating algorithm would be beneficial.

[1] As of June 2025, the original Hansen et al. (2011) article has been cited 1408 times according to Web of Science's relatively conservative metric, and 2577 times according to Google Scholar's wider measure.

[2] The list of papers reviewed, their area of study and the collection size is provided in the supplementary material.

[3] A survey of 660 completed Kaggle competitions, dated April 5 2025 and provided in the supplementary material, revealed 265 with a collection size larger than 1000 models, with the largest containing 8751 models.

2. The R-rule elimination MCS

2.1. Notation and implementation

In order to be able to discuss the properties of the proposed MCS updating algorithm, it is important to first present the existing MCS R-rule elimination implementation used as its basis. The notation broadly follows Hansen et al. (2011), with a few modifications added further below. With N empirical observations and a collection $\mathcal{M}$ containing M models, the MCS procedure requires an $N \times M$ set of losses L to calculate the relative loss $d_{i,j,n}$ of model i relative to model j.[4]

$$d_{i,j,n} \equiv L_{n,i} - L_{n,j} \tag{1}$$

The true, unobserved, mean pairwise loss deviation is $\mu_{i,j}$, its variance $\sigma^2_{i,j}$, and the sample mean pairwise deviation observed in the loss data L are defined respectively as:

$$\mu_{i,j} \equiv E_n\left[d_{i,j,n}\right], \qquad \sigma^2_{i,j} \equiv \mathrm{Var}_n\left(d_{i,j,n}\right), \qquad \bar{d}_{i,j} \equiv \frac{\sum_n d_{i,j,n}}{N} \tag{2}$$

Hansen et al. (2011) define the set of superior models as the subset $\mathcal{M}^* \subseteq \mathcal{M}$ of models that offer at least equivalent performance relative to other models in the collection:

$$\mathcal{M}^* \equiv \{i \in \mathcal{M} : \quad \mu_{i,j} \leq 0, \quad \forall\, j \in \mathcal{M}\} \tag{3}$$

The collection $\mathcal{M}$ can be partitioned into the set of superior models (3) and its complement $\mathcal{M}^\epsilon$, which contains the models eliminated from the original collection:

$$\mathcal{M}^\epsilon = \mathcal{M} \setminus \mathcal{M}^* \tag{4}$$

The elimination MCS aims to identify $\mathcal{M}^*$ by successively removing the model that exhibits statistically significant worse performance. The procedure uses the empirical average deviations (2) to calculate the following set of t-statistics, under the null hypothesis that all pairwise deviations are zero-valued, i.e. $H_0 : \mu_{i,j} = 0, \forall\, i, j$.

$$t_{i,j} = \frac{\bar{d}_{i,j}}{\sqrt{\mathrm{Var}_b\left(\delta_{i,j,b}\right)}} \tag{5}$$

The standard error of $\bar{d}_{i,j}$ is estimated using a bootstrap, where a set of $N \times B$ bootstrap indices $\mathcal{B}$ allows us to generate an $N \times M \times B$ array of resampled loss matrices $\mathcal{L}$. This is used to calculate resampled pairwise deviations:

$$\delta_{i,j,b} = \frac{\sum_n \left(\mathcal{L}_{n,i,b} - \mathcal{L}_{n,j,b}\right)}{N} \tag{6}$$

The R rule uses the following elimination rule and equivalence test statistic in each iteration k, identifying the model generating the largest pairwise t-statistic $t_{i,j}$, i.e. the one with the highest average normalised loss relative to other models in the collection.

$$e_k = \arg\max_{i \in \mathcal{M}} \max_{j \in \mathcal{M}} \left(t_{i,j}\right), \qquad T_{e_k} = \max_{i \in \mathcal{M}} \max_{j \in \mathcal{M}} |t_{i,j}| \tag{7}$$

Algorithm 1 R-rule elimination MCS $(L, \mathcal{B})$

Require: L: $N \times M$ matrix of losses
Require: $\mathcal{B}$: $N \times B$ matrix of bootstrap indexes

1: $t \leftarrow$ Calculate matrix of t-statistics with Eq. (5)
2: $\tau \leftarrow$ Calculate matrices of bootstrapped statistics with Eq. (9)
3: **for** $k = 0 \rightarrow |\mathcal{M}|$ **do**
4: $\quad e_k, T_1 \leftarrow$ Find worst model with elimination rule Eq. (7)
5: $\quad \mathcal{T}_{k,b} \leftarrow$ Find bootstrapped statistics Eq. (10)
6: $\quad P_k \leftarrow$ Calculate bootstrapped p-value Eq. (10)
7: $\quad$ Remove row/column e_k from t and τ
8: **end for**
return e, T, P

For completeness, Hansen et al. (2011) also propose an alternative elimination rule, referred to as the *max* rule, where the t-statistics, candidate model and elimination statistics are given by:

$$t_i = \frac{\bar{d}_i}{\sqrt{\mathrm{Var}_b\left(\delta_{i,b}\right)}}, \qquad e_k = \arg\max_{i \in \mathcal{M}} \left(t_i\right), \qquad T_{e_k} = \max_{i \in \mathcal{M}} |t_i| \tag{8}$$

[4] The choice of loss function to use is important for model selection, and the effect of that choice on the MCS procedure is discussed in Laurent et al. (2013). In particular the MCS does not natively penalise overfitting, implying this needs to be handled through the choice of loss function. Here we follow Hansen et al. (2011) and simply assume that a suitable set of losses is available for the model predictions.

Here $\bar{d}_i = N^{-1}\sum_n L_{n,i} - \bar{L}$ measures the average loss of model i relative to the average loss across all models remaining in the collection, $\bar{L} = (MN)^{-1}\sum_n \sum_{i\in\mathcal{M}} L_{n,i}$. While identifying the candidate for elimination using *max* rule (8) requires fewer comparisons than with the R rule (7), the reason why the latter is nevertheless preferred for the fast MCS implementation is clarified in Section 2.3.

The distribution of the elimination statistic (7), which is used to test H_0 against the alternative hypothesis $H_A : \mu_{i,j} \neq 0$, is obtained using bootstrapped pairwise deviations:

$$\tau_{i,j,b} = \frac{\delta_{i,j,b} - \bar{d}_{i,j}}{\sqrt{\mathrm{Var}_b\left(\delta_{i,j,b}\right)}} \tag{9}$$

which are used to generate the bootstrapped elimination statistics and associated bootstrapped p-values, where $I(\dots)$ is the Boolean indicator function:

$$\mathcal{T}_{e_k,b} = \max_{i\in\mathcal{M}} \max_{j\in\mathcal{M}} |\tau_{i,j,b}|, \qquad P_{e_k} = \max\left(P_{e_{k-1}},\ \frac{1}{B}\sum_b I\left(\mathcal{T}_{e_k,b} \geq T_{e_k}\right)\right) \tag{10}$$

After T_{e_k} and P_{e_k} have been obtained, model e_k is removed from the collection and the next candidate for elimination can be examined. This iterative elimination implementation is summarised in algorithm 1. Once all models have been processed, given a choice of significance α one can obtain the MCS $\widehat{\mathcal{M}}^*_{1-\alpha} = \left\{i \in \mathcal{M} : P_i < 1-\alpha\right\}$. The main properties of $\widehat{\mathcal{M}}^*_{1-\alpha}$ depend on the distributional properties of the losses L used in algorithm 1. Hansen et al. (2011) make the following assumption in this regard:

Assumption 1 (*Hansen et al., 2011 - Stationarity of Relative Losses*)**.** For some $r > 2$ and $\gamma > 0$, it holds that $E|d_{i,j,n}|^{r+\gamma} < \infty$, $\forall\, i,j \in \mathcal{M}$ and that $\{d_{i,j,n}\}_{i,j\in\mathcal{M}}$ is strictly stationary with $\mathrm{Var}_n(d_{i,j,n}) > 0$ and α-mixing of order $-r/(r-2)$.

Given this assumption, Hansen et al. (2011) prove the following convergence results:

$$\begin{cases} \liminf_{N\to\infty} \Pr\left(\mathcal{M}^* \subset \widehat{\mathcal{M}}^*_{1-\alpha}\right) \geq 1-\alpha & \text{(Theorem 1)} \\ \lim_{N\to\infty} \Pr\left(e_k \notin \mathcal{M}^* \mid H_A\right) = 1 & \text{(Theorem 4)} \end{cases} \tag{11}$$

The implication of these are that $\widehat{\mathcal{M}}^*_{1-\alpha}$ statistically recovers the set of superior models $\mathcal{M}^*$ with the significance level α directly controlling the type I error rate, i.e. the probability that superior models are excluded from the MCS. Of particular importance for later is the fact that algorithm 1 will almost surely eliminate all the inferior models $\mathcal{M}^{\epsilon}$ before it moves on to examining models from the superior set $\mathcal{M}^*$.

2.2. Conditions for vector updating

Some additional notation and definitions are required to formalise the concept of vector updating of model rankings. The elimination sequence produced by algorithm 1 naturally orders the models from largest T_k to smallest T_k. When updating a collection $\mathcal{M}$ with additional models, this natural connection between algorithm iterations and model ordering is lost. Therefore, ε_m is used to index the location of an arbitrary model m in the elimination sequence, i.e. if $\varepsilon_m = k$ then $e_k = m$.[5] In addition we denote $\mathcal{E}^+_m \equiv \{i \in \mathcal{M} : \varepsilon_i > \varepsilon_m\}$ as the set of models eliminated after m (i.e. ranked better) and $\mathcal{E}^-_m \equiv \{i \in \mathcal{M} : \varepsilon_i < \varepsilon_m\}$ as the set of models eliminated before m, so that by construction, $\mathcal{M} = \mathcal{E}^+_m \cup m \cup \mathcal{E}^-_m$. Finally, m^+ is the model eliminated just after m, i.e. $\varepsilon_{m^+} = \varepsilon_m + 1$, while m^- is the model eliminated just before m, i.e. $\varepsilon_{m^-} = \varepsilon_m - 1$. The formalisation for the updating of a collection with an extra model is defined as follows:

Definition 1 (*Collection Updating*)**.** Given $\mathcal{M}$, a collection of $M > 1$ models, let $\mathcal{M}' = \mathcal{M}\backslash\{m\}$ be the collection obtained by removing an arbitrary model $m \in \mathcal{M}$. t is the $M\times M$ matrix of pairwise t-statistics (5) for $\mathcal{M}$ and t' is the equivalent $(M-1)\times(M-1)$ matrix for $\mathcal{M}'$. Similarly, T, e and ε respectively denote the vector of elimination statistics, the elimination sequence and the model rankings obtained by running the elimination algorithm 1 on $\mathcal{M}$, and T', e' and ε' are the corresponding vectors obtained by running the same procedure on $\mathcal{M}'$. The elimination sets are denoted as $\mathcal{E}^+_m$ and $\mathcal{E}^-_m$ for $\mathcal{M}$, and $\mathcal{E}'^+_m$ and $\mathcal{E}'^-_m$ for $\mathcal{M}'$. Note that by construction, $\mathcal{M}' = \mathcal{E}^+_m \cup \mathcal{E}^-_m$. Finally, $\mathcal{T}, \mathcal{T}'$ are the bootstrapped distributions (9) obtained for $\mathcal{M}$ and $\mathcal{M}'$ respectively given a common set of bootstrap indices $\mathcal{B}$.

Conditions 2.1–2.3 provide requirements for vector updating of the test statistic (7) and bootstrapped statistics (10) to be feasible in this context.

Condition 2 (*Conditions for Vector Updating*)**.** *Given the setting and notation specified in Definition 1, we set the following conditions:*

- **2.1** $\quad \arg\max_i(t_{m,i}) \in \mathcal{E}^+_m \qquad i \in \mathcal{M}'$
- **2.2** $\quad \arg\max_i(t_{k,i}) = \arg\max_j(t'_{k,j}) \qquad i \in \mathcal{E}^+_k \setminus \{m\},\ j \in \mathcal{E}'^+_k,\ k \in \mathcal{E}^-_m$
- **2.3** $\quad \mathcal{E}^+_k = \mathcal{E}'^+_k \cup \{m\} \qquad \forall\ k \in \mathcal{E}^-_m$

[5] For example $\varepsilon_m = 2$ indicates that model m is the second to be eliminated.

Condition 2.1 states that the model i which produces the largest t-statistic for m, referred to as the eliminator of m, is ranked better than m itself. Condition 2.2 ensures that for existing models k ranked worse than m, their eliminator in $\mathcal{E}_k^{'+}$ is still the eliminator in $\mathcal{E}_k^+ \setminus \{m\}$. Condition 2.3 is much stronger than 2.2, in that it requires that for every model k ranked worse than m, the new set of preferred models $\mathcal{E}_k^+$ is unchanged from $\mathcal{E}_k^{'+}$, apart from the addition of m.[6]

These conditions formalise an important circularity inherent to the switch from elimination to updating: in order to verify that 2.1–2.3 are satisfied before updating collection $\mathcal{M}'$, one would need to already have the rankings for collection $\mathcal{M}$. Intuitively, one would expect the eliminator of a given model $m \in \mathcal{M}$ to be a better-ranked model. However, the motivation behind the need for the MCS procedure is precisely that it is not always possible to reliably rank models when the data is not informative enough, therefore the conditions will not be satisfied in general. This issue is addressed in Section 3, which will show that a collection $\mathcal{M}$ possesses an internal structure ensuring they are almost surely satisfied for all models in $\mathcal{M}^\epsilon$.

Given Definition 1, Lemma 1 establishes that when conditions 2.1 and 2.2 are satisfied, the R-rule equivalence statistics (7) for collection $\mathcal{M}$ can be obtained by updating the corresponding statistics from $\mathcal{M}'$ using only the vector of pairwise t-statistics $t_{m,i}$.

Lemma 1 (*Vector Updating of Equivalence Statistics*)**.** *Given the setting and notation specified in Definition 1, if condition 2.1 is satisfied then the following updating equations hold:*

1.1 The equivalence statistic of the additional model T_m is given by:

$$T_m = \max_i \left(t_{m,i}\right) \qquad i \in \mathcal{M}' \tag{12}$$

1.2 Given the value of T_m obtained with (12), the elimination sets can be recovered using:

$$\begin{cases} \mathcal{E}_m^+ = \{k \in \mathcal{M}' : T_k' < T_m\} \\ \mathcal{E}_m^- = \mathcal{M}' \setminus \mathcal{E}_m^+ \end{cases} \tag{13}$$

1.3 The updated equivalence statistics $T_k \ \forall \ k \in \mathcal{E}_m^+$ are simply:

$$T_k = T_k' \tag{14}$$

In addition, if condition 2.2 is satisfied for a given $k \in \mathcal{E}_m^-$, then its corresponding equivalence statistic T_k can be updated by:

$$T_k = \max\left(T_k', t_{k,m}\right) \tag{15}$$

Proof. This is provided in Appendix A.

Lemma 2 below states that a similar set of updating rules relate the bootstrapped equivalence statistics in collection $\mathcal{M}$ to those in the smaller collection $\mathcal{M}'$.

Lemma 2 (*Vector Updating of Bootstrapped Statistics*)**.** *Given the setting and notation specified in Definition 1, if condition 2.1 is satisfied, then the following vector updating equations hold $\forall$ b:*

2.1 The bootstrapped equivalence statistics of the additional model $\mathcal{T}_{m,b}$ are given by:

$$\mathcal{T}_{m,b} = \max\left(\mathcal{T}'_{m^+,b}, \max_i |\tau_{m,i,b}|\right) \qquad i \in \mathcal{E}_m^+ \tag{16}$$

2.2 The updated bootstrapped equivalence statistics $\mathcal{T}_{k,b} \ \forall k \in \mathcal{E}_m^+$ are simply:

$$\mathcal{T}_{k,b} = \mathcal{T}'_{k,b} \tag{17}$$

In addition, if condition 2.3 is satisfied for $k \in \mathcal{E}_m^-$, then $\forall\, b$ the corresponding bootstrapped equivalence statistics $\mathcal{T}_{k,b}$ can be updated by:

$$\mathcal{T}_{k,b} = \max\left(\mathcal{T}'_{k,b}, \max_i |\tau_{m,i,b}|\right) \qquad i \in \mathcal{E}_k^+ \tag{18}$$

Proof. This is provided in Appendix A.

While the two lemmas are similar, a key difference is that Lemma 2 requires the strong condition 2.3 be satisfied, while Lemma 1 only requires the weaker condition 2.2. This is because the elimination rule (7) always removes the row/column of the matrix of equivalence statistics containing largest $t_{i,j}$ value, ensuring that the sequence of elimination statistics T_k is strictly decreasing over iterations. This is not the case for the bootstrapped matrices, as the rows/columns removed in each iteration do not necessarily contain the largest value. As a result, the sequence of equivalence statistics in each bootstrapped matrix is only non-increasing with iterations, therefore updating them based on the decomposability of the max(.) function requires a stronger condition.

[6] Condition 2.3 is stronger in the sense that if it is satisfied, then so is condition 2.2, however the reverse is not true.

2.3. Computational complexity and relation to other confidence set implementations

Each iteration k of algorithm 1 requires finding the largest entry in the $(M-k+1)\times(M-k+1)$ matrices of t and τ_b. The total number of operations required is therefore proportional to the square pyramidal number $\sum_{k=1}^{M} k^2$, implying a run-time complexity of $\mathcal{O}(M^3)$. The t-statistics (5) and B bootstrapped t-statistics (9) each require a $M\times M$ matrix, leading to memory scaling as $\mathcal{O}(M^2)$. If Lemmas 1 and 2 are used to iteratively update the collection, then the time complexity becomes proportional to the triangular number $\sum_{k=1}^{M} k$ and is reduced by an order magnitude to $\mathcal{O}(M^2)$. Storing M-length vectors rather than matrices similarly reduces the memory requirement to $\mathcal{O}(M)$.

The reason why the R rule is best suited for such a vector updating implementation requires some clarification, as the *max* rule (8) might initially seem better suited: the equivalence test statistic is already the maximum of a vector, and the fact that the average model loss is measured relative to the average loss of the collection $\bar{L}$ seemingly removes the circularity apparent in Lemmas 1 and 2 above, as by construction the eliminator $\bar{L}$ will always be smaller than the largest individual model loss.

The computational challenge when updating a collection $\mathcal{M}'$ with a model m is finding the location where it enters the rankings, in other words identifying the elimination sets $\mathcal{E}_m^+$ and $\mathcal{E}_m^-$, which is done by finding its elimination statistic T_m. Given that the model m is fixed, this requires searching for the correct eliminator. Under the *max* rule, the eliminator is the average loss $\bar{L}$ over the set of preferred models $\mathcal{E}_m^+$. The circularity mentioned above is therefore still present: correctly calculating the T_m value requires already knowing the elimination set $\mathcal{E}_m^+$. In fact, the circularity problem is computationally worse, as the space of candidate eliminators $\bar{L}$ that needs to be searched grows exponentially with collection size M. While it may be feasible to find vector updating rules that can achieve this, it cannot be done with Lemmas 1 and 2. Under the R rule by contrast, the eliminator in (5) is another model in the collection, therefore given m, identifying the T_m statistic only requires a vector evaluation across all current models $i \in \mathcal{M}'$.

This link between the ease of identification of the eliminator and the computational complexity of the resulting updating algorithm also applies to the VSCS proposed by Ferrari and Yang (2015). The key assumption in their framework is that models are linear, with Gaussian errors, and contain a subset of p available regressors:

$$Y_n = \beta_0 + \sum_{i=1}^{p} \beta_i X_{n,i} + \varepsilon_n, \quad \forall\, n = 1, \dots, N, \qquad \varepsilon_n \underset{i.i.d}{\sim} \mathcal{N}(0, \sigma^2) \tag{19}$$

Within this framework, models are identified using a p-length Boolean vector γ indicating which regressors are included in the model. Letting γ_f be the index of the 'full' model,[7] a model γ can be eliminated from the VSCS using the following F-test:

$$\hat{F}_{\gamma_f,\gamma} = \frac{RSS_\gamma - RSS_{\gamma_f}}{RSS_{\gamma_f}} \frac{df_{\gamma_f}}{df_\gamma - df_{\gamma_f}} \quad > \quad F_{df_\gamma - df_{\gamma_f}, df_{\gamma_f}}(\alpha) \tag{20}$$

The elimination statistic $\hat{F}_{\gamma_f,\gamma}$ is a penalised relative loss between the two models, similar to (5), where given the linear model and Gaussian errors assumptions, γ_f is by construction the eliminator for all other models γ. This results in a computationally efficient updating framework, as obtaining $\hat{F}_{\gamma_f,\gamma}$ when a new γ is added to the collection requires a single operation. The VSCS therefore scales as $\mathcal{O}(M)$, with the added computational benefit that the use of an F-test removes the need for bootstrapped p-values.

The VSCS approach, however, suffers from two problems that limit its applicability. The first is that while it scales linearly with the number of candidate specifications M, given p regressors there are 2^p possible specifications. As p increases, processing the full collection induced by those regressors rapidly becomes unfeasible, despite the computational efficiency on a single model γ. The second is the loss of generality involved in assuming a linear model (19). The MCS, by contrast, can handle any model losses that satisfy assumption 1, as illustrated by the literature reviewed in Section 1.

3. A fast updating implementation for the R elimination rule MCS

Lemmas 1 and 2 establish that vector-based updating of collections is possible, as long as $\forall\, m \in \mathcal{M}$, the eliminator i is always located in the set of preferred models $\mathcal{E}_m^+$ (condition 2.1), and that this set is stable as extra models are added to the collection, either in a weak manner, under condition 2.2, or more strongly under condition 2.3.

We now show that the definition of the set of superior objects (3) directly induces an ordering of the models which spans the full collection $\mathcal{M}$ and ensures that conditions 2.1 and 2.2 are satisfied in a manner that enables reliable vector updating.[8]

3.1. Ordering of models under the R elimination rule

In order to show this we first define a recursive partitioning of $\mathcal{M}$. This is similar to the bucketing procedure suggested by Hurlin et al. (2017), except that the recursion uses the set of superior models $\mathcal{M}_0^*$ rather than the MCS $\widehat{\mathcal{M}}_{1-\alpha}^*$ to partition $\mathcal{M}$.

[7] If $p < N$, the full model can straightforwardly obtained via OLS, when $p > N$ Ferrari and Yang (2015) propose the use of screening methods to obtain γ_f.
[8] It does not satisfy condition 2.3, however, and Section 3.2 provides a fall-back heuristic for this case.

Definition 2 (*Recursive Partitioning of Superior Models*)**.** Given a collection of models $\mathcal{M}_0$, let $\{\mathcal{M}_0^*, \mathcal{M}_1\}$ denote the partition (4) of $\mathcal{M}_0$ into the set of superior models and its complement (3). Subsequent partitions $\{\mathcal{M}_k^*, \mathcal{M}_{k+1}\}$ are obtained by recursively applying (4) as follows:

$$\begin{cases} \mathcal{M}_{k+1} = \mathcal{M}_k \setminus \mathcal{M}_k^*, & k \in \{0, 1, \dots, K-2\} \\ \mathcal{M}_{K-1} = \mathcal{M}_{K-1}^* \end{cases} \tag{21}$$

The recursive partitioning in definition 2 allows us to prove the following lemma relating to the existence of an ordering of the models in any given collection $\mathcal{M}_0$.

Lemma 3 (*Strict Weak Ordering of Models*)**.** *Applying definition 2 to a collection $\mathcal{M}_0$ creates an ordered partition $\mathcal{P} = \{\mathcal{M}_0^*, \mathcal{M}_1^*, \dots, \mathcal{M}_{K-1}^*\}$ with order relation $\succ$:*

$$\mathcal{M}_0^* \succ \mathcal{M}_1^* \succ \dots \succ \mathcal{M}_{K-1}^* \tag{22}$$

where each $\mathcal{M}_k^*$ is an equivalence class, i.e. $\forall\, k$:

$$i \not\succ j, \quad j \not\succ i \qquad \forall\, i, j \in \mathcal{M}_k^* \tag{23}$$

This induces a strict weak ordering on the models in $\mathcal{M}_0$, where given $b > a$:

$$i \succ j \qquad \forall\, i \in \mathcal{M}_a^*, \ \forall\, j \in \mathcal{M}_b^* \tag{24}$$

Proof. This is provided in Appendix A.

The ordered partition $\mathcal{P}$ and strict weak ordering of all models $m \in \mathcal{M}_0$ formalises the relationship between the two intuitions mentioned above: we expect a model to be outperformed by models in a higher-ranked equivalence class, but we also expect not to be able to distinguish models within the same equivalence class. The existence of this strict weak ordering leads to the following corollary[9]:

Corollary 3.1 (*Strict Dominance of Superior Models*)**.** *Let $\mathcal{M}_0$ be a collection of models partitioned into the set of superior models $\mathcal{M}_0^*$ and its complement, $\mathcal{M}_1$. Given definition 2, a model $i \in \mathcal{M}_0^*$ if and only if it satisfies $\mu_{j,i} > \mu_{j,k}$ $\forall\, j \in \mathcal{M}_0$, $\forall\, k \in \mathcal{M}_1$.*

Proof. This is provided in Appendix A.

Corollary 3.1 provides an alternative but equivalent definition of $\mathcal{M}_0^*$ as the subset of $\mathcal{M}_0$ with respect to which the largest $\mu_{i,j}$ values are obtained:

$$\mathcal{M}_0^* \equiv \{i \in \mathcal{M}_0 : \ \mu_{j,i} \geq \mu_{j,k} \quad \forall\, j, k \in \mathcal{M}_0\} \tag{25}$$

Lemma 3 and Corollary 3.1 provide some theoretical insights on how the MCS algorithm is able to achieve such strong results (11) using only a simple pairwise test. In light of Lemma 3, the definition of the set of superior models (3) can be interpreted as the combination of two conditions: $\mathcal{M}_0^*$ is the subset of models $i, j \in \mathcal{M}_0$ for which the null hypothesis $H_0 : \mu_{i,j} = 0$ holds and for which $\mu_{k,j} > 0$, $\forall\, k \in \mathcal{M}_0 \setminus \mathcal{M}_0^*$. The elimination algorithm 1 only tests for H_0, and because there are $1 \leq K \leq M$ equivalence classes $\mathcal{M}_k^* \in \mathcal{P}$ within which H_0 holds, testing for H_0 clearly cannot by itself identify $\mathcal{M}_0^*$. This is ensured instead by the iterative elimination of any model k for which there is evidence of a positive $\mu_{k,j}$. Crucially, it is the strict weak ordering (24) induced by the ordered partition $\mathcal{P}$ that enables the methodology to get around the multiple hypothesis problem mentioned in the introduction. Specifically, (24) ensures that once we fail to reject H_0 for a given pair of models i and j, we have evidence that it holds for all models in $\widehat{\mathcal{M}}_0^*$ and that $\mu_{k,j} > 0$, $\forall\, k \in \mathcal{M}_0 \setminus \widehat{\mathcal{M}}_0^*$, despite never explicitly testing for any of these additional pairwise statements.

The practical implication of Lemma 3 and Corollary 3.1, outlined in Corollary 3.2, is that given a realisation of $N \times M$ losses L that satisfy assumption 1 for a collection $\mathcal{M}_0$, the largest t-statistic (5) for all models $i \in \mathcal{M}_0$ will almost surely be obtained with respect to models in the superior set $\mathcal{M}_0^*$.

Corollary 3.2 (*Stochastic Dominance of Superior Models*)**.** *Given definition 2, if assumption 1 holds for the $N \times M$ losses L, we have $\forall\, i \in \mathcal{M}_0^*$, $\forall\, j \in \mathcal{M}_0$ and $\forall\, k \in \mathcal{M}_1$:*

$$\lim_{N \to \infty} \Pr\left(t_{j,i} > t_{j,k}\right) = 1 \tag{26}$$

and similarly, $\forall\, i, j \in \mathcal{M}_0$:

$$\lim_{N \to \infty} \Pr\left(\arg\max_i (t_{j,i}) \in \mathcal{M}_0^*\right) = 1 \tag{27}$$

[9] Given the existence of a set of equivalence classes $\mathcal{M}_k^*$, we now explicitly use the notation of definition 2 rather than that of the original partition (4). In particular, we now use $\mathcal{M}_0^*$ rather than $\mathcal{M}^*$ to denote the set of superior objects (3).

This corollary, specifically Eq. (27), is what provides guarantees that the conditions required for the vector updating Lemmas 1 and 2 are almost surely satisfied in practice. Specifically, when adding m to $\mathcal{M}'$, if $m \notin \mathcal{M}_0^*$, we have $\mathcal{M}_0^* \subset \mathcal{E}_m^+$, ensuring that conditions 2.1 and 2.2 hold almost surely as $N \to \infty$. This is not the case, however, when $m \in \mathcal{M}_0^*$ as in this case $\mathcal{M}_0^* \cap \mathcal{E}_m^- \neq \emptyset$. Similarly, this result does not provide any guarantee that condition 2.3 holds in general. Nevertheless, as shown below, these guarantees are sufficient to ensure that an updating algorithm using Lemmas 1 and 2 can provide equivalent output to the elimination algorithm 1 as $N \to \infty$.

Eqs. (26) and (27) converge at different rates, which has important consequences for the algorithm's practical performance on small samples. Eq. (37), which underpins the proof of Corollary 3.2, establishes that $\forall\, j \in \mathcal{M}_0$, $\forall\, i \in \mathcal{M}_0^*$ and $\forall\, k \in \mathcal{M}_1$, the pairwise difference $t_{j,i} - t_{j,k}$ diverges to infinity at rate $\sqrt{N}$, directly providing the convergence rate for (26). Eq. (27), however, relates to the distribution of $\max_i(t_{j,i}) - \max_k(t_{j,k})$, which is an order statistic. As pointed out by Hansen et al. (2011), this depends on several nuisance parameters, which motivates the choice of bootstrap methods.[10] Intuitively however, if $|\mathcal{M}_0^*| > 1$ then as N increases, for a given model j there will be multiple $t_{j,i}$ values diverging from the other $t_{j,k}$ values, each at rate $\sqrt{N}$. The probability that the largest $t_{j,i}$ statistic comes from one of these multiple candidates will therefore converge to one at a faster rate than the base $\sqrt{N}$ rate of individual deviations (26). This suggests that conditions 2.1 and 2.2 can hold even for relatively small values of N, which we explicitly test in Section 4.

Algorithm 2 2-Pass fastMCS $(L, \mathcal{B})$

Require: L: $N \times M$ matrix of losses
Require: $\mathcal{B}$: $N \times B$ matrix of bootstrap indexes

Pass 1: Get model rankings
for $m \in \mathcal{M}$ **do**
  $t_m \leftarrow$ Calculate vector of t-statistics with Eq. (5)
  $T \leftarrow$ Update model rankings with Eqs. (12)–(15)
end for
$e \leftarrow$ Sort models by ascending value of T

Pass 2: Get p-values
for $i = 0 \to |\mathcal{M}|$ **do**
  $m \leftarrow e_i$
  $\tau_{m,b} \leftarrow$ Calculate vectors of bootstrapped statistics with Eq. (9)
  $\mathcal{T}_{m,b} \leftarrow$ Obtain bootstrapped statistics with Eq. (16)
  $P_m \leftarrow$ Calculate bootstrapped p-value with Eq. (10)
end for
return e, T, P

3.2. Equivalence of a vector-based two-pass fast MCS implementation

The vector updating Lemmas 1 and 2 can be combined to provide a fast MCS implementation, outlined in algorithm 2, which processes the collection $\mathcal{M}_0$ in two separate passes, the first to generate the elimination sequence (7) and the second to generate the corresponding p-values (10). By relying on the model rankings obtained in the first pass, the bootstrapped distributions (10) can be updated from the best model to the worst, avoiding the problem of condition 2.3 not holding in general.

We now prove the following proposition, which states that given identical inputs, the MCS algorithms 1 and 2 almost surely provide equivalent outputs.

Proposition 1 (*Equivalence of Fast and Elimination Algorithms*). *Let $\mathcal{M}_0$ be a collection of M models and let ϵ_m, P_m be the elimination index and MCS p-values produced $\forall\, m \in \mathcal{M}_0$ by the R-rule elimination algorithm 1 on an $N \times M$ loss matrix L. Similarly, let ϵ_m^f, P_m^f be the equivalent values produced by the fast algorithm 2 on the same losses L. If the losses L satisfy assumption 1, then as $N \to \infty$ we have $\forall\, m \in \mathcal{M}_1$:*

$$\Pr\left(\epsilon_m^f = \epsilon_m\right) \to 1, \qquad \Pr\left(P_m^f = P_m\right) \to 1$$

Proof. This is provided in Appendix A.

Proposition 1 applies only for models $m \in \mathcal{M}_1$. In practice, algorithms 1 and 2 almost surely produce the same elimination sequence and p-values for the inferior set $\mathcal{M}_1$, but they might rank models differently in the superior set $\mathcal{M}_0^*$. If $m \in \mathcal{M}_1$, Corollary

[10] In particular, given the setup implied by definition 2, one would need to know the number of equivalence classes K in $\mathcal{P}$, the cardinality of each equivalence class $|\mathcal{M}_k^*|$ and the expected pairwise losses $\mu_{i,j}$ between classes.

3.2 provides a strong guarantee that conditions 2.1 and 2.2 are satisfied, however, if $m \in \mathcal{M}_0^*$, then $\mathcal{M}_0^* \nsubseteq \mathcal{E}_m^+$ and we lose that guarantee. This ties in with the general intuition of Section 3.1 that ranking models within a given equivalence class is meaningless, and what matters in practice is being able to separate inferior and superior models.

3.3. Heuristic for a single-pass fast MCS implementation

While the fast algorithm 2 scales more favourably than the elimination algorithm 1, it requires two passes on the loss data L to get around the problem of condition 2.3 not holding in general, therefore it retains one of the latter's drawbacks: if new models are added to a previously processed collection, it is not possible to update the original MCS without re-running the procedure on the larger collection. The following definition provides an updating heuristic for the bootstrapped t-statistic which enables the MCS to updated with further models.

Definition 3 (*Updating Heuristic for Bootstrapped Statistics*). $\forall\, k \in \mathcal{E}_m^-$ where condition 2.3 is not satisfied, applying the updating Eqs. (16) and (18) produces two values $\forall\, b$:

$$\begin{cases} \mathcal{T}_{k,b}^L = \max\left(\mathcal{T}'_{k^+,b}, \max\limits_i |\tau_{m,i,b}|\right) & i \in \mathcal{E}_k^+ \\ \mathcal{T}_{k,b}^U = \max\left(\mathcal{T}'_{k,b}, \mathcal{T}_{k,b}^L\right) \end{cases} \tag{28}$$

where by construction $\mathcal{T}_{k,b}^L \leq \mathcal{T}_{k,b}^U$. The midpoint of the two values provides the following updating heuristic:

$$\mathcal{T}_{k,b} \approx \frac{\mathcal{T}_{k,b}^L + \mathcal{T}_{k,b}^U}{2} \tag{29}$$

The lower bound in (28) corresponds to (16) and treats model k as if it were a new addition to the collection, by discarding the existing (and potentially incorrect) value of $\mathcal{T}'_{k,b}$. Because this information is not included, it will tend to produce a lower value that can be used to bound the true but unknown $\mathcal{T}_{k,b}$ from below. Replacing $\mathcal{T}_{k,b}^L$ into the upper bound of (28) yields $\mathcal{T}_{k,b}^U = \max\left(\mathcal{T}'_{k,b}, \mathcal{T}'_{k^+,b}, \max_i |\tau_{m,i,b}|\right)$. When combined with the implicit constraint from the definition of the bootstrapped elimination statistics (10) that $\mathcal{T}'_{k,b} \geq \mathcal{T}'_{k^+,b}$, one recovers the updating Eq. (18) Unlike the $\mathcal{T}_{k,b}^L$ bound, $\mathcal{T}_{k,b}^U$ includes the potentially incorrect value of $\mathcal{T}'_{k,b}$, thus providing the heuristic upper bound.

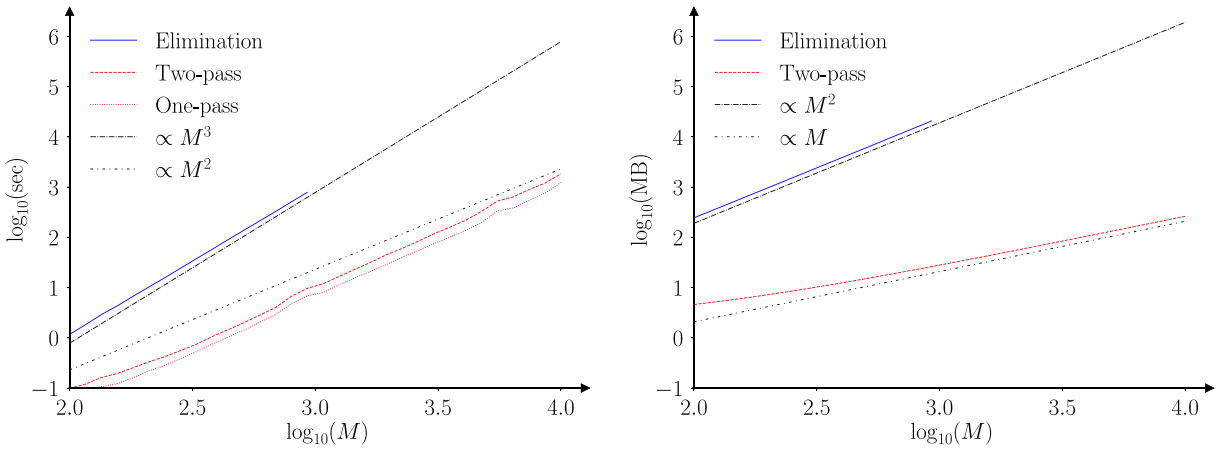
The use of the heuristic (29) enables the equivalence statistics (5) and bootstrapped statistics (10) to be processed in the same iteration, resulting in the single-pass algorithm 3. In practice, once a model m is added to the collection and rankings have been updated using (12), the algorithm checks if condition 2.3 is satisfied and if so uses rule (18) to update the bootstrapped statistics. If the condition is not satisfied, the heuristic (29) is used instead. In addition, in order to minimise the number of times the heuristic is used, the algorithm sorts the models by average loss as a preliminary step, so that better performing models are processed early. This is because they are more likely to be in $\mathcal{M}_0^*$ and therefore cause shifts in the rankings of other models.

Algorithm 3 1-Pass fastMCS $(L, \mathcal{B})$

Require: L: $N \times M$ matrix of losses
Require: $\mathcal{B}$: $N \times B$ matrix of bootstrap indexes

```
L' ← sort L in order of increasing average loss Eq. (2)
for m ∈ M do
    t_m ← Calculate vector of t-statistics with Eq. (5)
    T ← Update model rankings with Eqs. (12)–(15)
    τ_{m,b} ← Calculate vectors of bootstrapped statistics with Eq. (9)
    T_{m,b} ← Obtain bootstrapped statistics with Eq. (16)
    for k ∈ E_m^- do
        if E_k^+ = E'^+_k ∪ {m} then
            T_{k,b} ← Update bootstrapped statistic with Eq. (18)
        else
            T_{k,b}^L, T_{k,b}^U ← Get heuristic bounds with Eq. (28)
            T_{k,b} ← Update bootstrapped statistics with Eq. (29)
        end if
    end for
end for
P ← Calculate bootstrapped p-values with Eq. (10)
return e, T, P
```

Because algorithm 3 updates the equivalence statistics (5) in the same manner as the two-pass algorithm 2, the two algorithms will provide identical elimination sequences and the $\Pr\left(\varepsilon_m^f = \varepsilon_m\right) \to 1$ result of Proposition 1 carries over. The limitation of the one-pass algorithm 3 is that the heuristic only approximates $\mathcal{T}_{k,b}$ when condition 2.3 is not satisfied, therefore Proposition 1 guarantee that the p-values (10) almost surely correspond to those of the elimination algorithm 1 no longer holds.



(a) Time complexities (b) Memory requirements

Fig. 1. MCS implementation benchmarking, $N = 250$.

4. Monte Carlo simulations

Two Monte Carlo (MC) exercises were carried out to check that the updating algorithms 2 and 3 perform as expected.[11] Both use the same data-generating process (DGP) as Hansen et al. (2011), where the synthetic $N \times M$ matrix of losses L is given by:

$$L_{n,i} = \theta_i + \frac{a_n}{\sqrt{E\left[a_n^2\right]}} X_{n,i} \tag{30}$$

With:

$$\begin{cases} \theta_i = \frac{\lambda}{\sqrt{N}} \left(0, \frac{1}{M-1}, \dots, \frac{M-2}{M-1}, 1\right) \\ a_n = \exp\left(y_n\right) \\ y_n = -\frac{\varphi}{2(1+\varphi)} + \varphi y_{n-1} + \sqrt{\varphi}\epsilon_n \end{cases} \tag{31}$$

The two stochastic elements are $\epsilon_n \underset{iid}{\sim} \mathcal{N}(0,1)$ and $X_{n,i} \underset{iid}{\sim} \mathcal{N}_M(0,\Sigma)$, where the diagonal elements of the variance covariance matrix Σ are equal to one, with off-diagonal elements set to ρ. Three free parameters control the characteristics of the simulated losses: λ controls the dispersion of the mean loss θ_i, making models easier or harder to distinguish, ρ controls the degree of correlation in losses across models, and finally φ allows for conditional heteroskedasticity, i.e. GARCH effects, in the losses. The ranges used for the parameters are the same as Hansen et al. (2011), in that we draw $\lambda \in [5, 40]$, $\rho \in [0, 0.95]$ and $\varphi \in [0, 0.8]$. All MC exercises use a block bootstrap with $B = 1000$ resamples of blocksize $l = 2$, and columns of each simulated loss matrix L were randomly shuffled to ensure that models were not ordered in a systematic manner.

4.1. Performance benchmarking

The first MC exercise aims to evaluate algorithms 2 and 3 in terms of their computational performance and accuracy relative to the elimination algorithm 1. This consists of 200 MC replications, each recording the running time, memory requirements, model rankings and p-values of algorithms 1–3, for 32 values of M such that $\log_{10}(M)$ is evenly spaced in the $[2, 4]$ interval. For each of the resulting 6400 specifications, the values of λ, ρ and ϕ are drawn from a 3-dimensional Sobol sequence, ensuring the algorithms are tested on the widest possible range of parameter combinations for the synthetic DGP (30)–(31). Two versions were run, one with $N = 250$ and another with $N = 30$, in order to test the equivalence result of Proposition 1 on small samples.

Fig. 1 shows the running time and memory requirement over the range of collection sizes M, in log units and averaged over the 200 MC replications. As expected, the time complexity of the elimination algorithm 1 scales as $\mathcal{O}\left(M^3\right)$, with memory requirements scaling as $\mathcal{O}\left(M^2\right)$. In fact, as is visible in Fig. 1, for the elimination algorithm 1 the MC exercise had to be stopped at $M = 928$, at which point the memory requirement was 20.1 GB.[12] Fig. 1 also confirms that both updating algorithms 2 and 3 have $\mathcal{O}\left(M^2\right)$ time complexity and $\mathcal{O}(M)$ memory requirement.[13]

[11] The python toolbox containing the fast MCS implementation can be downloaded from https://github.com/Sylvain-Barde/fastMCS. The numerical work for Sections 4 and 5 was carried out on a 36-worker High-Performance Computing node consisting of two 18-core Intel Xeon 6150 CPUs running at 2.70 GHz, with 512 GB DDR4 of 2666 MB/s RAM, and all replication material can be downloaded from https://github.com/Sylvain-Barde/mArch_mcs_analysis.

Table 1
Monte Carlo comparison of Two-pass fast MCS algorithm performance.

Diagnostics, Two-pass versus:	$N = 30$		$N = 250$	
	Elim.	One-pass	Elim.	One-pass
Number of replications compared	3200	6400	3200	6400
MCS size diagnostics				
% Replications with MCS size differences	0	14.531	0	7.172
Mean absolute difference in MCS size	–	2.267	–	2.516
% relative to collection size	–	0.110	–	0.094
Median absolute difference in MCS size	–	1	–	2
% relative to collection size	–	0.051	–	0.054
Model ranking diagnostics				
% Replications with ranking differences	0	0	0	0
% Replications with T stat. differences	0	0	0	0
p-value diagnostics				
% Replications with p-value differences	0	93.016	0	88.609
Mean % share of models affected	–	7.676	–	6.343
Mean absolute difference in affected P-values	–	0.002	–	0.002

Note: Absolute difference diagnostics are calculated for affected models only. For the MCS size diagnostics, both mean and median sizes are reported as the distribution of size differences is highly skewed: its lower bound, 1, is also its mode, occurring in 51.61% of cases for $N = 30$ and 48.37% of cases for $N = 250$.

Table 1 provides the diagnostics for the MCS comparison of algorithm 2 (two-pass) against algorithms 1 (elimination) and 3 (one-pass) respectively. The first key finding is that in all 3200 cases, both for $N = 250$ and $N = 30$, the two-pass algorithm provides the same model rankings and p-values as the elimination algorithm. Given that $|\mathcal{M}_0^*| = 1$ in (30), this not only confirms the theoretical result of Proposition 1, which guarantees equivalent results for models within $\mathcal{M}_1$, but illustrates that this equivalence holds even in small samples. A similar comparison of the two-pass and one-pass heuristic algorithms does reveal differences in the MCS sizes, again confirming the predictions of Section 3. This difference is small, however: while the size of the MCS differs in 7.2% of replications in the $N = 250$ case, the mean absolute deviation in MCS size is only 0.1% of the original collection size M. The results also confirm that this difference does not come from the model rankings, which are identical to those produced by the two-pass algorithm, but instead from the bootstrapped p-values (10): 88.6% of replications show p-value differences, within which an average of 6.3% of p-values are affected. This confirms that condition 2.3 does not hold in general, but also that the heuristic (29) is able to mitigate the impact of this on affected p-values, as shown by the low absolute difference of 0.002.

4.2. Large-scale power and size analysis

The second MC analysis takes advantage of the computational gains of the fast MCS implementation to extend the size and power analysis of Hansen et al. (2011) to larger collections. The aim is to evaluate the behaviour of type I errors (superior models from $\mathcal{M}^*$ being excluded from the MCS) or type II errors (inferior models from $\mathcal{M}^\epsilon$ included in MCS) as collection size M increases, for a fixed number of observations $N = 250$. The MC design uses the DGP (30)–(31) and for comparability is identical to that used in Hansen et al. (2011), with one exception. In all cases the size of the set of superior models is increased to $|\mathcal{M}^*| = 10$, up from the $|\mathcal{M}^*| = 1$ value implied by (31). This is done to allow for the possibility of type I errors, as corollary 1 of Hansen et al. (2011) shows that $\liminf_{N\to\infty} \Pr\left(\mathcal{M}^* \subset \widehat{\mathcal{M}}^*_{1-\alpha}\right) = 1$ when $|\mathcal{M}^*| = 1$, as opposed to the more general result in (11) when $|\mathcal{M}^*| > 1$. For a collection size M this is achieved by generating $M - 9$ sets of losses using (31), and another 9 where a value $\theta_i = 0$ is imposed.

Table 2 presents the results of the analysis for collection sizes of 500, 1000 and 2000.[14] The size results show that the type I error rate always remain below α, confirming Theorem 1 of Hansen et al. (2011) in (11). The raw power results are also in line with Hansen et al. (2011), in that keeping the other DGP parameters equal, the absolute size of the MCS increases with collection size M. Correctly interpreting this result, however, requires taking into account the fact that the DGP (31) bounds the average loss θ_i of the worst-performing model, implying that increases in M translate to increases in the density of models within a fixed $[0, \lambda/\sqrt{N}]$ average loss interval. Given this, Table 2 shows that the share of elements in the MCS is stable as M increases, consistent with the interpretation of the MCS as a confidence interval over average model losses.[15] Overall, both sets of results confirm that when the number of observations N is fixed, the theoretical properties of the MCS hold, even for large collections.

[12] The collection size for the following iteration was $M = 1077$ and would have exceeded the maximum RAM allocation of 25 GB per HPC node worker.

[13] Table 7 in Appendix B provides these scaling characteristics in standard units, obtained using polynomial regressions of the benchmarking data, in order to extrapolate the linear-in-logarithms trend of the elimination algorithm 1 to larger collection sizes, confirming the latter's computational intractability on larger collections.

[14] The table is designed to be comparable to table II of Hansen et al. (2011, p. 478).

[15] The slight decrease in the shares as M increases, all other parameters constant, is a reflection of the fact that we set $|\mathcal{M}^*| = 10$, skewing the shares upwards for smaller values of M.

Table 2
Monte Carlo large scale size and power analysis.

λ	$M = 500$, $\rho =$				$M = 1000$, $\rho =$				$M = 2000$, $\rho =$			
	0	0.5	0.75	0.95	0	0.5	0.75	0.95	0	0.5	0.75	0.95
	Panel 1: $\phi = 0$											
	Frequency at which $\mathcal{M}^* \subset \widehat{\mathcal{M}}^*_{90}$ *(size)*											
5	0.999	0.997	0.997	0.990	1.000	0.998	0.998	0.998	1.000	1.000	0.994	0.997
10	0.996	0.991	0.990	0.973	0.997	0.996	0.996	0.985	1.000	0.998	0.997	0.993
20	0.988	0.985	0.982	0.964	0.997	0.993	0.989	0.982	0.997	0.997	0.996	0.993
40	0.984	0.973	0.965	0.937	0.993	0.984	0.983	0.947	0.994	0.991	0.987	0.982
	Average share of elements in $\widehat{\mathcal{M}}^*_{90}$ *(power)*											
5	0.815	0.594	0.418	0.187	0.833	0.611	0.427	0.186	0.847	0.627	0.438	0.192
10	0.417	0.290	0.207	0.095	0.428	0.299	0.209	0.093	0.441	0.310	0.215	0.094
20	0.206	0.148	0.106	0.054	0.210	0.147	0.105	0.048	0.216	0.150	0.105	0.047
40	0.106	0.078	0.059	0.035	0.105	0.073	0.053	0.027	0.106	0.074	0.052	0.024
	Panel 2: $\phi = 0.5$											
	Frequency at which $\mathcal{M}^* \subset \widehat{\mathcal{M}}^*_{90}$ *(size)*											
5	0.999	1.000	1.000	0.996	1.000	0.999	0.999	0.998	1.000	1.000	0.999	0.999
10	0.999	0.999	0.996	0.987	0.999	0.998	0.999	0.996	1.000	1.000	0.999	0.997
20	0.997	0.988	0.992	0.970	0.998	0.994	0.998	0.979	0.999	0.998	1.000	0.994
40	0.992	0.983	0.985	0.944	0.993	0.989	0.993	0.974	0.998	0.999	0.998	0.985
	Average share of elements in $\widehat{\mathcal{M}}^*_{90}$ *(power)*											
5	0.771	0.555	0.394	0.174	0.787	0.565	0.397	0.174	0.799	0.575	0.403	0.180
10	0.389	0.276	0.194	0.092	0.399	0.279	0.195	0.088	0.404	0.284	0.200	0.089
20	0.194	0.138	0.101	0.052	0.197	0.139	0.099	0.046	0.200	0.139	0.098	0.044
40	0.102	0.074	0.057	0.034	0.098	0.070	0.050	0.026	0.099	0.069	0.050	0.023
	Panel 3: $\phi = 0.8$											
	Frequency at which $\mathcal{M}^* \subset \widehat{\mathcal{M}}^*_{90}$ *(size)*											
5	1.000	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
10	1.000	1.000	1.000	0.996	1.000	1.000	1.000	0.998	1.000	1.000	1.000	1.000
20	0.998	0.998	0.996	0.997	1.000	0.999	0.999	0.993	1.000	1.000	1.000	0.999
40	0.997	0.995	0.993	0.982	1.000	0.999	0.997	0.989	1.000	1.000	0.998	0.995
	Average share of elements in $\widehat{\mathcal{M}}^*_{90}$ *(power)*											
5	0.636	0.460	0.328	0.149	0.639	0.456	0.322	0.145	0.644	0.456	0.324	0.146
10	0.325	0.231	0.168	0.082	0.328	0.229	0.165	0.076	0.323	0.229	0.162	0.074
20	0.167	0.120	0.089	0.048	0.163	0.118	0.084	0.041	0.163	0.116	0.083	0.038
40	0.089	0.067	0.052	0.033	0.083	0.061	0.045	0.024	0.082	0.059	0.042	0.021

Note: The power results are reported using the share $|\widehat{\mathcal{M}}^*_{90}|/M$, the absolute number of models included can be recovered by multiplying each share by the corresponding value of M.

5. Empirical application to multivariate GARCH forecasting

The empirical application combines elements of the design of Laurent et al. (2012) and of Liu et al. (2015), in order to examine which of the multivariate correlation specification (Laurent et al., 2012) or the individual volatility predictors (Liu et al., 2015) has the most impact on performance when forecasting the volatility of multiple correlated series. Because both designs individually examine a large number of specifications, the combination of both dimensions easily produces collection sizes that are intractable for the elimination algorithm 1.

5.1. Multivariate GARCH specifications

The multivariate GARCH modelling strategy used in the forecasting exercise is standard, and broadly follows that of Laurent et al. (2012), while basing each forecast on a volatility predictor, as done in Liu et al. (2015). Given a set of n financial indices, each $n \times 1$ vector of demeaned returns $\boldsymbol{r}_t$ is modelled as:

$$\boldsymbol{r}_t = \boldsymbol{\Theta}_t^{\frac{1}{2}} \boldsymbol{\nu}_t, \qquad E\left[\boldsymbol{\nu}_t\right] = 0,\ E\left[\boldsymbol{\nu}_t \boldsymbol{\nu}_t'\right] = \boldsymbol{I} \tag{32}$$

Where the time-varying covariance matrix $\boldsymbol{\Theta}_t$ forms the target of the forecast $\boldsymbol{H}_t$. This forecasted covariance matrix is decomposed into two components:

$$\boldsymbol{H}_t = \boldsymbol{D}_t \boldsymbol{R}_t \boldsymbol{D}_t \tag{33}$$

These are a $n \times n$ diagonal matrix $\boldsymbol{D}_t = \text{diag}\left(\sigma_{1,t}, \sigma_{2,t}, \dots \sigma_{n,t}\right)$ containing the time-varying standard deviations from a vector of volatility predictors $\boldsymbol{z}_t$, and a time-varying correlation matrix $\boldsymbol{R}_t$. This enables a 2-stage estimation process, where the univariate volatilities $\sigma_{n,t}$, are first estimated from the volatility predictor and combined into $\boldsymbol{D}_t$, before their correlations are estimated using various conditional correlation models for $\boldsymbol{R}_t$.

Table 3 summarises the 50 univariate specifications used to model the diagonal entries of $\boldsymbol{D}_t$, from simple ARCH models to more sophisticated specifications such as power, exponential or fractionally integrated GARCH models. Each of these univariate

Table 3
Description of univariate volatility models.

Code	Description	Parameterisations	Number
AR	ARCH(P)	$P \in \{1,2\}$	2
GA	GARCH($P, O, Q, power$)	$P \in \{1,2\}$, $O \in \{0,1\}$, $Q \in \{1,2\}$, $power \in \{1,2\}$	16
FI	FIGARCH($P, Q, power$)	$P \in \{1,2\}$, $O \in \{0,1\}$, $power \in \{1,2\}$	8
EG	EGARCH(P, O, Q)	$P \in \{1,2\}$, $O \in \{0,1\}$, $Q \in \{1,2\}$	8
HA	HARCH($lags$)	$lags \in \{\{1,5\},\{1,5,44\}\}$	2
MI	MIDAS($m, asym$)	$m \in \{5,22,44\}$, $asym \in \{TRUE, FALSE\}$	6
AP	APARCH(P, O, Q)	$P \in \{1,2\}$, $O \in \{0,1\}$, $Q \in \{1,2\}$	8
	Total number of univariate specifications:		**50**

Note: Each set of model specifications is generated using combinations of the parameter values provided. P, O and Q are respectively the order of the symmetric innovation, the asymmetric innovation and the order of the lagged conditional variance, while *power* refers to the power exponent on the innovations. For the HARCH model, two sets of lags are used, while for the MIDAS model, m represents the highest-order lag included and $asym$ is a Boolean flag indicating whether innovations are asymmetric.

specifications is estimated using both standard normal and Student t innovations, resulting in a total of 100 univariate specifications per volatility predictor.

The correlation between the series, $\boldsymbol{R}_t$, is modelled using 4 distinct multivariate models. The most flexible specification, which allows for time evolution of $\boldsymbol{R}_t$ and asymmetric responses to positive and negative shocks, is a scalar version of the Asymmetric Dynamical Conditional Correlation (DCCA) model of Cappiello et al. (2006).[16]

$$\begin{cases} \boldsymbol{R}_t = \boldsymbol{Q}_t^{*-\frac{1}{2}} \boldsymbol{Q}_t \boldsymbol{Q}_t^{*-\frac{1}{2}} \\ \boldsymbol{Q}_t = (1-\alpha-\beta)\bar{\boldsymbol{Q}} + \alpha \boldsymbol{\epsilon}_{t-1}\boldsymbol{\epsilon}_{t-1}' + \beta \boldsymbol{Q}_{t-1} + \gamma(\boldsymbol{a}_{t-1}\boldsymbol{a}_{t-1}' - \bar{\boldsymbol{A}}) \end{cases} \tag{34}$$

Where $\boldsymbol{\epsilon}_t = \boldsymbol{D}_t^{-1}\boldsymbol{z}_t$ is the vector of standardised returns for the volatility predictor $\boldsymbol{z}_t$ and $\boldsymbol{a}_t = I\left(\boldsymbol{\epsilon}_t < 0\right) \odot \boldsymbol{\epsilon}_t$ isolates the negative entries of $\boldsymbol{\epsilon}_t$ with the indicator function $I(\dots)$, thus allowing asymmetric responses to positive and negative shocks. $\bar{\boldsymbol{Q}} = E\left[\boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t'\right]$ and $\bar{\boldsymbol{A}} = E\left[\boldsymbol{a}_t\boldsymbol{a}_t'\right]$ are the covariance matrices for the corresponding vectors, which in practice use the sample covariances $(\sum_t \boldsymbol{\epsilon}_t\boldsymbol{\epsilon}_t')/T$ and $(\sum_t \boldsymbol{a}_t\boldsymbol{a}_t')/T$ respectively. $\boldsymbol{Q}_t^* = \text{diag}\left(q_{1,1,t}, q_{2,2,t}, \dots, q_{n,n,t}\right)$ is a diagonal matrix containing the entries of the main diagonal of $\boldsymbol{Q}_t$, which is used to normalise $\boldsymbol{Q}_t$, ensuring a unit main diagonal in $\boldsymbol{R}_t$. A sufficient condition for ensuring that (34) produces a sequence of positive-definite matrices $\boldsymbol{R}_t$ is that $1-\alpha-\beta-\gamma\lambda_{\max} > 0$, where $\lambda_{\max}$ is the largest eigenvalue of $\bar{\boldsymbol{Q}}^{-\frac{1}{2}}\bar{\boldsymbol{A}}\bar{\boldsymbol{Q}}^{-\frac{1}{2}}$.[17]

The 3 other multivariate specifications considered are restricted versions of the DCCA specification (34). The first is the Dynamical Conditional Correlation (DCC) model of Engle (2002), which maintains a time-varying correlation matrix, but treats positive and negative shocks symmetrically. This amounts to assuming $\gamma = 0$, reducing the condition for positive-definiteness of $\boldsymbol{R}_t$ to $1-\alpha-\beta > 0$. A further restriction is the Conditional Correlation (CCC) model of Bollerslev (1990), which ignores the possibility of time-varying correlations, setting a fixed $\boldsymbol{R}_t = \bar{\boldsymbol{R}}\ \forall\ t$. This is equivalent to imposing the additional restriction that $\alpha = \beta = 0$ in (34), leading to $\boldsymbol{Q}_t = \bar{\boldsymbol{Q}}$. Finally, the most restricted specification ignores even the possibility of correlation between returns, naïvely setting $\boldsymbol{R}_t = \boldsymbol{I}\ \forall\ t$, treating returns as independent univariate processes. While this latter specification clearly discards information from correlations between series that could be used to improve forecasts, it does reflect the setting where a researcher uses univariate specifications to forecast a time-series when additional, correlated, series are available which might improve the forecast. By re-casting of a set of n univariate specifications into a multivariate framework, this provides a benchmark against which the performance of the 3 correlated specifications (CCC, DCC and DCCA) can be assessed.

5.2. Data, forecasting and MCS methodologies

The dataset used in the analysis is version 0.3 of the Oxford-Man realised volatility library of Heber et al. (2009), which provides a wide range of realised volatility (RV) measures, pre-computed from tick-by-tick data, for a selection of stock market indices. Table 4 lists the 12 realised variance estimators used as predictors $\boldsymbol{z}_t$ in the forecasting exercise, including several of the jump-robust indicators found to perform well in Liu et al. (2015). Overall, this set of 12 predictors is much less rich than theirs, in particular on the sampling frequency dimension, as Liu et al. (2015) include predictors sampled at 1 s, 5 s, 1 m, 5 m, and 15 m frequencies, as well as tick-by-tick and daily volatility.

The design contains 50 distinct univariate volatility specifications, with 2 distributions of the innovation terms, all estimated on 12 RV predictors, which are then combined with 4 multivariate specifications for the correlation between predictors. Because the spirit of the MCS is to serve as a confidence interval for models over a discrete support, the spread of the MCS along these 4 design dimensions directly indicates which is more critical in ensuring the performance of a forecast. Individually, there are fewer

[16] In principle the full DCCA specification allows for parameters in (34) to be full matrices.
[17] This parameter restriction is verified and imposed if necessary during the estimation of the corresponding multivariate model.

Table 4
Description of Variance Estimators and stock indices.

Variance Estimators		Stock market indices	
Code	Description	Symbol	Name
bv	Bipower Variation (5-min)	$.SPX$	S&P 500
bv_ss	Bipower Variation (5-min subs)	$.IXIC$	Nasdaq 100
medrv	Median RV (5-min)	$.FCHI$	CAC 40
rk_pa	Kernel RV (Non-Flat Parzen)	$.FTSE$	FTSE 100
rk_th2	Kernel RV (Tukey-Hanning(2))	$.STOXX50E$	EURO STOXX 50
rk_ts	Kernel RV (Two-Scale/Bartlett)	$.AORD$	All Ordinaries
rsv	Realised Semi-variance (5-min)	$.HSI$	HANG SENG
rsv_ss	Realised Semi-variance (5-min subs)	$.N225$	Nikkei 225
rv10	RV (10-min)	$.KS11$	KOSPI
rv10_ss	RV (10-min subs)	$.SSEC$	Shanghai Composite
rv5	RV (5-min)		
rv5_ss	RV (5-min subs)		

Note: 'RV' refers to realised variance, 'subs' to a sub-sampled calculation.

RV predictors than Liu et al. (2015), and fewer multivariate GARCH specification than Laurent et al. (2012), however combining these dimensions results in 4800 models, which is an order of magnitude larger than the collection examined by either.[18]

Table 4 also lists the 10 stock market series used in the analysis. These are spread across the main financial time zones, in order to capture correlations from both local and global events, while avoiding indices that are too strongly correlated (for example, including both the S&P500 and Dow Jones indices), which would have led to multicollinearity problems in the estimation of the correlation matrix $\boldsymbol{R}_t$. Note that while the number of series used in the analysis corresponds to what is used in Laurent et al. (2012), the change of focus from individual stocks to aggregated stock indices might affect the findings.

The forecasting scheme follows Laurent et al. (2012) by using two time periods: one where the testing data is taken in a relatively calm, low-volatility, period and one which covers the 2008–2009 financial crisis, providing a high volatility setting. Fig. 2 illustrates these periods for 3 of the 10 indices used. Each testing period consists of 550 observations, with both the 1-day and 5-day horizon forecasts generated using a rolling window of 22 days, with the underlying forecasting model re-estimated using the previous 1500 observations in each case.[19] This choice of sample size for the estimation of the models is designed to offer a compromise between the 500 used in Liu et al. (2015) and the 2740 used in Laurent et al. (2012).

The loss function used for evaluating the performance of a given volatility forecast $\boldsymbol{H}_t$ relative to the true quadratic variation $\boldsymbol{\Theta}_t$ is the following Stein loss. In practice, $\boldsymbol{\Theta}_t$ provided by a proxy which is discussed below.

$$L_t = \operatorname{tr}\left(\boldsymbol{H}_t^{-1}\boldsymbol{\Theta}_t\right) - \ln\left|\boldsymbol{H}_t^{-1}\boldsymbol{\Theta}_t\right| - n \tag{35}$$

The Stein loss (35) maintains comparability with both base designs: it is one of the loss functions investigated in Laurent et al. (2012), and it forms the multivariate counterpart to the univariate QLIKE loss used by Liu et al. (2015), where $L_t = \theta_t/h_t - \ln\left(\theta_t/h_t\right) - 1$.

As pointed out by Patton (2011), the true quadratic variation $\boldsymbol{\Theta}_t$ is not observable, therefore a proxy is required for the loss (35). The forecasting exercise follows Liu et al. (2015) and uses a one-day lead of the 5 min RV (rv5) series as the proxy, with the lead ensuring that the error in the proxy is uncorrelated with the error in the forecast. As a robustness check, the analysis also uses the lead of the standard squared open-to-close daily returns. Patton (2011), Liu et al. (2015) point out that while this results in a noisier measure of the true $\boldsymbol{\Theta}_t$, it nevertheless allows consistent estimation of the losses.

Finally, the MCS reported below are obtained by running the two-pass fast MCS algorithm 2 on each of the 550×4800 loss matrices, setting $\alpha = 0.1$. The analysis uses 1000 replications of the Politis and Romano (1994) stationary bootstrap, with an average bootstrap block size of 10, as in Liu et al. (2015).[20] In addition, the performance gains associated with the fast MCS algorithm allow us to run a Hurlin et al. (2017) bucketing analysis, in line with definition 2, in order to obtain an estimate $\widehat{\mathcal{P}}_{1-\alpha}$ of the true partition $\mathcal{P}$. This partition analysis helps identify the determinants of forecast performance by examining the distribution of models over all estimated equivalence classes $\widehat{\mathcal{M}}^*_{k,1-\alpha}$ rather than the MCS $\widehat{\mathcal{M}}^*_{0,1-\alpha}$ alone.

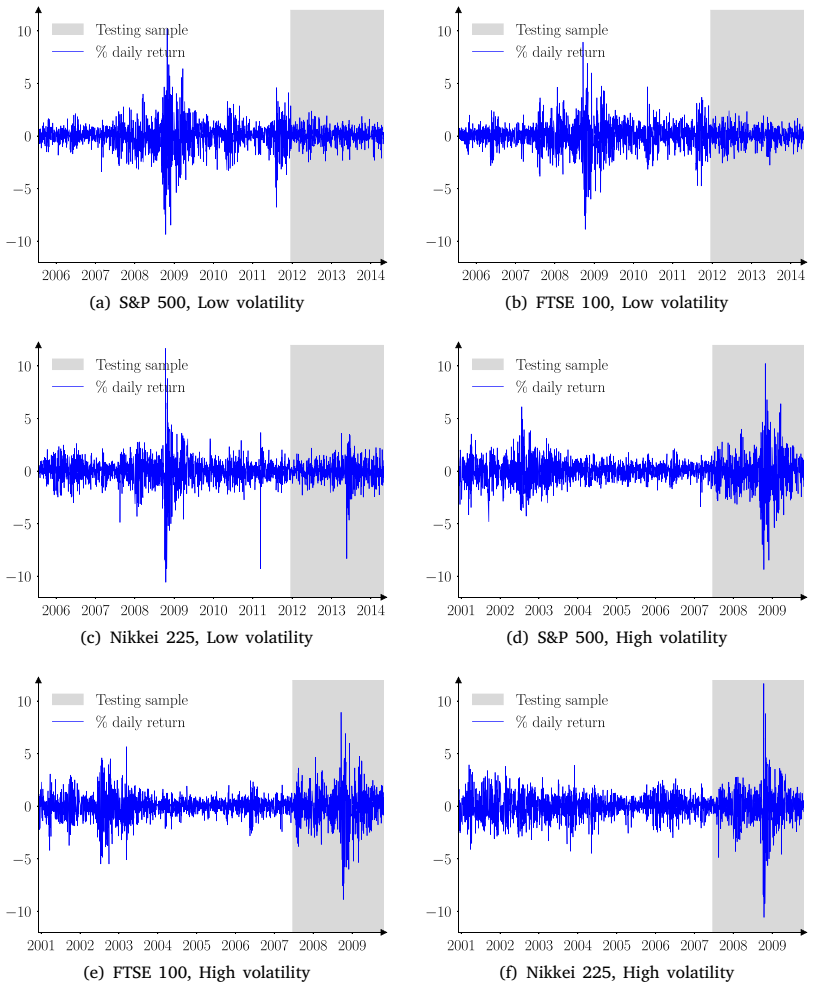
5.3. Distribution of the MCS

Tables 5 presents the MCS obtained using the 5-minute RV proxy. Each cell corresponds to a given multivariate specification and volatility predictor combination, and reports the number of models included in the MCS for $\alpha = 0.1$. The intensity of the shaded background reflects the average equivalence class index $\bar{k}$ of the 100 univariate specifications in that cell, with darker

[18] fastMCS can process this collection in 5 min, using 200 MB of memory. Extrapolating from Table 7, the elimination algorithm 1 would require 24 h and 483 GB.

[19] For the low volatility sample, the testing sample runs from 16/12/2011 to 02/05/2014, with the training data starting on 21/07/2005. The high volatility training data starts on 06/12/2000, with the testing period running from 20/06/2007 to 26/10/2009.

[20] A second analysis was run for robustness using a standard block bootstrap with a fixed block size of 10 observations. Results are very similar and are therefore provided in the supplementary material.



(a) S&P 500, Low volatility (b) FTSE 100, Low volatility

(c) Nikkei 225, Low volatility (d) S&P 500, High volatility

(e) FTSE 100, High volatility (f) Nikkei 225, High volatility

Fig. 2. Illustration of training and testing samples.

colours indicating lower average equivalence class indices (better average ranking). The results show that the MCS is concentrated practically exclusively on the DCCA specifications, which supports the findings of Laurent et al. (2012) and suggests the presence of significant correlated and asymmetric responses to shocks. The performance of the DCC and CCC specifications, however, is not far removed from the DCCA. First, both the low and high volatility MCS contain an additional Naïve and CCC specification for the 5-day horizon forecasts, in both cases based on an EGARCH specification. This is in line with their point that adequate modelling of asymmetries at the univariate level can help offset their absence in the conditional correlation model. Second, the partition analysis confirms that the average performance of the CCC, DCC and DCCA specifications is often close. Finally, the MCS obtained using the open-to-close proxy, shown in Table 6, shows that the use of a noisier volatility proxy for $\boldsymbol{\Theta}_t$ widens the MCS to include more CCC and DCC specifications.

The MCS shows no equivalent concentration on other dimensions of the design. Regarding specific volatility predictors, the bi-power variation and realised semi-variance, along with their subsampled variants, do dominate in Table 5 however forecasts based on every predictor are still included in the MCS. While this suggests that the choice of the of RV predictor is less critical that that of the multivariate volatility model, it is important to point out that the conclusions of Liu et al. (2015) regarding the predictive power of RV measures still hold, as several aspects of these findings are consistent with their results. First, their results for composite indices similarly show that nearly all RV proxies are present in the MCS to some degree. This is not the case however for individual stocks and futures, where 5-minute RV dominates more strongly, suggesting that the present analysis might not generalise to those variables. Second, the results presented here suffer from a *de facto* selection effect compared to the wider sampling frequencies in the Liu et al. (2015) design. The RV library of Heber et al. (2009) contains mainly predictors sampled at 5-minute frequencies, which are known to perform well, and having selected predictors based on evidence of their forecasting performance, it should not be surprising that they are all present in the MCS. Finally, the partition analysis confirms that certain predictors lead to clear improvement in average performance, especially the bi-power variation and realised semi-variance measures. This is also the case,

Table 5
Models in the $\alpha = 0.1$ MCS for the rv5 proxy, stationary bootstrap.

	1-day horizon				5-day horizon			
	Naïve	CCC	DCC	DCCA	Naïve	CCC	DCC	DCCA
Low volatility sample								
	$\hat{K} = 88$				$\hat{K} = 126$			
bv	–	–	–	9	–	–	–	9
bv_ss	–	–	–	10	–	–	–	10
medrv	–	–	–	36	–	–	–	25
rk_pa	–	–	–	21	–	–	–	21
rk_th2	–	–	–	44	–	–	–	44
rk_ts	–	–	–	38	–	–	–	38
rsv	–	–	–	30	–	–	–	30
rsv_ss	–	–	–	44	–	–	–	43
rv10	–	–	–	10	–	–	–	10
rv10_ss	–	–	–	10	–	–	–	10
rv5	–	–	–	23	–	–	–	23
rv5_ss	–	–	–	28	1	–	–	28
Total	–	–	–	**303**	**1**	–	–	**291**
High volatility sample								
	$\hat{K} = 100$				$\hat{K} = 84$			
bv	–	–	–	24	–	–	–	24
bv_ss	–	–	–	38	–	–	–	38
medrv	–	–	–	21	–	–	–	18
rk_pa	–	–	–	9	–	–	–	9
rk_th2	–	–	–	26	–	–	–	26
rk_ts	–	–	–	25	–	–	–	25
rsv	–	–	–	35	–	–	–	35
rsv_ss	–	–	–	30	–	–	–	34
rv10	–	–	–	13	–	–	–	13
rv10_ss	–	–	–	10	–	–	–	10
rv5	–	–	–	9	–	–	–	9
rv5_ss	–	–	–	11	–	1	–	11
Total	–	–	–	**251**	–	**1**	–	**252**

Note: Each combination contains 100 specification (50 univariate, 2 distributions).
Legend for the partition analysis: $\bar{k} = \hat{K}$ | $\bar{k} = \hat{K}/2$ | $\bar{k} = 0$

although to a lesser degree, in the noisier open-to-close proxy case of Table 6, where median RV performs poorly everywhere. The choice of predictor can therefore still make a difference for average performance in the high volatility case.

Tables 8 and 9, in Appendix B, provide the distribution of the MCS over univariate specifications, and reveal a similarly dispersed pattern. While the GARCH, EGARCH and FIGARCH specifications are more numerous and ARCH, HARCH and MIDAS are excluded more often, all univariate specifications are included in the MCS to some extent. The partition analysis confirms that for the 1-day horizon, the choice of RV predictor is more important that the univariate specification for average model ranking, especially in high-volatility situations. This is not the case for the 5-day horizon, as the GARCH, EGARCH and FIGARCH specifications show slightly better average performance.

6. Discussion and use cases

The main contribution of this paper is to establish that the R-rule MCS of a collection can be obtained using a vector updating strategy. The resulting implementation, referred to as fastMCS, offers one polynomial order less time complexity and memory cost, and is proven to almost surely replicate the output of the existing elimination approach with no additional assumptions required. This dramatic improvement in time complexity and memory allows for faster processing, larger collections and potentially, collaborative updating of collections as new models are added.

However, while the computational improvements highlighted in the Monte Carlo simulations and empirical application are significant, two issues raise the question of the practical usefulness of the fastMCS algorithm. First of all, as mentioned in Section 1, a review of the literature reveals that while the MCS methodology is popular, the majority of empirical applications use small collection sizes, where the performance gains from using fastMCS would be negligible in practice. Second, the possibility of collaborative research raised by White (2000) and enabled by fastMCS is limited by the fact that data observations have to remain fixed, due to the bootstrap implementation. Ex-post updating of the $N \times M$ losses L with fastMCS is therefore only possible when adding extra models. Given that any collaborative MCS exercise would require coordination, its practical life-expectancy would therefore be limited by the frequency at which new data is collected, and forecasting or empirical windows shift, a particularly acute problem in the high-frequency data environment of financial forecasting.

Nevertheless, there are at least three use cases for fastMCS that bypass these two limitations. First, there can be cases where a given MCS analysis has to be carried out a large number of times, making the improved performance desirable even on smaller datasets, and regardless of any ex-post collaborative effort. Examples of this are when the MCS algorithm forms part of a wider

Table 6
Models in the $\alpha = 0.1$ MCS for the open-to-close proxy, stationary bootstrap.

	1-day horizon				5-day horizon			
	Naïve	CCC	DCC	DCCA	Naïve	CCC	DCC	DCCA
Low volatility sample								
	$\hat{K} = 45$				$\hat{K} = 82$			
bv	–	–	–	18	–	2	1	33
bv_ss	–	–	1	16	–	3	2	34
medrv	–	–	–	-	–	–	–	1
rk_pa	–	–	–	21	–	–	–	37
rk_th2	–	–	–	44	–	–	–	44
rk_ts	–	–	–	38	–	–	–	39
rsv	–	–	–	26	–	–	–	16
rsv_ss	–	–	–	38	–	–	–	26
rv10	–	–	–	10	–	–	–	10
rv10_ss	–	–	–	10	–	–	–	10
rv5	–	–	–	23	–	–	–	23
rv5_ss	–	–	1	28	1	–	–	28
Total	–	–	**2**	**272**	**1**	**5**	**3**	**301**
High volatility sample								
	$\hat{K} = 64$				$\hat{K} = 58$			
bv	–	–	–	54	–	13	11	55
bv_ss	–	–	–	69	–	11	9	68
medrv	–	–	–	-	–	–	–	7
rk_pa	–	–	–	10	–	54	51	41
rk_th2	–	–	–	26	–	–	–	26
rk_ts	–	–	–	25	–	–	–	25
rsv	–	–	–	35	–	–	–	31
rsv_ss	–	–	–	31	–	–	–	28
rv10	–	–	–	13	–	19	17	24
rv10_ss	–	–	–	10	–	19	18	19
rv5	–	–	–	9	–	23	22	20
rv5_ss	–	1	–	11	–	23	22	19
Total	–	**1**	–	**293**	–	**162**	**150**	**363**

Note: Each combination contains 100 specification (50 univariate, 2 distributions).
Legend for the partition analysis: $\bar{k} = \hat{K}$ | $\bar{k} = \hat{K}/2$ | $\bar{k} = 0$

Monte-Carlo analysis, as in Section 4 or in Samuels and Sekkel (2017), where the collection only contains 107 models, but the trimming analysis requires 500 MC replications. An additional setting that would fit this use case is the bucketing procedure of Hurlin et al. (2017), with a collection size of only 94, but requiring up to 49 sequential MCS runs.

Second are cases where the empirical dataset of interest is either fixed or updated slowly. In such cases, collaborative efforts can grow model collection sizes significantly before any empirical data shift becomes a concern for the relevance of the analysis. In addition, assuming that new data does arrive at a slower rate, the improved performance of fastMCS would allow rapid re-evaluation of the MCS when the dataset is updated with newer observations. A good example of such an empirical setting from the literature reviewed is the previously mentioned climate science evaluation of Newell et al. (2021), which already processes one of the largest existing collections of models.

A third use case, which would be useful even on faster-evolving data, is to use fastMCS as a benchmarking tool for evaluating the performance of new forecasting or prediction methodologies. This would directly be useful in the large-scale machine-learning competitions mentioned in Section 1, which may receive many similar submissions yet currently do not test for superior predictive ability. Similarly, a large collection of standard prediction or forecasting methods could be prepared, for example along the lines of the design used in Section 5, with the MCS evaluated on a series of reference datasets. These could be made available to researchers, who could update them with their own methodologies, thus evaluating their performance. Such a standardised benchmarking tool would prove useful in the context of the many contributions to forecasting of commodities, financial indices and cryptocurrencies discussed in Section 1, and help provide a more holistic view of the relative performance of these methodologies.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The author is extremely grateful to Miguel León-Ledesma, Stefano Grassi and Guy Tchuente for their helpful advice on data snooping and the MCS methodology, to Jagjit Chadha for his comments on an earlier version of this paper, to participants of the CFE 2015 session on forecasting evaluation for a stimulating discussion, and to two anonymous referees for their helpful suggestions.

Special thanks goes to James Holdsworth, Mark Wallis and specialist and High Performance Computing systems provided by Information Services at the University of Kent for maintaining the computer cluster on which the Monte Carlo analysis was run. Any errors in the manuscript remain of course the author's.

Appendix A. Proofs

This appendix contains the proofs for the lemmas, corollaries and the main proposition relating to the theoretical equivalence of the fast and elimination MCS algorithms.

Proof of Lemma 1. $\forall k \in \mathcal{M}$, the equivalence statistic (7) is $T_k = \max_i \max_j |t_{i,j}|$, $\forall i,j \in \mathcal{E}_k^+ \cup \{k\}$. Decomposability of the maximum function and the skew-symmetry of (1) imply that:

$$T_k = \max\left(\max_i \max_j |t_{i,j}|, \max_i(t_{k,i})\right) \qquad \forall i,j \in \mathcal{E}_k^+$$

The iterative nature of the elimination rule ensures $T_k > T_i \ \forall i \in \mathcal{E}_k^+$, therefore $T_k = \max_i(t_{k,i}) \ \forall i \in \mathcal{E}_k^+$. Given this, we now prove that (12), (13), (14) and (15) each hold when the relevant conditions are satisfied.

First, for m itself, the equivalence statistic (7) is $T_m = \max_i(t_{m,i}) \ \forall i \in \mathcal{E}_m^+$. Given $\mathcal{M}' = \mathcal{E}_m^+ \cup \mathcal{E}_m^-$, if condition 2.1 is satisfied, then $\max_i(t_{m,i}) = \max_j(t_{m,j})$, $i \in \mathcal{E}_m^+$, $j \in \mathcal{M}'$ and (12) holds.

Next, for $k \in \mathcal{E}_m^+$ the elimination rule (7) ensures that $m \notin \mathcal{E}_k^+$. This implies that $\forall i,j \in \mathcal{E}_m^+$ we have $t_{i,j} = t'_{i,j}$, $\max_i(t_{i,j}) = \max_i(t'_{i,j})$ and $T_k = T'_k$. It immediately follows from this that Eq. (14) holds $\forall k \in \mathcal{E}_m^+$. In addition, if condition 2.1 holds and T_m is correctly obtained using (12), this also implies:

$$\begin{aligned}\{k \in \mathcal{M}' : T'_k < T_m\} &= \{k \in \mathcal{M} : T_k < T_m\} \\ &= \{k \in \mathcal{M} : \varepsilon_k > \varepsilon_m\} \equiv \mathcal{E}_m^+\end{aligned}$$

With the second equality following from $\varepsilon_i > \varepsilon_j$ iff $T_i < T_j$ under the elimination rule (7). The first equation in (13) therefore holds, with the second equation holding by construction given that $\mathcal{M}' = \mathcal{E}_m^+ \cup \mathcal{E}_m^-$.

Finally, for $k \in \mathcal{E}_m^-$, the equivalence statistic (7) is $T_k = \max_i(t_{k,i})$, $i \in \mathcal{E}_k^-$. Because $m \in \mathcal{E}_k^+$, the decomposability of the max function implies $T_k = \max\left(\max_i(t_{k,i}), t_{k,m}\right) \ \forall \ i \in \mathcal{E}_k^+ \setminus \{m\}$. If either condition 2.2 or 2.3 are satisfied, then $\max_i(t_{k,i}) = \max_i(t'_{k,i}) = T'_k$ and Eq. (15) holds. ■

Proof of Lemma 2. The proof follows the same structure as that of Lemma 1, *mutatis mutandis*. $\forall k \in \mathcal{M}$ and $\forall b \in \mathcal{B}$ the bootstrapped test statistic (10) is $\mathcal{T}_{k,b} = \max_i \max_j |\tau_{i,j,b}| \ \forall i,j \in \mathcal{E}_k^+ \cup \{k\}$. Decomposability of the maximum function and the skew-symmetry of (1) imply that:

$$T_{k,b} = \max\left(\max_i \max_j |\tau_{i,j,b}|, \max_i(\tau_{k,i,b})\right) \qquad \forall i,j \in \mathcal{E}_k^+$$

Given this, we now prove that (16), (17) and (18) each hold for all bootstrap replications $b \in \mathcal{B}$ when the relevant conditions are satisfied.

First, for m itself, the bootstrapped test statistic (10) is $T_{m,b} = \max\left(\max_i \max_j |\tau_{i,j,b}|, \max_i(\tau_{m,i,b})\right) \ \forall i,j \in \mathcal{E}_m^+$. If condition 2.1 is satisfied, then Lemma 1 ensures that $\mathcal{E}_m^+$ and $\mathcal{E}_m^-$ are correctly identified. By definition $\forall i,j \in \mathcal{E}_m^+$, $\max_i \max_j |\tau_{i,j,b}| = \mathcal{T}_{m^+,b} = \mathcal{T}'_{m^+,b}$ is the bootstrapped equivalence statistic of the model eliminated immediately after m therefore Eq. (16) holds.

Next, for $i \in \mathcal{E}_m^+$ the elimination rule (7) ensures that $m \notin \mathcal{E}_i$. This implies that $\forall i,j \in \mathcal{E}_m^+$ we have $\tau_{i,j} = \tau'_{i,j}$, $\max_i \max_j |\tau_{i,j,b}| = \max_i \max_j |\tau'_{i,j,b}|$ and $\mathcal{T}_{k,b} = \mathcal{T}'_{k,b}$, therefore Eq. (17) holds.

Finally, for $k \in \mathcal{E}_m^-$, the bootstrapped test statistic (10) is $T_{k,b} = \max\left(\max_i \max_j |\tau_{i,j,b}|, \ \max_i |\tau_{m,i,b}|\right) \ \forall i,j \in \mathcal{E}_k^+ \setminus \{m\}$. If condition 2.3 is satisfied, then $\mathcal{E}_k^+ \setminus \{m\} = \mathcal{E}_k'^+$. As a result, $\max_i \max_j |\tau_{i,j,b}| = \max_i \max_j |\tau'_{i,j,b}| = \mathcal{T}'_{k,b} \ \forall i,j \in \mathcal{E}_k'^+$ and Eq. (18) holds. ■

Proof of Lemma 3. First, $\mathcal{P}$ meets all the conditions required of a partition of $\mathcal{M}_0$. Because $|\mathcal{M}_k^*| \geq 1$, each subset is non-empty and the number of subsets $K = |\mathcal{P}| \leq M$ given the finite collection size $M = |\mathcal{M}_0|$. Because $\mathcal{P}$ is obtained by recursively partitioning $\mathcal{M}_0$, by construction one has $\mathcal{M}_0 = \bigcup_{k=0}^{P-1} \mathcal{M}_k^*$ and $\mathcal{M}_a^* \cap \mathcal{M}_b^* = \varnothing$ for $a \neq b$.

Next, for $\mathcal{P}$ to be an ordered partition, each $\mathcal{M}_k^* \in \mathcal{P}$ must be an equivalence class and the set of equivalence classes must be strictly ordered. From the definition of the set of superior objects (3) and the skew-symmetry of (1), one has $\mu_{i,j} > 0 \ \forall i \in \mathcal{M}_a^*$, $\forall j \in \mathcal{M}_b^*$ and $b > a$. In turn this implies $\mu_{i,j} = 0$, $\forall i,j \in \mathcal{M}_k^*$, $\forall k$. From the latter statement, any two models i,j in the same class $\mathcal{M}_k^*$ offer equivalent performance: models cannot be ordered within classes, so each $\mathcal{M}_k^*$ is an equivalence class and (23) holds. From the former statement and given $b > a$, any model $i \in \mathcal{M}_a^*$ offers better performance than any model $j \in \mathcal{M}_b^*$, and $\mathcal{M}_a^*$ as a whole is therefore strictly preferred to $\mathcal{M}_b^*$. $\mathcal{P}$ is therefore an ordered partition, which immediately implies the existence of a strict weak ordering of all the models in $\mathcal{M}_0$. ■

Proof of Corollary 3.1. Let i, j and k be arbitrary models in $\mathcal{M}_0$. Then the definition of deviations (1) and the linearity of the expectations operator $E_n(.)$ in (2) ensures $\mu_{j,i} = \mu_{j,k} + \mu_{k,i}$. First, assume that model $i \in \mathcal{M}_0^*$. $\forall k \in \mathcal{M}_1$, this implies $\mu_{k,i} > 0$, which in turn ensures $\mu_{j,i} > \mu_{j,k}$, $\forall j \in \mathcal{M}_0$. Conversely, let i satisfy $\mu_{j,i} > \mu_{j,k}$, $\forall j \in \mathcal{M}_0$ and $\forall k \in \mathcal{M}_1$. Given that $\mathcal{M}_0 = \mathcal{M}_0^* \cup \mathcal{M}_1$ two cases are possible for j. In the case where $j \in \mathcal{M}_0^*$, let us suppose that $i \in \mathcal{M}_a^* \subset \mathcal{M}_1$. From Lemma 3, $\exists! \, k \in \mathcal{M}_a^* : \mu_{k,i} = 0$, which implies

that for that k we have $\mu_{j,i} = \mu_{j,k}$. This is a contradiction, therefore $i \in \mathcal{M}_0^*$. In the case where $j \in \mathcal{M}_a^* \subset \mathcal{M}_1$, then from Lemma 3, $\exists! \, k \in \mathcal{M}_a^* : \mu_{j,k} = 0$. This implies $\mu_{j,i} > 0$, and given the skew-symmetric nature of (1), it follows that $\mu_{i,j} < 0$, and therefore $i \in \mathcal{M}_0^*$. ■

Proof of Corollary 3.2. Given assumption 1, the central limit theorem for strongly mixing sequences ensures that $\sqrt{N}\sigma_{i,j}^{-1}(\bar{d}_{i,j} - \mu_{i,j}) \xrightarrow{d} \mathcal{N}(0,1)$ as $N \to \infty$. Given this, the t-statistics $t_{j,i}$ converge in distribution as follows:

$$t_{j,i} \xrightarrow{d} \mathcal{N}\left(\sqrt{N}\mu_{j,i}, 1\right) \tag{36}$$

The difference between two t-statistics $t_{j,i}$ and $t_{j,k}$ converges in distribution to:

$$t_{j,i} - t_{j,k} \xrightarrow{d} \mathcal{N}\left(\sqrt{N}(\mu_{j,i} - \mu_{j,k}), 2(1 - \text{corr}(t_{j,i}, t_{j,k}))\right) \tag{37}$$

For $i \in \mathcal{M}_0^*$, $j \in \mathcal{M}_0$ and $k \in \mathcal{M}_1$ Corollary 3.1 ensures $\mu_{j,i} > \mu_{j,k}$ therefore $t_{j,i} - t_{j,k}$ diverges to infinity at rate $\sqrt{N}$. This implies that $\Pr\left(t_{j,i} - t_{j,k} > 0\right) = \Pr\left(t_{j,i} > t_{j,k}\right) \to 1$ as $N \to \infty$. Given a collection of models $\mathcal{M}_0$ and a fixed model $j \in \mathcal{M}_0$ if $t_{j,i} - t_{j,k}$ diverges to infinity as $N \to \infty$, $\forall \, i \in \mathcal{M}_0^*, \forall \, k \in \mathcal{M}_1$, then clearly so does $\max_i(t_{j,i}) - \max_k(t_{j,k})$, therefore $\Pr\left(\arg\max_i(t_{j,i}) \in \mathcal{M}_0^*\right) \to 1$ as $N \to \infty$. ■

Proof of Proposition 1. In any iteration of either pass of algorithm 2, let $\mathcal{M}'$ be the collection of models that has already been processed, let m be an arbitrary model picked from $\mathcal{M}_0 \setminus \mathcal{M}'$, let $\mathcal{M} = \mathcal{M}' \cup \{m\}$, and let $T^* = \max_i \max_j |t_{i,j}|$, $\forall \, i, j \in \mathcal{M}^*$.

We first prove that the first pass of algorithm 2 results in $\Pr\left(\epsilon_m^f = \epsilon_m\right) \to 1 \; \forall \, m \in \mathcal{M}_1$. If $\mathcal{M}' = \varnothing$, then m is the first model processed and by construction, $\mathcal{M}^* = \{m\}$ and $T_m = 0$. Once $|\mathcal{M}'| \geq 1$, two cases are possible when a new model m is added to $\mathcal{M}'$:

1. Either $m \notin \mathcal{M}^*$. In this the case $\mathcal{M}^* \subseteq \mathcal{E}_m^+$, and Corollary 3.2 ensures that $\Pr\left(\arg\max_i(t_{m,i}) \in \mathcal{M}^*\right) \to 1$ as $N \to \infty$. Therefore conditions 2.1 and 2.2 hold almost surely as $N \to \infty$, and Lemma 1 almost surely provides the correct updates to the T_k values $\forall \, k \in \mathcal{M}$.
2. Or $m \in \mathcal{M}^*$. In this case, $\mathcal{M}^* \nsubseteq \mathcal{E}_m^+$ therefore even with Corollary 3.2, conditions 2.1 and 2.2 do not hold in general and Lemma 1 may produce incorrect values of T_k. Eq. (27) in Corollary 3.2 nevertheless provides two guarantees as $N \to \infty$. First, $\forall \, k \in \mathcal{M}^*$ we have $T_k \leq \max_i \max_j |t_{i,j}| = T^*$ almost surely $\forall \, i, j \in \mathcal{M}^*$. Second, $\forall \, k \notin \mathcal{M}^*$ it ensures that condition 2.2 holds almost surely, in which case (15) almost surely updates the corresponding T_k correctly.

In either case $T_k^f = T_k$ almost surely $\forall \, k \notin \mathcal{M}^*$ in each iteration. Once all $m \in \mathcal{M}_0$ have been added, we have $T_k^f = T_k$ almost surely $\forall \, k \in \mathcal{M}_1$ and $T_k^f \leq T^* \; \forall \, k \in \mathcal{M}_0^*$. Given this, sorting the models in $\mathcal{M}_0$ according to T_k to obtain the elimination sequence almost surely results in $\varepsilon_k^f = \varepsilon_k$ and $\varepsilon_i^f > \varepsilon_k \; \forall \, k \in \mathcal{M}_1$ and $\forall \, i \in \mathcal{M}_0^*$.

Next, we prove that the second pass of algorithm 2 ensures that $\Pr\left(P_m^f = P_m\right) \to 1 \; \forall \, m \in \mathcal{M}_1$. Because models are processed in reverse elimination order, in each iteration the additional model m is the first that would be eliminated from $\mathcal{M}$ under algorithm 1. As a result, only (16) is needed to update $\mathcal{T}_{m,b}$ in each iteration. Given that $\mathcal{M}_0 = \mathcal{M}_0^* \cup \mathcal{M}_1$, two cases are possible as $N \to \infty$:

1. $\forall \, k \in \mathcal{M}_1$, condition 2.1 holds almost surely, and the first pass almost surely results in $T_k^f = T_k$ and $\varepsilon_k^f = \varepsilon_k$. This in turn ensures that $\Pr(\mathcal{T}_{k,b}^f = \mathcal{T}_{k,b}) \to 1$, which means that given (10), we have $\Pr(P_k^f = P_k) \to 1$.
2. $\forall \, k \in \mathcal{M}_0^*$, condition 2.1 does not hold and the first pass does not guarantee $\varepsilon_k^f = \varepsilon_k$, allowing the possibility that $\mathcal{T}_{k,b}^f \neq \mathcal{T}_{k,b}$. However, let $k^* \in \mathcal{M}_0^* : T_{k^*}^f = T^*$ and $m^* \in \mathcal{M}_0^* : T_{m^*} = T^*$ indicate the models in $\mathcal{M}_0^*$ identified by algorithms 2 and 1 respectively as possessing the largest test statistic T^*. As $T_{k^*}^f = T_{m^*} = T^*$ we have $\varepsilon_{k*}^f = \varepsilon_{m^*}$ therefore $\Pr(\mathcal{T}_{k^*,b}^f = \mathcal{T}_{m^*,b}) \to 1$ and $\Pr(P_{k^*}^f = P_{m^*}) \to 1$. Given the definition of P-values (10), this ensures $\Pr(P_k^f \geq P_{m^*}) \to 1 \; \forall \, k \in \mathcal{M}_0^*$.

Once all $k \in \mathcal{M}_0$ have been processed, this ensures $\Pr(P_k^f = P_k) \to 1$ for $\forall \, k \in \mathcal{M}_1$ and $\Pr(P_k^f \geq P_{m^*}) \to 1 \; \forall \, k \in \mathcal{M}_0^*$ as $N \to \infty$. ■

Appendix B. Tables

See Tables 7–9.

Appendix C. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jeconom.2025.106123.

Table 7
Scaling projections of MCS algorithms, $N = 250$.

	Elimination		Two-pass	
	Time (sec)	Memory (MB)	Time (sec)	Memory (MB)
$M = 500$	128	6017	2	15
$M = 1000$	982	24,055	11	28
$M = 2000$	7667	96,219	50	54
$M = 5000$	117,988	601,399	366	132
$M = 10000$	939,038	2,405,669	1799	263

Note: Table entries are rounded point estimates obtained from OLS polynomial regressions on the 3200 and 6400 MC benchmarking data points, for the elimination and two-pass algorithms respectively.

Table 8
Models in the $\alpha = 0.1$ MCS for the rv5 proxy, stationary bootstrap.

	1-day horizon							5-day horizon						
	AR	GA	FI	EG	HA	MI	AP	AR	GA	FI	EG	HA	MI	AP
Low volatility sample														
	$\hat{K} = 88$							$\hat{K} = 126$						
bv	–	2	1	2	–	3	1	–	2	1	2	–	3	1
bv_ss	–	4	2	2	–	1	1	–	4	2	2	–	1	1
medrv	1	10	7	6	2	5	5	1	6	6	1	2	5	4
rk_pa	2	10	2	2	–	3	2	2	10	2	2	–	3	2
rk_th2	2	13	8	8	2	5	6	2	13	8	8	2	5	6
rk_ts	2	11	4	7	2	4	8	2	11	4	7	2	4	8
rsv	–	8	6	5	–	6	5	–	9	6	5	–	5	5
rsv_ss	–	12	9	9	1	6	7	–	11	9	9	1	6	7
rv10	1	5	1	1	–	1	1	1	5	1	1	–	1	1
rv10_ss	2	4	1	–	1	1	1	2	4	1	–	1	1	1
rv5	2	10	3	4	2	1	1	2	10	3	4	2	1	1
rv5_ss	1	9	5	3	2	5	3	1	9	5	4	2	5	3
Total	**13**	**98**	**49**	**49**	**12**	**41**	**41**	**13**	**94**	**48**	**45**	**12**	**40**	**40**
High volatility sample														
	$\hat{K} = 100$							$\hat{K} = 84$						
bv	–	10	5	5	1	1	2	–	10	5	5	1	1	2
bv_ss	1	12	7	6	2	5	5	1	12	7	6	2	5	5
medrv	–	7	5	4	1	1	3	1	4	6	3	1	3	–
rk_pa	–	2	3	3	–	–	1	–	2	3	3	–	–	1
rk_th2	1	8	5	4	1	4	3	1	8	5	4	1	4	3
rk_ts	1	8	4	3	2	3	4	1	8	4	3	2	3	4
rsv	1	10	9	8	1	5	1	1	10	10	8	1	5	–
rsv_ss	2	8	8	5	–	4	3	2	10	10	6	–	4	2
rv10	1	5	2	2	–	1	2	1	5	2	2	–	1	2
rv10_ss	1	3	2	2	–	–	2	1	3	2	2	–	–	2
rv5	1	1	3	1	–	2	1	1	1	3	1	–	2	1
rv5_ss	1	2	4	3	–	–	1	1	2	4	4	–	–	1
Total	**10**	**76**	**57**	**46**	**8**	**26**	**28**	**11**	**75**	**61**	**47**	**8**	**28**	**23**

Note: Each combination contains the following number of specifications: AR - 16, GA - 128, FI - 64, EG - 64, HA - 16, MI - 48, AP - 64.

Legend for the partition analysis: $\bar{k} = \hat{K}$ | $\bar{k} = \hat{K}/2$ | $\bar{k} = 0$

Table 9
Models in the $\alpha = 0.1$ MCS for the open-to-close proxy, stationary bootstrap.

	1-day horizon							5-day horizon						
	AR	GA	FI	EG	HA	MI	AP	AR	GA	FI	EG	HA	MI	AP
Low volatility sample														
	$\hat{K} = 45$							$\hat{K} = 82$						
bv	–	3	7	3	–	4	1	–	15	7	5	1	4	4
bv_ss	–	5	4	3	1	3	1	–	19	5	7	1	2	5
medrv	–	–	–	–	–	–	-	1	–	–	–	–	–	–
rk_pa	2	10	2	2	–	3	2	2	17	7	4	–	4	3
rk_th2	2	13	8	8	2	5	6	2	13	8	8	2	5	6
rk_ts	2	11	4	7	2	4	8	2	11	4	7	2	5	8
rsv	–	8	3	5	1	5	4	–	5	4	1	1	5	–
rsv_ss	–	12	4	8	1	6	7	–	7	6	3	2	7	1
rv10	1	5	1	1	–	1	1	1	5	1	1	–	1	1
rv10_ss	2	4	1	–	1	1	1	2	4	1	–	1	1	1
rv5	2	10	3	4	2	1	1	2	10	3	4	2	1	1
rv5_ss	1	9	6	3	2	5	3	1	9	5	4	2	5	3
Total	**12**	**90**	**43**	**44**	**12**	**38**	**35**	**13**	**115**	**51**	**44**	**14**	**40**	**33**
High volatility sample														
	$\hat{K} = 64$							$\hat{K} = 58$						
bv	–	20	9	12	3	2	8	–	28	14	25	3	4	5
bv_ss	1	24	13	10	3	5	13	2	32	18	18	3	7	8
medrv	–	–	–	–	–	–	-	1	–	5	–	–	1	–
rk_pa	–	2	3	4	–	–	1	–	62	23	42	–	18	1
rk_th2	1	8	5	4	1	4	3	1	8	5	4	1	4	3
rk_ts	1	8	4	3	2	3	4	1	8	4	3	2	3	4
rsv	1	9	9	8	1	5	2	2	9	10	4	1	5	–
rsv_ss	2	8	8	5	–	4	4	2	6	10	3	1	4	2
rv10	1	5	2	2	–	1	2	1	43	10	2	–	1	3
rv10_ss	1	3	2	2	–	–	2	1	42	8	2	–	–	3
rv5	1	1	3	1	–	2	1	1	40	20	1	–	2	1
rv5_ss	1	2	4	4	–	–	1	1	40	18	4	–	–	1
Total	**10**	**90**	**62**	**55**	**10**	**26**	**41**	**13**	**318**	**145**	**108**	**11**	**49**	**31**

Note: Each combination contains the following number of specifications: AR - 16, GA - 128, FI - 64, EG - 64, HA - 16, MI - 48, AP - 64.
Legend for the partition analysis: $\bar{k} = \hat{K}$ | $\bar{k} = \hat{K}/2$ | $k = 0$

References

Akyildirim, Erdinc, Goncu, Ahmet, Sensoy, Ahmet, 2021. Prediction of cryptocurrency returns using machine learning. Ann. Oper. Res. 297, 3–36.

Alonso-Monsalve, Saúl, Suárez-Cetrulo, Andrés L, Cervantes, Alejandro, Quintana, David, 2020. Convolution on neural networks for high-frequency trend prediction of cryptocurrency exchange rates using technical indicators. Expert Syst. Appl. 149, 113250.

Amado, Cristina, Teräsvirta, Timo, 2014. Conditional correlation models of autoregressive conditional heteroscedasticity with nonstationary GARCH equations. J. Bus. Econom. Statist. 32, 69–87.

Audrino, Francesco, Sigrist, Fabio, Ballinari, Daniele, 2020. The impact of sentiment and attention measures on stock market volatility. Int. J. Forecast. 36 (2), 334–357.

Baumeister, Christiane, Korobilis, Dimitris, Lee, Thomas K., 2022. Energy markets and global economic conditions. Rev. Econ. Stat. 104 (4), 828–844.

Bernardi, Mauro, Catania, Leopoldo, 2018. The model confidence set package for R. Int. J. Comput. Econ. Econ. 8 (2), 144–158.

Bessec, Marie, Fouquau, Julien, 2018. Short-run electricity load forecasting with combinations of stationary wavelet transforms. European J. Oper. Res. 264 (1), 149–164.

Bollerslev, Tim, 1990. Modelling the coherence in short-run nominal exchange rates: a multivariate generalized ARCH model. Rev. Econ. Stat. 498–505.

Boudt, Kris, Danielsson, Jon, Laurent, Sébastien, 2013. Robust forecasting of dynamic conditional correlation GARCH models. Int. J. Forecast. 29, 244–257.

Caporale, Guglielmo Maria, Zekokh, Timur, 2019. Modelling volatility of cryptocurrencies using Markov-switching GARCH models. Res. Int. Bus. Financ. 48, 143–155.

Caporin, Massimiliano, McAleer, Michael, 2014. Robust ranking of multivariate GARCH models by problem dimension. Comput. Statist. Data Anal. 76, 172–185.

Cappiello, Lorenzo, Engle, Robert F., Sheppard, Kevin, 2006. Asymmetric dynamics in the correlations of global equity and bond returns. J. Financ. Econ. 4 (4), 537–572.

Conrad, Christian, Kleen, Onno, 2020. Two are better than one: Volatility forecasting using multiplicative component GARCH-MIDAS models. J. Appl. Econometrics 35 (1), 19–45.

Corbet, Shaen, Lucey, Brian, Urquhart, Andrew, Yarovaya, Larisa, 2019. Cryptocurrencies as a financial asset: A systematic analysis. Int. Rev. Financ. Anal. 62, 182–199.

Corradi, Valentina, Swanson, Norman R., 2013. A survey of recent advances in forecast accuracy comparison testing, with an extension to stochastic dominance. In: Recent Advances and Future Directions in Causality, Prediction, and Specification Analysis. pp. 121–143.

Degiannakis, Stavros, Filis, George, 2017. Forecasting oil price realized volatility using information channels from other asset classes. J. Int. Money Financ. 76, 28–49.

Diebold, Francis X., Mariano, Roberto S., 1995. Comparing predictive accuracy. J. Bus. Econom. Statist. 13, 134–144.

Dumitrescu, Elena, Hué, Sullivan, Hurlin, Christophe, Tokpavi, Sessi, 2022. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. European J. Oper. Res. 297 (3), 1178–1192.

Engle, Robert, 2002. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. J. Bus. Econom. Statist. 20 (3), 339–350.

Ferrari, Davide, Yang, Yuhong, 2015. Confidence sets for model selection by F-testing. Statist. Sinica 1637–1658.
Hamid, Alain, 2014. Prediction power of high-frequency based volatility measures: a model based approach. Rev. Manag. Sci. 9.
Hansen, Peter Reinhard, 2005. A test for superior predictive ability. J. Bus. Econom. Statist. 23, 365–380.
Hansen, Peter R., Lunde, Asger, Nason, James M., 2011. The model confidence set. Econometrica 79, 453–497.
Hansen, Peter Reinhard, Lunde, Asger, Voev, Valeri, 2014. Realized beta GARCH: a multivariate GARCH model with realized measures of volatility. J. Appl. Econometrics 29, 774–799.
Heber, Gerd, Lunde, Asger, Shephard, Neil, Sheppard, Kevin, 2009. Oxford-man institute's realized library. Version 0.3, Oxford-Man Institute, University of Oxford.
Huang, Yumeng, Dai, Xingyu, Wang, Qunwei, Zhou, Dequn, 2021. A hybrid model for carbon price forecasting using GARCH and long short-term memory network. Appl. Energy 285, 116485.
Hurlin, Christophe, Laurent, Sébastien, Quaedvlieg, Rogier, Smeekes, Stephan, 2017. Risk measure inference. J. Bus. Econom. Statist. 35 (4), 499–512.
Iltuzer, Zeynep, Tas, Oktay, 2013. The forecasting performances of volatility models in emerging stock markets: Is a generalization really possible? J. Appl. Financ. Bank. 3, 49–73.
Kristjanpoller, Werner, Minutolo, Marcel C., 2016. Forecasting volatility of oil price using an artificial neural network-GARCH model. Expert Syst. Appl. 65, 233–241.
Laurent, Sébastien, Rombouts, Jeroen VK, Violante, Francesco, 2012. On the forecasting accuracy of multivariate GARCH models. J. Appl. Econometrics 27, 934–955.
Laurent, Sébastien, Rombouts, Jeroen VK, Violante, Francesco, 2013. On loss functions and ranking forecasting performances of multivariate volatility models. J. Econometrics 173, 1–10.
Liang, Chao, Umar, Muhammad, Ma, Feng, Huynh, Toan L.D., 2022. Climate policy uncertainty and world renewable energy index volatility forecasting. Technol. Forecast. Soc. Change 182, 121810.
Liu, Lily, Patton, Andrew J., Sheppard, Kevin, 2015. Does anything beat 5-minute RV? A comparison of realized measures across multiple asset classes. J. Econometrics 187, 293–311.
Liu, Jian, Zhang, Ziting, Yan, Lizhao, Wen, Fenghua, 2021. Forecasting the volatility of EUA futures with economic policy uncertainty using the GARCH-MIDAS model. Financ. Innov. 7, 1–19.
Ma, Feng, Liang, Chao, Ma, Yuanhui, Wahab, Mohamed Ismail Mohamed, 2020. Cryptocurrency volatility forecasting: A Markov regime-switching MIDAS approach. J. Forecast. 39 (8), 1277–1290.
Ma, Feng, Wahab, M.I.M., Zhang, Yaojie, 2019. Forecasting the US stock volatility: An aligned jump index from G7 stock markets. Pacific-Basin Financ. J. 54, 132–146.
Masini, Ricardo P., Medeiros, Marcelo C., Mendes, Eduardo F., 2023. Machine learning advances for time series forecasting. J. Econ. Surv. 37 (1), 76–111.
Neumann, Michael, Skiadopoulos, George, 2013. Predictable dynamics in higher-order risk-neutral moments: Evidence from the S&P 500 options. J. Financ. Quant. Anal. 48, 947–977.
Newell, Richard G., Prest, Brian C., Sexton, Steven E., 2021. The GDP-temperature relationship: implications for climate change damages. J. Environ. Econ. Manag. 108, 102445.
Patton, Andrew J., Sheppard, Kevin, 2009. In: Andersen, T.G., Davis, R.A., Kreiss, J.-P., Mikosch, Th.V (Eds.), Handbook of Financial Time Series. Oxford University Press.
Patton, Andrew J., 2011. Volatility forecast comparison using imperfect volatility proxies. J. Econometrics 160 (1), 246–256.
Politis, Dimitris N., Romano, Joseph P., 1994. The stationary bootstrap. J. Amer. Statist. Assoc. 89, 1303–1313.
Rendon-Sanchez, Juan F., de Menezes, Lilian M., 2019. Structural combination of seasonal exponential smoothing forecasts applied to load forecasting. European J. Oper. Res. 275 (3), 916–924.
Samuels, Jon D., Sekkel, Rodrigo M., 2017. Model confidence sets and forecast combination. Int. J. Forecast. 33 (1), 48–60.
Walther, Thomas, Klein, Tony, Bouri, Elie, 2019. Exogenous drivers of bitcoin and cryptocurrency volatility–A mixed data sampling approach to forecasting. J. Int. Financ. Mark. Institutions Money 63, 101133.
Wei, Yu, Liu, Jing, Lai, Xiaodong, Hu, Yang, 2017. Which determinant is the most informative in forecasting crude oil market volatility: Fundamental, speculation, or uncertainty? Energy Econ. 68, 141–150.
White, Halbert, 2000. A reality check for data snooping. Econometrica 68, 1097–1126.
Wilhelmsson, Anders, 2013. Density forecasting with time-varying higher moments: A model confidence set approach. J. Forecast. 32, 19–31.