A GENERAL FRAMEWORK FOR MULTI-STEP AHEAD ADAPTIVE CONFORMAL HETEROSCEDASTIC TIME SERIES FORECASTING *

Martim Sousa  
IEETA/DETI, University of Aveiro  
martimsousa@ua.pt

Ana Maria Tomé  
IEETA/DETI, University of Aveiro  
a@ua.pt

José Moreira  
IEETA/DETI, University of Aveiro  
jo@ua.pt

ABSTRACT

The exponential growth of machine learning (ML) has prompted a great deal of interest in quantifying the uncertainty of each prediction for a user-defined level of confidence since nowadays ML is increasingly being used in high-stakes settings. Reliable ML via prediction intervals (PIs) that take into account jointly the epistemic and aleatory uncertainty is therefore imperative and is a step towards increased trust in model’s forecasts. Conformal prediction (CP) is a lightweight distribution-free uncertainty quantification framework that works for any black-box model, yielding PIs that are valid under the mild assumption of exchangeability. CP-type methods are gaining popularity due to being easy to implement and computationally cheap; however, the exchangeability assumption immediately excludes time series forecasting from the stage. Although recent papers tackle distribution shift and asymptotic versions of CP, this is not enough for the general time series forecasting problem of producing H-step ahead valid PIs. To attain such a goal, we propose a new method called AEnbMIMOCQR (Adaptive ensemble batch multi-input multi-output conformalized quantile regression), which produces valid PIs asymptotically and is appropriate for heteroscedastic time series. We compare the proposed method against state-of-the-art competitive methods in the NN5 forecasting competition dataset. All the code and data to reproduce the experiments are made available.

Keywords Conformal prediction · Conformalized quantile regression · Conformal time series forecasting · Distribution shift · Multi-step ahead forecasting

1 Introduction

In a time series forecasting problem, relying on past observations \([x_{n-d+1}, \ldots, x_n] \) the goal is to predict the next observations \([x_{n+1}, x_{n+2}, \ldots, x_{n+H}] \), where H is the number of steps to be predicted, often referred to as the horizon; and d is the number of lags. Unfortunately, when it comes to practical deployments, the forecasts \([\hat{x}_{n+1}, \hat{x}_{n+2}, \ldots, \hat{x}_{n+H}] \) have an unknown associated uncertainty given by

\[
\epsilon_{n+h} = x_{n+h} - \hat{x}_{n+h}, \quad \forall h \in \{1, 2, \ldots, H\}.
\]  

*Remark: This is a preprint whose final form is not yet published.
Therefore, throughout this article, we focus on building PIs around each forecast that faithfully assesses the said uncertainty of the ML model for a user-defined level of confidence. Table (1) contains the notation used throughout this article.

| Notation                  | Meaning                                           |
|---------------------------|---------------------------------------------------|
| \( \alpha \)             | Miscoverage rate \( \alpha \)                     |
| \( \epsilon \)           | set of non-conformity scores                       |
| \( \text{Quantile}(\epsilon; 1 - \alpha) \) | 1 - \( \alpha \) order quantile of \( \epsilon \) |
| \( \hat{C}_{1-\alpha}(x_n) \) | 1 - \( \alpha \) PI on input \( x_n \)           |
| \( \hat{C}_{1-\alpha}(x_n) \) t | 1 - \( \alpha \) PI on input \( x_n \) and timestep \( t \) |
| \( x \)                   | a scalar                                          |
| \( x_{1:d} \)             | Compact version of \( (x_1, \ldots, x_d) \)      |
| \( x_{1:d} \)             | Compact version of \( (x_1, \ldots, x_d) \)      |
| \( X \)                   | a matrix or a random variable (easily distinguished) |
| \( X \)                   | a tensor or a random vector (easily distinguished)  |

Table 1: Notation used throughout this article.

Inductive conformal prediction (ICP) is a distribution-free uncertainty quantification technique that aims to give valid PIs in the sense of Theorem (1.1) [1, 2, 3, 4, 5]. Unlike full conformal prediction [2], ICP does not require retraining the model at the cost of lower statistical efficiency. ICP, also known as split conformal prediction, uses a calibration set \( D_{cal} \), not used during training, to compute \( \epsilon = \{(\epsilon_i)\}_{i=1}^{n} \) non-conformity scores. The aforementioned scores are calculated using a non-conformity score function \( s(x, y) \), which works as a heuristic notion of uncertainty, where larger scores encode worse agreement between the input \( x \) and true output \( y \). Afterward, we compute \( \hat{q} = \text{Quantile}(\epsilon; 1 - \alpha) \) and rely on this information to build PIs on new data as

\[
\mathbb{P}\{y_{n+1} \in \hat{C}_{1-\alpha}(x_{n+1})\} = \mathbb{P}\{y \in \mathcal{Y} : s(x_{n+1}, y) \leq \hat{q}\}.
\]

Even though ICP works for any non-conformity score function \( s \), as stated in Theorem (1.1), the choice of the non-conformity score function has a direct impact on the PIs width. From a practical point of view, we want valid PIs with the shortest possible width to be informative in common decision-making tasks. Ideally, we also seek conditional coverage given by Eq.(4). Obviously, conditional coverage implies marginal coverage, while the reciprocal is not true. Conditional coverage is proved impossible with ICP; however, we can heuristically approximate it. Note that conditional coverage is an important property as we want PIs to adapt to heteroscedasticity.

**Definition 1** (Exchangeability). A sequence of random variables \( Z_1, Z_2, \ldots, Z_n \in \mathcal{Z} \) are exchangeable if and only if for any permutation \( \pi : \{1, 2, \ldots, n\} \rightarrow \{1, 2, \ldots, n\} \) and every measurable set \( E \subseteq \mathcal{Z}^n \), we have

\[
\mathbb{P}\{(Z_1, Z_2, \ldots, Z_n) \in E\} = \mathbb{P}\{(Z_{\pi(1)}, Z_{\pi(2)}, \ldots, Z_{\pi(n)}) \in E\}
\]

(2)

**Theorem 1.1** (Marginal coverage guarantee). Let \( (X_1, Y_1), \ldots, (X_n, Y_n) \) be exchangeable random vectors with no ties almost surely drawn from a distribution \( P \), additionally if for a new pair \( (X_{n+1}, Y_{n+1}), (X_1, Y_1), \ldots, (X_n, Y_n) \) are still exchangeable, then by constructing \( C_{1-\alpha}(X_{n+1}) \) using ICP, the following inequality holds for any non-conformity score function \( s : X \times \mathcal{Y} \rightarrow \mathcal{A} \subseteq \mathbb{R} \) and any \( \alpha \in (\frac{1}{n+1}, 1) \)

\[
1 - \alpha \leq \mathbb{P}\{Y_{n+1} \in C_{1-\alpha}(X_{n+1})\} \leq 1 - \alpha + \frac{1}{n+1}.
\]

(3)

**Proof.** See [1].

**Definition 2** (Conditional coverage). An ICP procedure guarantees conditional coverage if

\[
\mathbb{P}\{Y_{n+1} \in C_{1-\alpha}(X_{n+1}) | X = x_{n+1}\} \geq 1 - \alpha.
\]

(4)

Recalling our time series forecasting problem, we easily verify that ICP methods cannot be applied directly to time series because the exchangeability assumption does not hold, the order of observations matter and distribution shifts over time. Therefore, we will rather focus on an online conformal setting as in [6, 7, 8], achieving asymptotic validity.
**Definition 3** (Asymptotic valid PIs). Let \( \hat{C}_{1-\alpha}^{(t)}(x_{n+h}) \) be the PI for \( x_{n+h} \) with \( 1 - \alpha \) confidence at timestep \( t \). These PIs are said asymptotic valid if

\[
\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbb{1}\{ x_{n+h} \in \hat{C}_{1-\alpha}^{(t)}(x_{n+h}) \} = 1 - \alpha, \quad \forall h \in \{1, 2, ..., H\},
\]

where \( \hat{C}_{1-\alpha}^{(t)}(x_{n+h}) = [\hat{x}^L_{n+h}, \hat{x}^U_{n+h}] \), with \( \hat{x}^L_{n+h} \) being a lower bound for \( x_{n+h} \) and \( \hat{x}^U_{n+h} \) an upper bound; \( \mathbb{1} \) denotes the indicator function.

**Article outline**

The article outline is as follows. Section 2 reviews two forecasting strategies that we will utilize, one for single-output models; and another for multi-output ones. Section 3 provides an overview of related work which inspired the proposed method. Section 4 introduces our approach called AEnbMIMOQQR. Section 5 compares the effectiveness of our method against state-of-the-art online conformal frameworks in the NN5 forecasting competition dataset. Finally, Section 6 draws main conclusions.

**2 Forecasting strategies**

Whenever dealing with an H-step ahead forecasting problem, we must decide which forecasting strategy to use. Although this is a vast active research topic that goes beyond the scope of this article, we will introduce two well-known methods for this purpose: the recursive strategy; and the multi-input multi-output (MIMO) to forecast each observation of the desired horizon \([9, 10]\).

**2.1 Recursive strategy**

Let \( \hat{f} : \mathbb{R}^d \to \mathbb{R} \) denote a ML model that predicts the next observation based on \( d \) lags. In the recursive strategy, each forecast of the horizon \( H \) is computed through the following equations

\[
\hat{x}_{n+h} = \begin{cases} 
\hat{f}(x_{n-d+1}, ..., x_n), & \text{for } h = 1, \\
\hat{f}(x_{n-d+h}, ..., x_n, \hat{x}_{n+1}, ..., \hat{x}_{n+h-1}), & \text{for } h \in \{2, ..., d\}, \\
\hat{f}(\hat{x}_{n+h-d}, ..., \hat{x}_{n+h-1}), & \text{for } h \in \{d+1, ..., H\}.
\end{cases}
\]

The major shortcoming of this approach is that the trained ML model \( \hat{f} \) is trained on \( d \) actual lag values, yet in the out-of-sample phase, previous forecasts are included as input, hence accumulating errors as \( h \) increases. Note that for \( h > d \) the input only contains forecasts.

**2.2 MIMO**

In the MIMO strategy, the trained ML model \( \hat{F} : \mathbb{R}^d \to \mathbb{R}^H \) directly learned to forecast each observation of the horizon in the training phase. Therefore, this strategy has no accumulation of errors, each observation of the horizon is predicted in a single step as

\[
\hat{F}(x_{n-d+1}, ..., x_n) = [\hat{x}_{n+1}, ..., \hat{x}_{n+H}].
\]

However, despite this appeal, compared to the recursive strategy, the underlying structure of MIMO is way more complex and thus harder to learn. Furthermore, few models besides neural networks can cope with multi-output regression tasks as most of the ML models are optimized for a single target. Additionally, the stochastic dependence between the observations of the horizon is lost.

**3 Related work**

In this section, we do a brief recap on competitive methods for probabilistic forecasting that inspired our method and address their drawbacks.
3.1 Probabilistic forecasting with ARIMA

ARIMA\((d, d^*, q)\) models are still among the most robust and best performing methods for time series forecasting [11]. PIs can be obtained from ARIMA\((d, d^*, q)\) models as

\[
\hat{C}_{1-\alpha}(x_{n+h}) = [\hat{x}_{n+h} - c_{1-\alpha}\hat{\sigma}_h, \hat{x}_{n+h} + c_{1-\alpha}\hat{\sigma}_h], \quad \forall h \in \{1, 2, ..., H\},
\]

where \(\hat{x}_{n+h}\) is the \(h\)-step ahead point prediction; \(c_{1-\alpha}\) is a constant whose value is the inverse cumulative distribution of the standard normal distribution calculated on \(1 - \alpha\); and \(\hat{\sigma}_h\) is the \(h\)-step ahead estimation of the standard deviation.

However, the standard deviation estimation is obtained from the model residuals, which are assumed to be serially uncorrelated and normally distributed. If both assumptions are not met, these PIs become unreliable. Moreover, \(\hat{\sigma}_h\) is optimized for marginal coverage, hence it does not change from the joint input (lags).

Hyndman & Athanasopoulos [12] leave an interesting remark that we want to call the reader’s attention to. If \(d^* = 0\), then the time series is deemed stationary, and thus \(\hat{\sigma}_h\) does not change for every \(h \in \{1, ..., H\}\), while if \(d^* \geq 1\), it has tendency and therefore PIs width increase in \(h \in \{1, ..., H\}\).

3.2 EnbPI algorithm

The EnbPI algorithm, summarized in Algorithm (1), is a framework gaining popularity that achieves asymptotic validity and does not require data splitting unlike other ICP methods [13]. It assumes that the samples \((x_{(n_t-d+1):n_t}, x_{n_t+1})\) come from the following process

\[
x_{n_t+1} = f(x_{(n_t-d+1):n_t}) + \epsilon_t, \quad t = 1, 2, ...
\]

where \(n_t = d + t - 1\).

The goal is to approximate the real unknown function \(f : \mathbb{R}^d \to \mathbb{R}\) with a homogeneous ensemble estimator \(\phi(\{\hat{f}_b(x_{(n_t-d+1):n_t})\}_{b=1}^B)\), where \(\phi\) is an aggregation function such as the mean, median, trimmed mean, etc and each model \(\hat{f}_b\) is trained on a bootstrap dataset. After the training phase, the method iterates through every row of the training dataset and each \(t\)-th row not used for training by at least one of the \(B\) bootstrap models \(\hat{f}_b\) generates an in-sample non-conformity score: 

\[
\epsilon^\phi_t = |\hat{x}_{n_t+1} - x_{n_t+1}|, \quad \text{where} \quad \hat{x}_{n_t+1} = \phi(\{\hat{f}_b(x_{(n_t-d+1):n_t}) \mid t \notin S_b\}),
\]

with \(S_b\) being the bootstrap index set, i.e., the ensemble is made up by those models who did not use the \(t\)-th row during training. This in-sample non-conformity score \(\epsilon^\phi_t\) is then added to a non-conformity set \(\epsilon\). Therefore, after computing \(\hat{q} = \text{Quantile} (\epsilon; 1 - \alpha)\) as in any ICP method, PIs come in the following form

\[
\hat{C}^{(t)}_{1-\alpha}(x_{n_t+1}) = [\hat{x}_{n_t+1} - \hat{q}, \hat{x}_{n_t+1} + \hat{q}],
\]

where \(\hat{x}_{n_t+1} = \phi(\{\hat{f}_b(x_{(n_t-d+1):n_t})\}_{b=1}^B)\).

Up to this point, this method seems not to differ much from ICP regression methods. The cleverness of this algorithm lies within the batch size \(H\) that makes this method dynamic and asymptotic valid. The non-conformity set \(\epsilon\) works as a sliding window, after \(H\)-step ahead forecasts, the first \(H\) non-conformity scores are discarded, and the most recent \(H\) are added, thus making it dynamic. The batch size \(H\) controls how fast the algorithm adapts to new information. Since the problem at hand is an \(H\)-step ahead forecasting problem, we assume that the ground truth values are only known after performing \(H\) 1-step ahead forecasts.
We further explore the theoretical analysis of EnbPI to explain why despite being asymptotic valid, as shown in [13], it may take a long time before EnbPI starts producing reliable PIs for time series.

**Lemma 3.1.** Let \( B \) be the number of bootstrap models of EnbPI with \( T \) training samples, then the length of the non-conformity set \( \epsilon \) is equal to \( T \left( 1 - (1 - (1 - \frac{1}{p})^T)^B \right) \) on average.

**Proof.** We know that the probability of an index not belonging to a bootstrap index set is equal to

\[
P(i \not\in S_b) = \left(1 - \frac{1}{T}\right)^T.
\]

Let \( X_i \) be the distribution that represents the number of bootstrap models \( \hat{f}^b \) that did not use the \( i \)-th row during training. We know that \( X_i \sim Bin(B, (1 - \frac{1}{T})^T) \) and so

\[
P(X_i \geq 1) = 1 - P(X_i = 0) = 1 - \left(1 - (1 - \frac{1}{T})^T\right)^B.
\]

From here, follows that the length distribution of \( \epsilon \) denoted by \( L \sim Bin(T, 1 - (1 - (1 - \frac{1}{p})^T)^B) \) and therefore

\[
E(L) = T \left(1 - \left(1 - (1 - \frac{1}{T})^T\right)^B\right).
\]
Lemma 3.2. Let $N = |\epsilon|$ be the non-conformity set length. If we use the recursive strategy, then only after $\max\{N, d\} + k$ forecasting steps the non-conformity set $\epsilon$ does not contain any in-sample non-conformity score and contains at least a cycle of accumulation of errors associated with the recursive strategy, where $k = 0$ if $\max\{N, d\} \mod H = 0$ and $k = H - (\max\{N, d\} \mod H)$ otherwise.

Proof. Follows immediately from the recursive strategy definition (see (2.1)) and Algorithm (1).

Lemmas (3.1) and (3.2) bring important insights to explain why the EnbPI algorithm has some pitfalls for time series PIs, despite being asymptotically valid. By straightforward application of Lemma (3.2), only after $\max\{\epsilon, d\} + k$ forecasting steps the non-conformity set is adapted to the accumulation of errors associated with the recursive strategy and none in-sample non-conformity score is in the non-conformity set $\epsilon$, where $k = \max\{N, d\} \mod H$. In essence, EnbPI computes in-sample non-conformity scores as (line 8 of Algorithm (1))

$$
\epsilon_i^\phi = [x_{n_i+1} - \hat{x}_{n_i+1}].
$$

(13)

However, these in-sample non-conformity scores are computed using $d$ actual lags as joint input and thus it does not faithfully reflect the true forecast uncertainty since in the out-of-sample phase the joint input incorporates lags that are previous forecasts and therefore uncertain. Lemma (3.2) provides what can be considered a warm-up period before EnbPI produces reliable PIs for time series forecasting.

Caveats and tips

- If the non-conformity set ($\epsilon$) length is too high it is possible to make it more adaptable to distribution shifts by sampling without replacement a fraction of observations from $\epsilon$ at the cost of statistical efficiency.
- EnbPI is not adaptive to heteroscedasticity, i.e., PIs width is always equal to $2\hat{q}$ within the same batch.

3.3 Conformalized quantile regression

Conformalized quantile regression (CQR) is a procedure that inherits the advantages of quantile regression (QR) and CP [1, 14, 15, 16]. The idea is to use QR to get an estimation of the true unknown conditional quantiles from the data and then apply CP to ensure marginal coverage in finite samples. Consequently, we get PIs that are adaptive to heteroscedasticity due to QR, i.e., PIs vary accordingly to the input $x$ and also marginal coverage because of CP.

Recall that the $\tau$-quantile of a conditional distribution $Y|X$ calculated on $X = x$ is given by

$$
Q_\tau(x) = \inf \{y \in \mathcal{Y} : F_{Y|X} (y|X = x) \geq \tau \}. \tag{14}
$$

Likewise, if $F_{Y|X} (y|X = x)$ has an inverse since it is an increasing function, we also get

$$
Q_\tau(x) = F_{Y|X}^{-1}(\tau|X = x) \tag{15}
$$

QR aims to approximate $Q_{\alpha/2}(x)$ and $Q_{1-\alpha/2}(x)$ to get a lower and upper bound, respectively, that approximate conditional coverage. This is achieved by simply replacing the common squared error loss with a quantile loss, also known as pinball loss formulated as

$$
\rho_\tau(y, f(x)) = \max\left(\tau(y - f(x)), (\tau - 1)(y - f(x))\right), \tag{16}
$$

Hence, we attempt to minimize the following objective

$$
\min\sum_{i=1}^{n_{\text{train}}} \rho_\tau(y_i, \hat{f}(x_i)) + \Omega(\theta), \tag{17}
$$

where $\Omega(\theta)$ is a potential regularizer; and $\hat{f} : \mathbb{R}^d \rightarrow \mathbb{R}$ any black-box regression model.

However, in finite samples, we only get estimates $\hat{Q}_{\alpha/2}(x)$ and $\hat{Q}_{1-\alpha/2}(x)$ after minimizing Eq.(17) that frequently do not attain marginal coverage. Therefore, (Romano et al., 2019) [14] had the idea of applying CP on top of QR via the following non-conformity score function

$$
s(x, y) = \max\{\hat{Q}_{\alpha/2}(x) - y, y - \hat{Q}_{1-\alpha/2}(x)\}. \tag{18}
$$
Subsequently, relying on the aforementioned \textit{non-conformity score function}, we compute
\[ \hat{q} = \text{Quantile} \left( \epsilon; 1 - \alpha \right), \]  
(19)
on a calibration set to deliver PIs as
\[ \left[ Q_{\alpha/2}(x) - \hat{q}, Q_{1-\alpha/2}(x) + \hat{q} \right], \]  
(20)
which attain marginal coverage and approximately conditional coverage.

### 3.4 EnbCQR algorithm

EnbCQR \cite{17} is a slight yet important modification of the EnbPI algorithm (recall Algorithm (1)). The unique difference is that EnbCQR uses CQR instead of using absolute errors as the \textit{non-conformity score function}, thereby making the method adaptable to heteroscedastic time series. Not replicated here in the interest of space.

### 3.5 Adaptive conformal inference

The major obstacle hindering the validity of the Theorem (1.1) is when exchangeability is broken. The main causes include covariate shift and distribution shift. That is, the distribution of the calibration dataset might be significantly altered over time, making PIs increasingly unreliable in terms of coverage as time moves forward. Many strategies \cite{18,19,20} have been proposed to tackle both covariate and distribution shift in a weighted fashion; however, the most versatile technique is adaptive conformal inference (ACI) \cite{7}.

The intuition behind ACI is easy to grasp. Instead of having a fixed \( \alpha \), ACI uses a time-varying \((\alpha_t)_{t \in \mathbb{N}}\) updated via the following equation
\[ \alpha_{t+1} = \alpha_t + \gamma(\alpha_t - 1 \{ y_t \notin \hat{C}_{1-\alpha_t}(x_t) \}). \]  
(21)
ACI starts with \( \alpha_1 = \alpha \), and \( \gamma \) works as a learning rate that controls how fast we desire the conformal method to adapt to distribution shifts. As shown in \cite{7}, ACI provides asymptotic validity (see Definition (3)) since \( \lim_{t \to \infty} \{ y_t \notin \hat{C}_{1-\alpha_t}(x_t) \} = \alpha \). Although the choice of \( \gamma \) was improved in \cite{8}, the minor problem of this method persists: \( \alpha_t \in [-\gamma, 1 + \gamma] \), and so nothing prevents \( \alpha_t < 0 \) or \( \alpha_t > 1 \), despite rarely happening in practice for a small \( \gamma \).

### 3.6 MIMOCQR algorithm

In a slightly different direction, several papers suggest grouping observations in a MIMO block structure which preserves dependency as \( B_t = (x_{(n_t-d+1):n_t}, x_{(n_t+1):n_t+H}) \) \cite{21,22}. Therefore, under the exchangeability assumption of those blocks (recall Definition (1)) it is possible to apply ICP. Algorithm (2) encompasses every step to do so for \textit{heteroscedastic} time series.

Suppose that \( \hat{F}_{\alpha/2} : \mathbb{R}^d \to \mathbb{R}^H \) and \( \hat{F}_{1-\alpha/2} : \mathbb{R}^d \to \mathbb{R}^H \) are two multi-output QR models that yield a multi-output lower bound and upper bound, respectively. Since QR does not ensure coverage in finite samples, to achieve it in the finite case, the following vector \textit{non-conformity score function} version of Eq.(18) must be utilized
\[ s(x_{(n_t-d+1):n_t}, x_{(n_t+1):n_t+H}) = \max \left\{ \hat{F}_{\alpha/2}(x_{(n_t-d+1):n_t}) - x_{(n_t+1):n_t+H}, x_{(n_t+1):n_t+H} - \hat{F}_{1-\alpha/2}(x_{(n_t-d+1):n_t}) \right\}. \]  
(22)
Note that Eq.(22) yields a vector of residuals \([\epsilon_1, ..., \epsilon_H]\), where \( \epsilon_i \) is associated with the \( i \)-th step ahead prediction interval. After applying Eq.(22) on a holdout set, to ensure \( 1 - \alpha \) coverage on each h-step ahead prediction interval, then we have to compute a different \( \hat{q} \) per horizon element as
\[ \hat{q}^{(h)} = \text{Quantile} \left( \epsilon_h; 1 - \alpha \right), \ \forall h \in \{1, ..., H\}, \]  
(23)
where \( \epsilon_h \) is the \textit{non-conformity score set} of the h-step ahead PIs. Subsequently, calibrated \textit{heteroscedastic} PIs with finite sample guarantees come as
\[ \left[ \hat{F}_{\alpha/2}(x_{(n_t-d+1):n_t}) - \hat{q}, \hat{F}_{1-\alpha/2}(x_{(n_t-d+1):n_t}) + \hat{q} \right], \]  
(24)
where \( \hat{q} = \bigcup_{h=1}^H \{ \hat{q}^{(h)} \} \).
This method circumvents the accumulation of errors associated with the recursive EnbPI and EnbCQR by employing the MIMO strategy; however, the distribution of blocks usually shifts over time, culminating in decreasing coverage due to
exchangeability violation. Therefore, it is essential to include some ideas drawn from dynamic methods such as EnbPI, EnbCQR, and ACI.

Algorithm 2 Multi-input multi-output conformalized quantile regression (MIMOCQR)

**Input:** A training set \( \{(B_i := x_{(n_i-d+1:n_i)}, x_{(n_i+1:(n_i+H))})\}_{i=1}^T \), miscoverage rate \( \alpha \), a multi-output QR model \( \hat{F}_t : \mathbb{R}^d \rightarrow \mathbb{R}^H \), a test set \( \{(x_{(n_t-d+1:n_t)}, x_{(n_t+1:(n_t+H))})\}_{t=T+1}^T \) with \( x_{(n_t+1:(n_t+H))} \) revealed only after \( H \) timesteps.

**Output:** PIs: \( \{\hat{C}_{1-\alpha}^t(x_{(n_t+1:(n_t+H))})\}_{t=T+1}^T \)

1. Sample an index set \( S_b = \{(i_1, ..., i_T)\} \) from indices (1,...,T)
2. Train \( \hat{F}_{\alpha/2}, \hat{F}_{1-\alpha/2} \) ← \( \hat{F}_{\alpha/2}((B_i) \mid i \in S_b), \hat{F}_{1-\alpha/2}((B_i) \mid i \in S_b)) \)
3. \( \hat{e}_h \leftarrow \{} \), \( \forall h \in \{1, ..., H\} \)
4. **for** \( i \leftarrow 1, ..., T \) **do**
5. **if** \( i \notin S_b \) **then**
6. \( [\hat{x}_{n_i+1}, ..., \hat{x}_{n_i+H}] \leftarrow \hat{F}_{\alpha/2}(x_{n_i-d+1}, ..., x_{n_i}) \)
7. \( [\hat{x}_{n_i+1}^U, ..., \hat{x}_{n_i+H}^U] \leftarrow \hat{F}_{1-\alpha/2}(x_{n_i-d+1}, ..., x_{n_i}) \)
8. Compute \( \hat{e}_h \leftarrow \hat{e}_h \cup \{\max\{\hat{x}_{n_i+h}^L - x_{n_t+h}, x_{n_i+h} - \hat{x}_{n_i+h}^U\}\}, \forall h \in \{1, ..., H\} \)
9. **end if**
10. **end for**
11. Compute \( \hat{q}^{(h)} \leftarrow \text{Quantile}(\hat{e}_h; 1-\alpha), \forall h \in \{1, ..., H\} \)
12. **for** \( t \leftarrow T+1, ..., T+T_1 \) **do**
13. \( [\hat{x}_{n_t+1}^L, ..., \hat{x}_{n_t+H}] \leftarrow \hat{F}_{\alpha/2}(x_{n_t-d+1}, ..., x_{n_t}) \)
14. \( [\hat{x}_{n_t+1}^U, ..., \hat{x}_{n_t+H}^U] \leftarrow \hat{F}_{1-\alpha/2}(x_{n_t-d+1}, ..., x_{n_t}) \)
15. Return \( \hat{C}_{1-\alpha}^t(x_{n_t+h}) \leftarrow [\hat{x}_{n_t+h}^L - \hat{q}^{(h)}, \hat{x}_{n_t+h}^U + \hat{q}^{(h)}], \forall h \in \{1, ..., H\} \)
16. **end for**

4 AEbMIMOCQR algorithm

In most cases, the distribution of blocks \( \{(B_i := x_{(n_i-d+1:n_i)}, x_{(n_i+1:(n_i+H))})\}_{i=1}^T \) proposed in Algorithm (2) are not i.i.d. nor exchangeable. That is, \( \exists \pi: \{1,2, ..., T\} \rightarrow \{1,2, ..., T\} \) such that

\[
\mathbb{P}\{(B_1, B_2, ..., B_T) \in E\} \neq \mathbb{P}\{(B_{\pi(1)}, B_{\pi(2)}, ..., B_{\pi(T)}) \in E\}. \tag{25}
\]

Hence, in such cases, MIMOCQR will not provide \( 1-\alpha \) coverage in the long run. Therefore, it is key to draw some ideas from adaptive and dynamic methods to make it asymptotic valid. The essence of AEbMIMOCQR is to improve Algorithm (2) by applying the following: a *homogeneous ensemble learner*; a sliding window of the *non-conformity scores* as in EnbPI and EnbCQR; sampling without replacement the in-sample *non-conformity scores* to speed up convergence; and an adaptive scheme with a time-varying \( (\alpha_t)_{t \in \mathbb{N}} \) as in ACI. Algorithm (3) encompasses every step highlighted here in a comprehensive and detailed manner. Additionally, the benefits of AEbMIMOCQR over other probabilistic forecasting methods are depicted in Table (2).

| Method     | Adaptive | Dynamic | non-conformity set | Multi-output | Heteroscedastic |
|------------|----------|---------|--------------------|--------------|-----------------|
| ARIMA      | ×        | ×       | ×                  | ×            | ×               |
| EnbPI      | ×        | ×       | ×                  | ×            | ×               |
| EnbCQR     | ×        | ×       | ×                  | ×            | ✓               |
| MIMOCQR    | ×        | ×       | ✓                  | ✓            | ✓               |
| AEbMIMOCQR | ✓        | ✓       | ✓                  | ✓            | ✓               |

Table 2: Summary of the probabilistic forecasting methods characteristics.
Algorithm 3 Adaptive ensemble batch multi-input multi-output conformalized quantile regression (AEEnbMIMOCQR)

Input: A training set \( \{(B_i := (x_{(n_i-d+1):n_i}, x_{(n_i+1):(n_i+H)}))\}_{i=1}^T \), miscoverage rate \( \alpha \), a multi-output QR model \( \hat{F}_x : \mathbb{R}^d \rightarrow \mathbb{R}^H \), an aggregation function \( \phi \), number of bootstrap models \( B \), \( n \geq H \) observations to sample without replacement, a test set \( \{(x_{(n_i-d+1):n_i}, x_{(n_i+1):(n_i+H)})\}_{i=T+1}^{T+T_1} \) with \( x_{(n_i+1):(n_i+H)} \) revealed only after \( H \) timesteps.

Output: PLs \( \hat{C}^{\text{(l)}}_{1-\alpha}(x_{(n_i+1):(n_i+H)}) \) \( T+T_1 \)

1: for \( b \leftarrow 1, \ldots, B \) do
2: \hspace{1em} Sample an index set \( S_b = (i_1, \ldots, i_T) \) from indices \( (1, \ldots, T) \)
3: \hspace{1em} \text{Train } [\hat{F}^b_{\alpha/2}, \hat{F}^b_{1-\alpha/2}] \leftarrow (\hat{F}^b_{\alpha/2}(\{(B_i) | i \in S_b\}), \hat{F}^{1-\alpha/2}_{\alpha/2}(\{(B_i) | i \in S_b\}))
4: end for
5: \( \epsilon_h \leftarrow \{\} \), \( \forall h \in \{1, \ldots, H\} \)
6: \( \alpha_h \leftarrow \alpha , \forall h \in \{1, \ldots, H\} \)
7: for \( i \leftarrow 1, \ldots, T \) do
8: \hspace{1em} \( [\hat{x}^L_{n_i+1}, \ldots, \hat{x}^L_{n_i+H}] \leftarrow \phi(\{(\hat{F}^b_{\alpha/2}(x_{(n_i-d+1):n_i})) | i \notin S_b\}) \)
9: \hspace{1em} \( [\hat{x}^U_{n_i+1}, \ldots, \hat{x}^U_{n_i+H}] \leftarrow \phi(\{(\hat{F}^{1-\alpha/2}_{\alpha/2}(x_{(n_i-d+1):n_i})) | i \notin S_b\}) \)
10: \hspace{1em} \( \epsilon_h \leftarrow \max(\{\hat{x}^L_{n_i+h} - x_{n_i+h}, x_{n_i+h} - \hat{x}^U_{n_i+h}\}, \forall h \in \{1, \ldots, H\} \)
11: \hspace{1em} \( \epsilon_h \leftarrow \epsilon_h \cup \{\epsilon_h^\text{(l)}\}, \forall h \in \{1, \ldots, H\} \)
12: end for
13: Sample without replacement \( \max(n, \lceil \epsilon_1 \rceil) \) elements from \( \epsilon_h \), \( \forall h \in \{1, \ldots, H\} \)
14: \( \hat{q}^{(h)} \leftarrow \text{Quantile}(\epsilon_h; 1 - \alpha_h), \forall h \in \{1, \ldots, H\} \)
15: \( \gamma \leftarrow 1/n \)
16: for \( t \leftarrow T + 1, \ldots, T + T_1 \) do
17: \hspace{1em} \( [\hat{x}^L_{n_i+1}, \ldots, \hat{x}^L_{n_i+H}] \leftarrow \phi(\{(\hat{F}^b_{\alpha/2}(x_{(n_i-d+1):n_i})) | b \in B\}) \)
18: \hspace{1em} \( [\hat{x}^U_{n_i+1}, \ldots, \hat{x}^U_{n_i+H}] \leftarrow \phi(\{(\hat{F}^{1-\alpha/2}_{\alpha/2}(x_{(n_i-d+1):n_i})) | b \in B\}) \)
19: \hspace{1em} \text{Return } \hat{C}_{1-\alpha}(x_{n_i+h}) \leftarrow [\hat{x}^L_{n_i+h} - \hat{q}^{(h)}(\hat{x}^U_{n_i+h} + \hat{q}^{(h)}(\hat{x}^U_{n_i+h} + \hat{q}^{(h)}))]\), \( \forall h \in \{1, \ldots, H\} \)
20: \hspace{1em} if \( t \equiv 0 \text{ mod } H \) then
21: \hspace{2em} for \( j \leftarrow t - H, \ldots, t - 1 \) do
22: \hspace{3em} \( \epsilon_h^{(s)} \leftarrow \max(\{\hat{x}^L_{n_i+h} - \hat{q}^{(h)} - x_{n_i+h}, x_{n_i+h} - (\hat{x}^U_{n_i+h} + \hat{q}^{(h)})\}, \forall h \in \{1, \ldots, H\} \)
23: \hspace{3em} \( \alpha_h \leftarrow \alpha_h + \gamma(\epsilon_h^{(s)} - 1 \{x_{n_i+h} \notin \hat{C}_{1-\alpha}^{(j)}(x_{n_i+h})\}), \forall h \in \{1, \ldots, H\} \)
24: \hspace{3em} \( \alpha_h \leftarrow \max(0, \min(\alpha_h, 1)), \forall h \in \{1, \ldots, H\} \)
25: \hspace{3em} \( \epsilon_h \leftarrow (\epsilon_h - \{\epsilon_h^\text{(s)}\}) \cup \{\epsilon_h^\text{(s)}\}, \forall h \in \{1, \ldots, H\} \) and reset index of \( \epsilon_h \)
26: \hspace{2em} end for
27: \hspace{2em} Update \( \hat{q}^{(h)} \leftarrow \text{Quantile}(\epsilon_h; 1 - \alpha_h), \forall h \in \{1, \ldots, H\} \)
28: \hspace{1em} end if
29: end for

4.1 Generalization to multivariate time series

So far, this article has solely tackled conformal univariate time series forecasting; however, in many applications, there are also covariates beyond past observations of the target variable. Recurrent neural networks (RNNs) such as the long-short term memory (LSTM) [23] can be trained in a tensor \( \mathbf{X} = \{(x_{(n_i-d+1):n_i}, x_{(n_i+1):(n_i+H)})\}_{i=1}^T \) to learn an underlying function \( F : \mathbb{R}^d \times F_1 \rightarrow \mathbb{R}^H \times F_2 \), where \( F_1 \geq F_2 \) are the number of input and output features, respectively. AEEnbMIMOCQR can handle this scenario by simply training a LSTM network or other type of RNN with the pinball loss. Furthermore, in Algorithm (3), we must have \( H \times F_2 \) non-conformity sets, \( \alpha_s \), and \( \hat{q}_s \), one per each feature and horizon as

\[
\epsilon_{h,f}, \forall h \in \{1, \ldots, H\} \forall f \in \{1, \ldots, F_2\}, \quad (26)
\]

\[
\alpha_{h,f}, \forall h \in \{1, \ldots, H\} \forall f \in \{1, \ldots, F_2\}, \quad (27)
\]

\[
\hat{q}_{h,f}, \forall h \in \{1, \ldots, H\} \forall f \in \{1, \ldots, F_2\}. \quad (28)
\]
In the interest of space, Algorithm (3) will not be replicated for this case, even though the generalization to the multivariate time series case should be intuitive from these hints.

5 Experiments

We use the NN5 forecasting competition dataset as a benchmark. This dataset contains 111 time series, each one consisting of 791 daily cash demand observations at many ATMs at different locations in England. Probabilistic forecasting with ARIMA, EnbPI, EnbCQR, MIMOCQR, and our proposal AEnbMIMOCQR is performed on each series. All the code and data to reproduce the experiments can be found here. As an evaluation criterion, we will look at coverage, median PIs width, and interquartile range (IQR). Min-max normalization on each time series was performed to remove the scale dependence. Specifically, we use the following performance measures

$$\text{Median}^* = \frac{1}{111} \sum_{i=1}^{111} \text{Median}_i,$$

$$\text{IQR}^* = \frac{1}{111} \sum_{i=1}^{111} \text{IQR}_i,$$

$$\text{Coverage}^* = \frac{1}{111} \sum_{i=1}^{111} \text{Coverage}_i,$$

where Median$_i$, IQR$_i$ and Coverage$_i$ are the median PI width, PI IQR width, and coverage of the time series with id=i, respectively.

Table (3) showcases the number of deemed stationary time series via the Augmented Dickey-Fuller (ADF) test [24, 25] for different significance levels and a maxlag of 40.

| Significance level | #series |
|-------------------|---------|
| 0.1               | 92      |
| 0.05              | 84      |
| 0.01              | 67      |

Table 3: Number of stationary time series for different significance levels of the ADF test.

| Hyperparameter        | Value          |
|-----------------------|----------------|
| Epochs                | 1000           |
| Batch size            | 100            |
| Activation            | ReLU           |
| Dropout               | 0.1            |
| First hidden layer    | 100 neurons    |
| Second hidden layer   | 100 neurons    |

Table 4: Hyperparameters used for FFNN training.

| Method          | Model | B | N | T   | T$_1$ | d | H  | α  | φ   |
|-----------------|-------|---|---|-----|------|---|----|----|-----|
| MIMOCQR         | FFNN  | N.A. | N.A. | 322 | 400  | 40 | 30 | 0.1 | N.A.|
| AEnbMIMOCQR     | FFNN  | 10 | 100 | 322 | 400  | 40 | 30 | 0.1 | mean|
| EnbPI           | FFNN  | 10 | N.A. | 351 | 400  | 40 | 30 | 0.1 | mean|
| EnbCQR          | FFNN  | 10 | N.A. | 351 | 400  | 40 | 30 | 0.1 | mean|
| ARIMA           | ARIMA(40, $d^*$, 40) | N.A. | N.A. | 391 | 400  | 40 | N.A.| 0.1 | N.A.|

Table 5: Probabilistic forecasting methods parameters.
Method Coverage* Median* IQR*
MIMOCQR 0.844 0.372 0.109
AEnbMIMOCQR 0.887 0.493 0.350
EnbPI 0.876 0.593 0.010
EnbCQR 0.750 0.476 0.387
ARIMA 0.867 0.553 0.062

Table 6: Relative performance measures.

Besides ARIMA, which has an in-built linear autoregressive model with moving averages, we used a feedforward neural network (FFNN) [26] for all methods using the hyperparameters shown in Table (4). The unique difference is that EnbPI FFNN has a unique output neuron, which represents the single-output forecast optimized with mean squared error loss, while EnbCQR has 3 output neurons, each one minimizing a different pinball loss for $\alpha/2$, 0.5, and $1 - \alpha/2$ to get an estimated lower bound, forecast, and upper bound, respectively. MIMOCQR and AEnbMIMOCQR generalize the former with a total of $H \times 2$ output neurons since they do not require previous forecasts.

The parameters used in the evaluation phase are as follows. We set $T_1 = 400$, i.e., we leave the last 400 timesteps for evaluation, as seen in Fig.(1), and leave the remainder for training for $\alpha = 0.1$. For the multi-output methods, we set $H = 30$ and $d = 40$. We also use $d = 40$ for EnbPI and EnbCQR. The selection of $d$ has a direct impact on the model’s quality and therefore it should be carefully optimized. However, this optimization goes beyond the scope of this article. Nevertheless, we keep the coherence across models by choosing the same $d$ among every model. Table (5) contains all the parameters for each method, whenever a parameter does not apply to the method in question we fill with "N.A.". For ARIMA, we also set $d = 40$, $q = 40$, and the order of differencing $d^*$ is determined by the ADF test for a significance level of $\alpha = 0.1$. Be aware that using $T_1 = 400$ and $H = 30$ results in $\lfloor 400/30 \rfloor = 13$ updates on the non-conformity set for the dynamic methods (AEnbMIMOCQR, EnbPI, and EnbCQR).

Figures (1-4) show the variation of coverage across the last 400 timesteps for all methods, wherein AEnbMIMOCQR is clearly the best method in terms of asymptotic validity. We can also note that MIMOCQR and ARIMA tend to decrease coverage over time due to not accommodated distribution shifts. We can also verify that EnbPI converges faster than EnbCQR, yet not as fast as AEnbMIMOCQR. Figures (5-8) show the last 30-step ahead PIs for all conformal methods on time series with id=4, in which we see that EnbPI PIs are not heteroscedastic, and the coverage differences. In a broader
scope, the performance measures uphold what can be inferred from the foregoing figures, i.e., AEnbMIMOCQR attains the best coverage (see Table 6). Moreover, AEnbMIMOCQR produces lower median PIs than EnbPI, on average, and its PIs vary more compared to EnbPI, on average, suggesting better adequacy for heteroscedastic time series and informative PIs to decision-making tasks. Recall that the small deviation seen in EnbPI PIs width is a mere consequence of updating the non-conformity set each $H = 30$ timesteps since PIs width is the same within the same batch.

Figure 5: Last 30-step ahead PIs (MIMOCQR, id=4).  
Figure 6: Last 30-step ahead PIs (AEnbMIMOCQR, id=4).  
Figure 7: Last 30-step ahead PIs (EnbPI, id=4).  
Figure 8: Last 30-step ahead PIs (EnbCQR, id=4).

6 Conclusion

In this article, we proposed a new method called AEnbMIMOCQR for multi-step ahead conformal heteroscedastic time series forecasting that seeks to produce asymptotic valid PIs, based on the MIMO strategy, a dynamic non-conformity set, and an adaptive conformal inference scheme. Scale-independent performance measures applied on a reasonable scale benchmark containing several non-stationary and seasonal time series suggest that AEnbMIMOCQR completely outperforms competitive conformal methods in terms of coverage and PIs width. Although retraining the model from time to time can optimize the PIs width, AEnbMIMOCQR does not require data splitting as in ICP nor retraining the model. Taking this strong evidence into consideration, henceforth AEnbMIMOCQR should be the state-of-the-art standard framework for conformal multi-step ahead heteroscedastic univariate and multivariate time series.

Acknowledgments

This work has been supported by COMPETE: POCI-01-0247-FEDER-039719 and FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UIDB/00127/2020.

References

[1] Martim Sousa, Ana Maria Tomé, and José Moreira. Improved conformalized quantile regression, 2022.  
[2] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. Algorithmic learning in a random world. 2005.  
[3] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. 2007.  
[4] Martim Sousa. Inductive conformal prediction: A straightforward introduction with examples in python, 2022.
[5] Anastasios N. Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification, 2021.

[6] Shai Feldman, Stephen Bates, and Yaniv Romano. Conformalized online learning: Online calibration without a holdout set, 05 2022.

[7] Isaac Gibbs and Emmanuel Candès. Adaptive conformal inference under distribution shift, 2021.

[8] Margaux Zaffran, Aymeric Dieuleveut, Olivier Féron, Yannig Goude, and Julie Josse. Adaptive conformal predictions for time series, 2022.

[9] Souhaib Ben Taieb, Gianluca Bontempi, Amir Atiya, and Antti Sorjamaa. A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. Expert Systems with Applications, 39, 08 2011.

[10] Souhaib Ben Taieb and Rob Hyndman. Recursive and direct multi-step forecasting: the best of both worlds. Monash Econometrics and Business Statistics Working Papers 19/12, Monash University, Department of Econometrics and Business Statistics, 2012.

[11] Robert H Shumway and David S Stoffer. Arima models. In Time series analysis and its applications, pages 75–163. Springer, 2017.

[12] Rob J Hyndman and George Athanasopoulos. Forecasting: principles and practice. OTexts, 2018.

[13] Chen Xu and Yao Xie. Conformal prediction interval for dynamic time-series. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 11559–11569. PMLR, 18–24 Jul 2021.

[14] Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[15] Roger W Koenker and Gilbert Bassett. Regression quantiles. Econometrica, 46(1):33–50, 1978.

[16] Roger Koenker and Kevin F Hallock. Quantile regression. Journal of economic perspectives, 15(4):143–156, 2001.

[17] Vilde Jensen, Filippo Maria Bianchi, and Stian Norman Anfinsen. Ensemble conformalized quantile regression for probabilistic time series forecasting, 2022.

[18] Ryan J. Tibshirani, Rina Foygel Barber, Emmanuel J. Candes, and Aaditya Ramdas. Conformal prediction under covariate shift, 2019.

[19] Rina Foygel Barber, Emmanuel J. Candes, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability, 2022.

[20] Lelying Guan. Conformal prediction with localization, 2019.

[21] Victor Chernozhukov, Kaspar Wuthrich, and Yinchu Zhu. Exact and robust conformal inference methods for predictive machine learning with dependent data. Proceedings of COLT 2018, 2018.

[22] Kamile Stankeviciute, Ahmed M. Alaa, and Mihaela van der Schaar. Conformal time-series forecasting. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, Advances in Neural Information Processing Systems, volume 34, pages 6216–6228. Curran Associates, Inc., 2021.

[23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. Neural computation, 9:1735–80, 12 1997.

[24] David A. Dickey and Wayne A. Fuller. Distribution of the estimators for autoregressive time series with a unit root. Journal of the American Statistical Association, 74(366):427–431, 1979.

[25] David A. Dickey and Wayne A. Fuller. Likelihood ratio statistics for autoregressive time series with a unit root. Econometrica, 49(4):1057–1072, 1981.

[26] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. MIT Press, Cambridge, MA, USA, 2016.