Expert Systems with Applications 41 (2014) 4235–4244

Contents lists available at ScienceDirect

Expert Systems with Applications

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

Neural network ensemble operators for time series forecasting

Nikolaos Kourentzes *, Devon K. Barrow, Sven F. Crone

Lancaster University Management School, Department of Management Science, Lancaster LA1 4YX, UK

ARTICLE INFO

Keywords:
Time series
Forecasting
Ensembles
Combination
Mode estimation
Kernel density estimation
Neural networks
Mean
Median

ABSTRACT

The combination of forecasts resulting from an ensemble of neural networks has been shown to outperform the use of a single "best" network model. This is supported by an extensive body of literature, which shows that combining generally leads to improvements in forecasting accuracy and robustness, and that using the mean operator often outperforms more complex methods of combining forecasts. This paper proposes a mode ensemble operator based on kernel density estimation, which unlike the mean operator is insensitive to outliers and deviations from normality, and unlike the median operator does not require symmetric distributions. The three operators are compared empirically and the proposed mode ensemble operator is found to produce the most accurate forecasts, followed by the median, while the mean has relatively poor performance. The findings suggest that the mode operator should be considered as an alternative to the mean and median operators in forecasting applications. Experiments indicate that mode ensembles are useful in automating neural network models across a large number of time series, overcoming issues of uncertainty associated with data sampling, the stochasticity of neural network training, and the distribution of the forecasts.

1. Introduction

With the continuing increase in computing power and availability of data, there has been a growing interest in the use of artificial Neural Networks (NNs) for forecasting purposes. NNs are typically used as ensembles of several network models to deal with sampling and modelling uncertainties that may otherwise impair their forecasting accuracy and robustness. Ensembles combine forecasts from the different models that comprise them. This paper proposes a new fundamental ensemble operator for neural networks that is based on estimating the mode of the forecast distribution, which has appealing properties compared to established alternatives.

Although the use of ensembles is nowadays accepted as the norm in forecasting with NNs (Crone, Hibon, & Nikolopoulos, 2011), their performance is a function of how the individual forecasts are combined (Stock & Watson, 2004). Improvements in the ensemble combination operators have direct impact on the resulting forecasting accuracy and the decision making that forecasts support. This has implications for multiple forecasting applications where NN ensembles have been used. Some examples include diverse forecasting applications such as: economic modelling and policy making (Inoue & Kilian, 2008; McAdam & McNelis, 2005), financial and commodities trading (Bodyanskiy & Popov, 2006; Chen & Leung, 2004; Versace, Bhatt, Hinds, & Shiffer, 2004; Yu, Wang, & Lai, 2008; Zhang & Berardi, 2001), fast-moving consumer goods (Trapero, Kourentzes, & Fildes, 2012), tourism (Pattie & Snyder, 1996), electricity load (Hippert, Pedreira, & Souza, 2001; Taylor & Buizza, 2002), temperature and weather (Langella, Basile, Bonfante, & Terribile, 2010; Roebber, Butt, Reinke, & Grafenauer, 2007), river flood (Campolo, Andreussi, & Soldati, 1999) and hydrological modelling (Dawson & Wilby, 2001), climate (Fildes & Kourentzes, 2011), and ecology (Araújo & New, 2007) to name a few. Zhang, Patuwo, and Hu (1998) lists multiple other forecasting applications where they have been employed successfully.

NN ensembles are fundamental for producing accurate forecasts for these various applications; hence, improvements in the construction of the ensembles are important. In this paper, the performance of the proposed mode operator is investigated together with the two existing fundamental ensemble operators: the mean and the median. Two different datasets, having in total 3443 real time series, are used to empirically evaluate the different operators. Furthermore, ensembles of both training initialisations and sampling (bagging) are used to investigate the performance of the operators. The proposed operator is found to be superior to established alternatives. Moreover, the robustness and good performance of the median operator is validated. The findings provide useful insights for the application of NNs in large scale forecasting systems, where robustness and accuracy of the forecasts are equally desirable.

The rest of the paper is organised as follows: Section 2 discusses the benefits of NN ensembles and the limitations of the established

* Corresponding author. Address: Department of Management Science, Lancaster University Management School, Lancaster, Lancashire LA1 4YX, UK. Tel.: +44 1524 592911.

E-mail address: n.kourentzes@lancaster.ac.uk (N. Kourentzes).

0957-4174/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2013.12.011

ensemble operators. Section 3 introduces multilayer perceptrons that will be used for this paper and Section 4 discusses the three fundamental ensemble operators and presents the proposed method for mode ensembles. Sections 5 and 6 discuss the experimental design and the results, respectively, followed by a discussion of the findings in Section 7.

2. Forecasting with neural networks

Over the last two decades there has been substantial research in the use of NNs for forecasting problems, with multiple successful applications (Zhang et al., 1998). Adya and Collopy (1998) found that NNs outperformed established statistical benchmarks in 73% of the papers reviewed. NNs are flexible nonlinear data driven models that have attractive properties for forecasting. They have been proven to be universal approximators (Hornik, Stinchcombe, & White, 1989; Hornik, 1991), being able to fit to any underlying data generating process. NNs have been empirically shown to be able to forecast both linear (Zhang, 2001) and nonlinear (Zhang, Patuwo, & Hu, 2001) time series of different forms. Their attractive properties have led to the rise of several types of NNs and applications in the literature (for examples, see Connor, Martin, & Atlas, 1994; Efendigil, Önüt, & Kahraman, 2009; Khashei & Bijari, 2010; Zhang et al., 1998).

While NNs powerful approximation capabilities and self-adaptive data driven modelling approach allow them great flexibility in modelling time series data, it also complicates substantially model specification and the estimation of their parameters. Direct optimisation through conventional minimisation of error is not possible under the multilayer architecture of NNs and the back-propagation learning algorithm has been proposed to solve this problem (Rumelhart, Hinton, & Williams, 1986), later discussed in the context of time series by Werbos (1990). Several complex training (optimisation) algorithms have appeared in the literature, which may nevertheless be stuck in local optima (Hagan, Demuth, & Beale, 1996; Haykin, 2009). To alleviate this problem, training of the networks may be initialised several times and the best network model selected according to some fitting criteria. However, this may still lead to suboptimal selection of parameters depending on the fitting criterion, resulting in loss of predictive power in the out-of-sample set (Hansen & Salamon, 1990). Another challenge in the parameter estimation of NNs is due to the uncertainty associated with the training sample. Breiman (1996a) in his work on instability and stabilization in model selection showed that subset selection methods in regression, including artificial neural networks, are unstable methods. Given a data set and a collection of models, a method is defined as unstable if a small change in the data results in large changes in the set of models.

These issues pose a series of challenges in selecting the most appropriate model for practical applications and currently no universal guidelines exist on how best to do this. In dealing with the first, the NN literature has strongly argued, with supporting empirical evidence, that instead of selecting a single NN that may be susceptible to poor initial values (or model setup), it is preferable to consider a combination of different NN models (Barrow, Crone, & Kourentzes, 2010; Ben Taieb, Bontempi, Atiya, & Sorjamaa, 2012; Crone et al., 2011; Hansen & Salamon, 1990; Versace et al., 2004; Zhang & Berardi, 2001). Naftaly, Intrator, and Horn (1997) showed that ensembles across NN training initialisations of the same model can improve accuracy while removing the need for identifying and choosing the best training initialisation. This has been verified numerous times in the literature (for example, see Zhang & Berardi, 2001). These ensembles aim at reducing the parameter uncertainty due to the stochasticity of the training of the networks. Instead of relying on a single network that may be stuck to a local minima during its training, with poor forecasting performance, a combination of several networks is used. In the case of uncertainty about the training data, Breiman (1996b) proposed Bagging (Bootstrap aggregation and combination) for generating ensembles. The basic idea behind bagging is to train a model on permutations of the original sample and then combine the resulting models. The resulting ensemble is robust to small changes in the sample, alleviating this type of uncertainty. Recent research has lead to a series of studies involving the application of the Bagging algorithm for forecasting purposes with positive results in many application areas (Chen & Ren, 2009; Hillebrand & Medeiros, 2010; Inoue & Kilian, 2008; Langella et al., 2010; Lee & Yang, 2006). Apart from improving accuracy, using ensembles also avoids the problem of identifying and choosing the best trained network.

In either case, neural network ensembles created from multiple initialisations or from the application of the Bagging algorithm, require the use of an ensemble combination operator. The forecast combination literature provides insights on how to best do this. Bates and Granger (1969) were amongst the first to show significant gains in forecasting accuracy through model combination. Newbold and Granger (1974) showed that a linear combination of univariate forecasts often outperformed individual models, while Ming Shi, Da Xu, and Liu (1999) provided similar evidence for nonlinear combinations. Makridakis and Winkler (1983) using simple averages concluded that the forecasting accuracy of the combined forecast improved, while the variability of accuracy amongst different combinations decreased as the number of methods in the average increased. The well known M competitions provided support to these results; model combination through averages improves accuracy (Makridakis et al., 1982; Makridakis & Hibon, 2000). Elliott and Timmermann (2004) showed that the good performance of equally weighted model averages is connected to the mean squared error loss function, and under varying conditions optimally weighted averages can lead to better accuracy. Agnew (1985) found good accuracy of the median as an operator to combine forecasts. Stock and Watson (2004) considered simple averages, medians and trimmed averages of forecast, finding the average to be the most accurate, although one would expect the more robust median or trimmed mean to perform better. On the other hand, McNees (1992) found no significant differences between the performance of the mean and the median. Kourentzes, Petropoulos, and Trapero (2014) showed that combining models fitted on data sampled at different frequencies can achieve better forecasting accuracy at all short, medium and long term forecast horizons, and found small differences in using either the mean or the median.

There is a growing consensus that model combination has advantages over selecting a single model not only in terms of accuracy and error variability, but also simplifying model building and selection, and therefore the forecasting process as a whole. Nonetheless, the question of how to best combine different models has not been resolved. In the literature there are many different ensemble methods, often based on the fundamental operators of mean and median, in an unweighted or weighted fashion. Barrow et al. (2010) argued that the distribution of the forecasts involved in the calculation of the ensemble prediction may include outliers that may harm the performance of mean-based ensemble forecasts. Therefore, they proposed removing such elements from the ensemble, demonstrating improved performance. Jose and Winkler (2008) using a similar argument advocated the use of trimmed and winsorised means. On the other hand, median based ensembles, are more robust to outliers and such special treatment may be unnecessary. However, the median, as a measure of central tendency is not robust to deviations from symmetric distributions. The median will merely calculate the middle value that separates the higher half from the lower half of the dataset, which is not

guaranteed to describe well the location of the distribution of the forecasts that are used to construct the ensemble.

Taking a different perspective, ensembles provide an estimate of where most forecasts tend to be. Mean and median are merely measures of the central tendency of the forecast distribution. In the case of normal distribution these coincide. Outliers and deviations from normality harm the quality of the estimation. An apparent alternative, that in theory is free of this problem, is the mode. This measure of central tendency has been overlooked in the combination literature because of its inherent difficulty in estimating it for unknown distributions. This paper exploits the properties of the mode to propose a new fundamental ensemble operator. In the following sections this operator is introduced and evaluated against established alternatives.

3. Multilayer perceptrons

The most commonly used form of NNs for forecasting is the feedforward multilayer perceptron. The one-step ahead forecast $\hat{y}_{t+1}$ is computed using inputs that are lagged observations of the time series or other explanatory variables. I denotes the number of inputs p_i of the NN. Their functional form is:

$$\hat{y}_{t+1} = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\gamma_{0i} + \sum_{i=1}^{I} \gamma_{hi} p_i\right). \quad (1)$$

In Eq. (1), $\boldsymbol{w} = (\boldsymbol{\beta}, \boldsymbol{\gamma})$ are the network weights with $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_H]$, $\boldsymbol{\gamma} = [\gamma_{11}, \ldots, \gamma_{HI}]$ for the output and the hidden layers, respectively. The β_0 and γ_{0i} are the biases of each neuron, which for each neuron act similarly to the intercept in a regression. H is the number of hidden nodes in the network and $g(\cdot)$ is a non-linear transfer function, which is usually either the sigmoid logistic or the hyperbolic tangent function. NNs can model interactions between inputs, if any. The outputs of the hidden nodes are connected to an output node that produces the forecast. The output node is often linear as in Eq. (1).

In the time series forecasting context, neural networks can be perceived as equivalent to nonlinear autoregressive models (Connor et al., 1994). Lags of the time series, potentially together with lagged observations of explanatory variables, are used as inputs to the network. During training pairs of input vectors and targets are presented to the network. The network output is compared to the target and the resulting error is used to update the network weights. NN training is a complex nonlinear optimisation problem, and the network can often get trapped in local minima of the error surface. In order to avoid poor quality results, training should be initialised several times with different random starting weights and biases to explore the error surface more fully. Fig. 1 provides an example of an error surface of a very simple NN. The example network is tasked to model a time series with a simple autoregressive input and is of the form $\hat{y}_{t+1} = g(w_2 g(w_1 y_{t-1}))$, where $g(\cdot)$ is the hyperbolic tangent and w_1 and w_2 its weights. Six different training initialisations, with their respective final weights, are shown. Observe that minor differences in the starting weights can result in different estimates, even for such a simple model. In order to counter this uncertainty an ensemble of all trained networks can be used. As discussed before, this approach has been shown to be superior to choosing a single set of estimated weights.

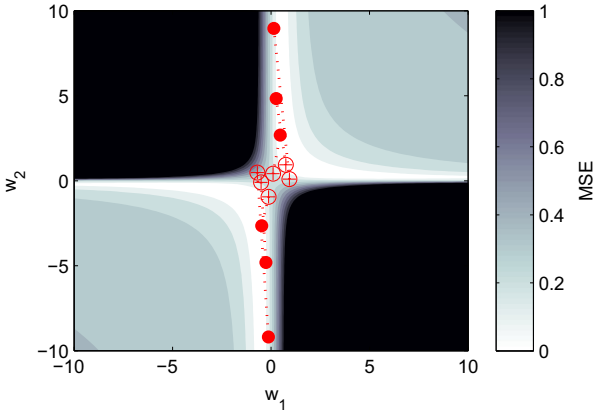

Fig. 1. Contour plot of the error surface of a neural network. The initial (⊕) and ending (•) weights for six different training initialisations are marked.

Note that the objective of training is not to identify the global optimum. This would result in the model over-fitting to the training sample and would then generalise poorly to unseen data (Bishop, 1996), in particular given their powerful approximation capabilities (Hornik, 1991). Furthermore, as new data become available, the prior global optimum may no longer be an optimum.

In general, as the fitting sample changes, with the availability of new information, so do the final weights of the trained networks, even if the initial values of the network weights were kept constant. This sampling induced uncertainty can again be countered by using ensembles of models, following the concept of bagging.

4. Ensemble operators

Let $\hat{y}_{mt}$ be a forecast from model m for period t, where $m = 1, \ldots, M$ and M the number of available forecasts to be combined in an ensemble forecast $\tilde{y}_t$. In this section the construction of $\tilde{y}_t$ using the mean, median and the proposed mode operators is discussed. To apply any of these operators reliably a unimodal distribution is assumed.

4.1. Mean ensemble

The mean is one of the most commonly used measures of central tendency and can be weighted or unweighted. Let w_m be the weight for the forecasts from model m. Conventionally $0 \leqslant w_m \leqslant 1$ and $\sum_{m=1}^{M} w_m = 1$. The ensemble forecast for period t is calculated as:

$$\tilde{y}_t^{\text{Mean}} = \sum_{m=1}^{M} w_m y_{mt}. \quad (2)$$

If all $w_m = M^{-1}$ the resulting combination is unweighted. The properties of the mean are well known, as well as its limitations. The mean is sensitive to outliers and unreliable for skewed distributions. To avoid some of its problems one might use a winsorised or truncuted mean (Jose & Winkler, 2008). In this case the mean behaves more closely to the median. For distributions with finite variance, which is true for sets of forecasts, the maximum distance between the mean and the median is one standard deviation (Mallows, 1991).

4.2. Median ensemble

Similarly the median can be unweighted or weighted, although the latter is rarely used. The median ensemble $\tilde{y}_t^{\text{Median}}$ is simply calculated sorting $w_m y_{mt}$ and picking the middle value if M is odd or the mean of the two middle values otherwise. Although the median is more robust than the mean it still suffers with non-symmetric distributions.

4.3. Mode ensemble

The mode is defined as the most frequent value in a set of data. The mode is insensitive to outliers, in contrast to the mean and median. There is no formula to calculate the mode of an unknown

distribution for continuous variables. There are two common ways to calculate it: either by discretising the data and identifying the most frequent bin, or by kernel density estimation. In this work the second approach is preferred in order to avoid the discretisation of the data. Furthermore, kernel density estimation lends itself well to the continuous-valued nature of forecasts.

Kernel density estimation is a non-parametric way to estimate the probability density function of a random variable, in this case the forecasts. Given forecasts of a distribution with unknown density f, we can approximate its shape using the kernel density estimator

$$\hat{f}_{th}(x) = (Mh)^{-1}\sum_{m=1}^{M} K\left(\frac{x-\hat{y}_{mt}}{h}\right), \tag{3}$$

where $K(\cdot)$ is a function with the property $\int K(x)dx = 1$ that is called kernel and $h > 0$ is its bandwidth. The kernel is often chosen to be a unimodal symmetric density function, therefore making $\hat{f}_{th}(x)$ a density function itself, which is often, for computational reasons, the Gaussian kernel $\phi(x)$:

$$\phi_h(x) = \frac{1}{\sqrt{2\pi}h} e^{\frac{-x^2}{2h^2}}. \tag{4}$$

Fig. 2 shows an example of the calculation of kernel density. A kernel with bandwidth h is fitted around each observation and the resulting sum approximates the density function of the sample.

A number of alternative kernel functions have been proposed in the literature, however the choice of kernel has been found to have minimal impact on the outcome for most cases (Wand & Jones, 1995). The bandwidth of the kernel h controls the amount of smoothing. A high bandwidth results in more smoothing. Therefore, the choice of h is crucial, as either under-smoothing or over-smoothing will provide misleading estimation of the density f (Silverman, 1981). The approximation by Silverman (1998) is often used in practice

$$h = \left(\frac{4\hat{\sigma}^5}{3M}\right)^{\frac{1}{5}}, \tag{5}$$

where $\hat{\sigma}$ is the standard deviation of the sample of the forecasts. This approximation is often adequate for Gaussian kernels. Botev, Grotowski, and Kroese (2010) propose a methodology to automatically select the bandwidth that is free from the arbitrary normal reference rules used by existing methods. This is preferred in the calculation of the mode ensemble as the resulting bandwidth h allows fast convergence and good performance of the ensemble, as it is discussed in the results.

The value x that corresponds to the maximum density approximates the mode of the true underlying distribution for a set of forecasts, which is also the value of the mode ensemble $\hat{y}^{\text{Mode}}_{t+h}$. This is true as long as the estimated distribution is unimodal. Although the probability of facing non-unimodal distributions when dealing with forecasts is low, the following heuristic is proposed to resolve such cases. Since there is no preference between the modes, the one closer to the previous (forecasted or actual) value is retained as the mode. This results in smooth trace forecasts. Eq. (3) results in unweigthed $\hat{y}^{\text{Mode}}_{t+h}$. It is trivial to introduce w_m individual weights for each model.

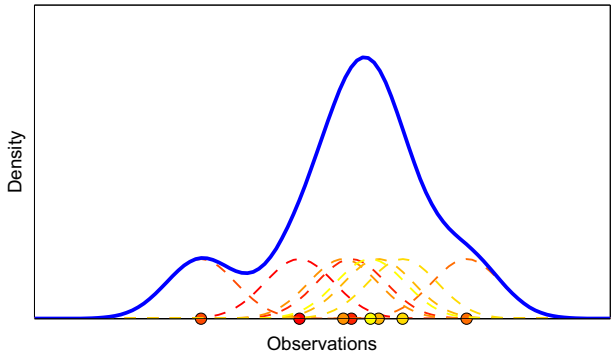

Fig. 2. Example calculation of kernel density estimation.

For kernel density estimation to adequately reveal the underlying density a relatively large number of observations are required. A small number of observations will lead to a bad approximation. This is illustrated in Fig. 3. It shows the mean, median, mode ensembles as well as the selected "best" model forecast, as selected using a validation sample for four different forecast horizons. Furthermore, the estimated kerned density for each horizon is plotted. It is apparent by comparing Fig. 3a and b that the kernel density estimation using only 10 models is very poor. While in Fig. 3a the shape of the distribution is successfully approximated, in Fig. 3b there are not enough forecasts to identify the underlying shape of the distribution of the forecasts. Furthermore, in Fig. 3a it is easy to see that neither the mean, median or the "best" model are close to the most probable value of the forecast distribution. The mode ensemble offers an intuitive way of identifying where forecasts from different models converge and provide a robust forecast, independent of distributional assumptions.

5. Empirical evaluation

5.1. Datasets

To empirically evaluate the performance of the mean, median and the proposed mode ensemble for NNs, two large datasets of real monthly time series are used. The first dataset comes from Federal Reserve Economic Data (FRED) of St. Luis.[1] From the complete dataset 3000 monthly time series that contain 108 or more observations (9 years) were sampled. Long time series were preferred to allow for adequate training, validation and test sets. The second dataset comes from the UK Office for National Statistics and contains 443 monthly retail sales time series.[2] Again, only time series with 108 or more observations were retained for the empirical evaluation.

A summary of the characteristics of the time series in each dataset is provided in Table 1. To identify the presence of trend in a time series the cox-stuart test was employed on a 12-period centred moving average fitted to each time series. The test was performed on the centred moving average to smooth any effects from irregularities and seasonality. To identify the presence of seasonality, seasonal indices were calculated for the de-trended time series and then these were tested for significant deviations from each other by means of a Friedman test. This procedure, based on non-parametric tests, is robust, however different tests may provide slightly different percentages to those in Table 1.

The last 18 observations from each time series are withheld as test set. The prior 18 observations are used as validation set to accommodate NNs training.

5.2. Experimental design

A number of NN ensemble models are fitted to each time series. Two are based on mean, two on median and two on mode ensembles. Hereafter, these are named *NN-Mean*, *NN-Median* and

[1] The dataset can be accessed at http://research.stlouisfed.org/fred2/.

[2] The dataset can be accessed at http://www.ons.gov.uk/ons/rel/rsi/retail-sales/January-2012/tsd-retail-sales.html.

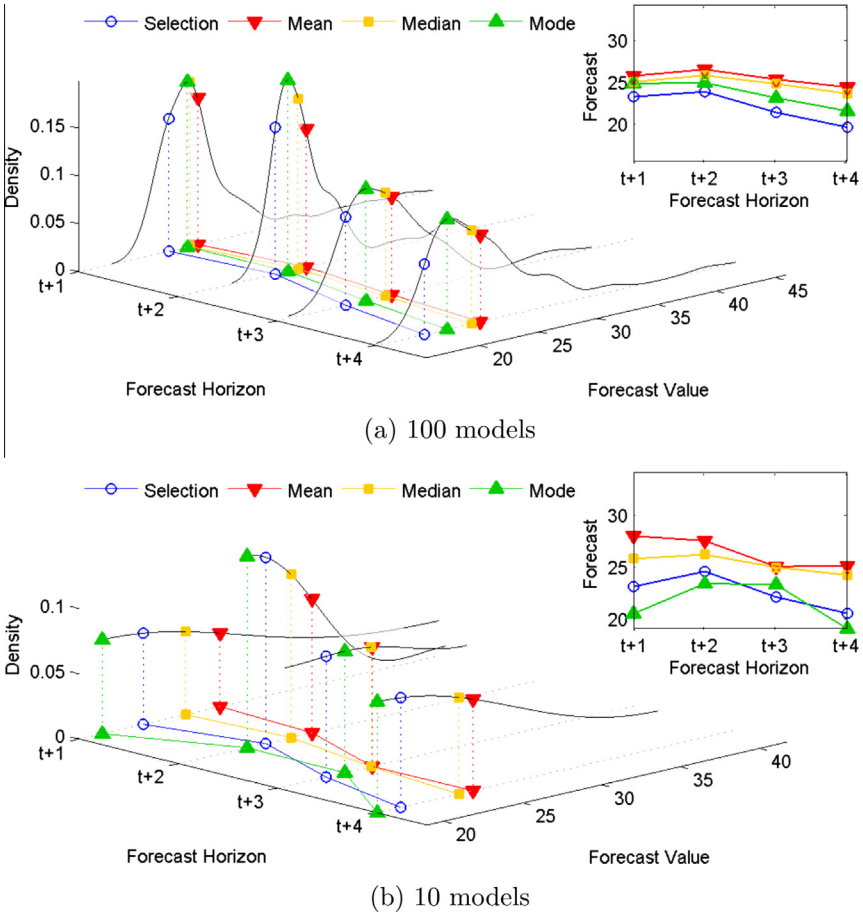

Fig. 3. Example of the distribution of NN forecasts of different number of models, as estimated by Gaussian kernel density estimation, for the first four steps ahead. The forecasts by model selection, mean, median and mode ensembles are provided.

Table 1
Dataset characteristics.

Dataset	Series	Series length			Series patterns			
		Min.	Mean	Max.	Level (%)	Trend (%)	Season (%)	Trend–Season (%)
FRED	3000	111	327	1124	5.37	40.70	5.80	48.13
Retail	443	179	270	289	15.12	48.98	1.81	34.09

NN-Mode, respectively. All combination operators are applied in their unweighted version, as the objective is to test their fundamental performance. In each pair of ensembles, the first is a training ensemble, combining multiple training initialisations and the second is based on bagging, bootstrapped as described by Kunsch (1989). This moving block bootstrap samples the original time series while preserving the temporal and spatial covariance structure, as well as the serial correlation of the time series data. By assessing the operators using different types of ensembles we aim to assess the consistency of their performance. Furthermore, different sizes of ensembles are evaluated, from 10 members up to 100 members, with steps of 10. Results for single NN models, based on selecting the best one, are not provided as there is compelling evidence in the literature that ensembles are superior (for examples, see Barrow et al., 2010; Zhang & Berardi, 2001). This was validated in our experiments as well.

The individual neural networks have identical setup. Following the suggestions of the literature, if trend is identified in a time series it is removed through first differencing (Zhang & Qi, 2005). The time series is then linearly scaled between −0.5 and 0.5 to facilitate the NN training. The inputs are identified through means of stepwise regression, which has been shown to perform well for identifying univariate input lags for NNs (Crone & Kourentzes, 2010; Kourentzes & Crone, 2010). All networks use the hyperbolic tangent transfer function for the hidden nodes and a linear output node. The number of hidden nodes was identified experimentally for each time series. Up to 60 hidden nodes were evaluated for each time series and the number of hidden nodes that minimised the validation Mean Squared Error (MSE) was chosen.

Each network was trained using the Levenberg–Marquardt (LM) algorithm. The algorithm requires setting a scalar μ_{LM} and its increase and decrease steps. When the scalar is zero, the LM algorithm becomes just Newton's method, using the approximate Hessian matrix. On the other hand, when μ_{LM} is large, it becomes gradient descent with a small step size. Newton's method is more accurate and faster near an error minimum, so the aim is to shift toward Newton's method as quickly as possible. If a step would increase the fitting error then μ_{LM} is increased. Here $\mu_{LM} = 10^{-3}$, with an increase factor of $\mu_{inc} = 10$ and a decrease factor of $\mu_{dec} = 10^{-1}$. For a detailed description of the algorithm and its parameters see Hagan et al. (1996). MSE was used as the training cost function. The maximum training epochs are set to 1000. The training can

stop earlier if μ_{LM} becomes equal or greater than $\mu_{max} = 10^{10}$. The MSE error at the validation set is tracked while training. If the error increases consequently for 50 epochs then training is stopped. The weights that give the lowest validation error are selected at the end of each training. This is common practice in the literature and helps to achieve good out-of-sample performance, since it avoids over-fitting to the training sample (Haykin, 2009).

Following the suggestions of the forecasting literature (Adya & Collopy, 1998) two statistical benchmarks are used in this study, namely the naive forecast (random walk) and exponential smoothing. This is done to assess the accuracy gains of using NNs against established simpler statistical methods. The *Naive* requires no parameterisation or setup, hence is used as a baseline that any more complex model should outperform. The appropriate exponential smoothing model is selected for each time series, depending on the presence of trend and/or seasonality using Akaike's Information Criterion. Model parameters are identified by optimising the log-likelihood function (Hyndman, Koehler, Snyder, & Grose, 2002; Hyndman, Koehler, Ord, & Snyder, 2008). Exponential smoothing was selected as a benchmark based on its widely demonstrated forecasting accuracy and robustness (Gardner, 2006; Makridakis & Hibon, 2000) and will be named *ETS* in this work. The use of these benchmarks can help establish the relative performance of the NN models. In total, eight forecasting models are fitted to each time series, six NNs and two statistical benchmarks.

Rolling trace forecasts of 12 months are produced using each model. The rolling origin evaluation enables collecting a large sample of forecasts and their errors, while being robust to irregular forecast origins and outliers, thus providing reliable error measurements. Based on the long test set, 7 trace forecasts from $t+1$ up to $t+12$ months are collected for each time series. The reader is referred to Tashman (2000) for a detailed description of the evaluation scheme and its advantages.

The forecasting accuracy is assessed using the Mean Absolute Scaled Error (MASE). This is preferred due to its favourable statistical properties. MASE is calculated for each trace forecast as:

$$\text{MASE} = m^{-1}\sum_{j=1}^{m}\frac{|\, y_j - \hat{y}_j \,|}{(n-1)^{-1}\sum_{r=2}^{n}|\, y_r - y_{r-1} \,|}, \tag{6}$$

where y_j and $\hat{y}_j$ are the actual and forecasted value for $j = 1, \ldots, m$ out-of-sample observations. The denominator is the mean absolute error of the random walk in the fitting sample of n observations and is used to scale the error. MASE, being a scaled error, permits summarising model performance across time series of different scale and units, which mean squared or absolute errors cannot do, and is less biased from errors like the mean absolute percentage error and its symmetric equivalent. Another advantage of this error is that it is very improbable that the denominator is zero, therefore making it easy to calculate in several scenarios and robust to time series with several values equal or close to zero (Hyndman & Koehler, 2006). Note that the Retail dataset contains several time series that do not permit the calculation of conventional percentage errors, due to zero values in the denominator. To summarise the results across the time series of each dataset the mean and median MASE across all series are calculated.

6. Results

Table 2 presents the results for the FRED time series. Numbers in brackets refer to median MASE, while the rest to mean MASE. The table provides results for ensembles from 10 to 100 members. The results for bagging and training ensembles are presented separately to assess the impact of the ensemble type on the different ensemble operators. In each row the best performing method according to mean and median MASE is highlighted in boldface.

Overall, the difference between the mean and median MASE results indicates that there are several difficult time series, particularly affecting the less robust mean MASE. Focusing on the bagging results, all *NN-Mean*, *NN-Median* and *NN-Mode* are more accurate than the benchmarks when considering mean MASE. Furthermore, as the ensembles increase in size their accuracy improves. In particular, for *NN-Mode* after there are 30 or more members the forecasts are very accurate. This was to be expected since the kernel density estimation becomes reliable once there is an adequate number of observations, as discussed in Section 4. For ensembles of 70 or more members *NN-Mode* provides consistently the best accuracy, closely followed by *NN-Median*. Note that achieving large numbers of ensemble members is trivial with NNs, as this merely implies that more training initialisations or bootstrapped samples are used. Therefore, the requirement of the mode operator for 30 or more ensemble members is not a limiting factor. In contrast, *NN-Mean* underperforms to the extent that *ETS* is more accurate for median MASE. This is an interesting finding, given how common is the mean operator for ensembles in the literature. The

Table 2
Mean (median) MASE for FRED dataset. The best result in each row is highlighted in bold.

Ensemble size	NN-Mean	NN-Median	NN-Mode	Naive	ETS
Bagging					
10	1.06 (0.66)	**0.92** (0.64)	1.30 (0.77)	1.11 (0.87)	3.43 (**0.62**)
20	1.06 (0.65)	**0.90** (0.63)	0.94 (0.65)	1.11 (0.87)	3.43 (**0.62**)
30	1.02 (0.65)	**0.89** (**0.62**)	**0.89** (**0.62**)	1.11 (0.87)	3.43 (**0.62**)
40	1.04 (0.65)	**0.88** (0.62)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)
50	1.03 (0.64)	**0.88** (0.62)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)
60	1.03 (0.64)	0.89 (0.62)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)
70	1.04 (0.65)	0.88 (0.62)	**0.87** (**0.61**)	1.11 (0.87)	3.43 (0.62)
80	1.03 (0.65)	0.88 (0.62)	**0.87** (**0.61**)	1.11 (0.87)	3.43 (0.62)
90	1.01 (0.65)	0.88 (**0.61**)	**0.87** (**0.61**)	1.11 (0.87)	3.43 (0.62)
100	1.01 (0.65)	0.88 (**0.61**)	**0.87** (**0.61**)	1.11 (0.87)	3.43 (0.62)
Training ensemble					
10	1.05 (0.64)	**0.95** (**0.62**)	1.17 (0.70)	1.11 (0.87)	3.43 (**0.62**)
20	1.03 (0.65)	**0.93** (**0.62**)	0.95 (0.64)	1.11 (0.87)	3.43 (**0.62**)
30	1.01 (0.64)	0.91 (**0.62**)	**0.90** (**0.62**)	1.11 (0.87)	3.43 (**0.62**)
40	1.02 (0.64)	0.91 (0.62)	**0.90** (**0.61**)	1.11 (0.87)	3.43 (0.62)
50	1.02 (0.64)	0.92 (0.62)	**0.89** (**0.61**)	1.11 (0.87)	3.43 (0.62)
60	1.01 (0.64)	0.91 (**0.62**)	**0.89** (**0.62**)	1.11 (0.87)	3.43 (**0.62**)
70	1.01 (0.64)	0.91 (**0.61**)	**0.89** (**0.61**)	1.11 (0.87)	3.43 (0.62)
80	1.01 (0.64)	0.91 (0.62)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)
90	1.01 (0.64)	0.91 (**0.61**)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)
100	1.01 (0.64)	0.91 (0.62)	**0.88** (**0.61**)	1.11 (0.87)	3.43 (0.62)

Table 3
Mean (median) MASE for retail dataset. The best result in each row is highlighted in bold.

Ensemble size	NN-Mean	NN-Median	NN-Mode	Naive	ETS
Bagging					
10	1.33 (**0.96**)	**1.11** (0.93)	1.44 (1.10)	1.45 (1.29)	1.12 (0.97)
20	1.37 (0.97)	**1.10** (**0.94**)	1.14 (**0.94**)	1.45 (1.29)	1.12 (0.97)
30	1.29 (0.96)	1.10 (**0.91**)	**1.09** (0.92)	1.45 (1.29)	1.12 (0.97)
40	1.31 (0.97)	1.10 (0.91)	**1.09** (**0.90**)	1.45 (1.29)	1.12 (0.97)
50	1.30 (0.97)	1.10 (0.92)	**1.09** (**0.89**)	1.45 (1.29)	1.12 (0.97)
60	1.26 (0.96)	**1.09** (0.91)	**1.09** (**0.89**)	1.45 (1.29)	1.12 (0.97)
70	1.26 (0.96)	**1.09** (0.91)	**1.09** (**0.90**)	1.45 (1.29)	1.12 (0.97)
80	1.26 (0.98)	1.09 (0.90)	**1.08** (**0.88**)	1.45 (1.29)	1.12 (0.97)
90	1.27 (0.97)	**1.09** (0.90)	**1.09** (**0.87**)	1.45 (1.29)	1.12 (0.97)
100	1.27 (0.95)	**1.09** (0.91)	**1.09** (**0.88**)	1.45 (1.29)	1.12 (0.97)
Training ensemble					
10	1.34 (0.97)	1.14 (**0.91**)	1.27 (0.97)	1.45 (1.29)	**1.12** (0.97)
20	1.33 (0.95)	1.14 (**0.91**)	1.14 (**0.91**)	1.45 (1.29)	**1.12** (0.97)
30	1.31 (0.96)	1.13 (**0.89**)	**1.11** (0.90)	1.45 (1.29)	1.12 (0.97)
40	1.28 (0.95)	1.12 (0.91)	**1.11** (**0.90**)	1.45 (1.29)	1.12 (0.97)
50	1.28 (0.96)	1.12 (**0.90**)	**1.11** (0.91)	1.45 (1.29)	1.12 (0.97)
60	1.29 (0.96)	1.12 (**0.89**)	**1.11** (0.91)	1.45 (1.29)	1.12 (0.97)
70	1.29 (0.95)	1.13 (**0.90**)	**1.11** (**0.90**)	1.45 (1.29)	1.12 (0.97)
80	1.29 (0.95)	1.13 (**0.90**)	**1.11** (**0.90**)	1.45 (1.29)	1.12 (0.97)
90	1.28 (0.95)	1.13 (**0.89**)	**1.12** (0.90)	1.45 (1.29)	**1.12** (0.97)
100	1.28 (0.96)	1.13 (**0.89**)	**1.12** (0.90)	1.45 (1.29)	**1.12** (0.97)

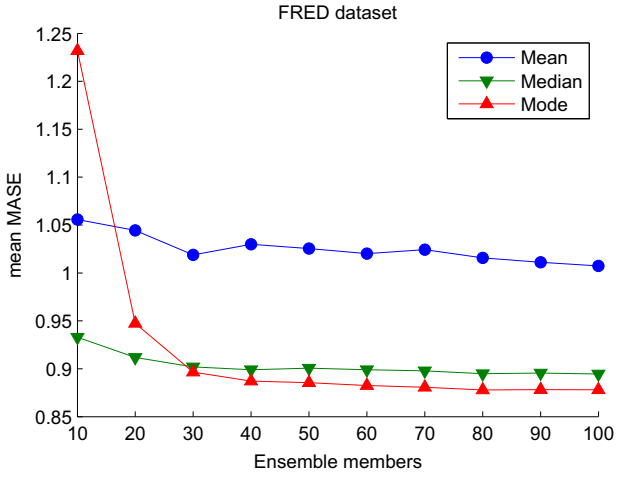

Fig. 4. Mean MASE for different number of ensemble members for the FRED dataset.

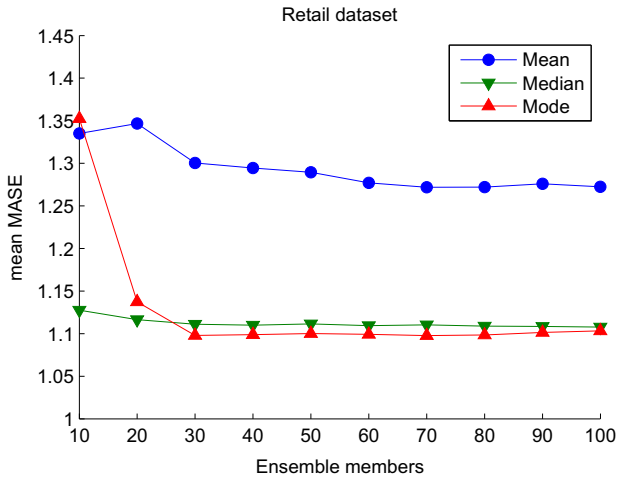

Fig. 5. Mean MASE for different number of ensemble members for the Retail dataset.

more robust behaviour of median and the in-sensitive to outliers nature of the mode result in more accurate ensemble forecasts. Looking at mean MASE, all NNs behave more robust than *ETS*, the latter being severely affected by outliers.

The results of the training ensembles are very similar. Again, as the number of members in the ensemble increases *NN-Mode* performs better and is the most accurate model for 40 or more ensemble members. *NN-Median* ranks second with small differences, while *NN-Mean* is substantially worse. Comparing the results between bagging and training ensembles we can see that the former is marginally more accurate for *NN-Median* and *NN-Mode* when mean MASE is considered. However, the same is not true for *NN-Mean*, indicating that the robustness and performance of this ensemble operator is affected by the type of ensemble.

Table 3 presents the results for the Retail dataset. Its structure is the same as in Table 2. The differences between mean and median MASE are smaller than the FRED results, showing that the time series in this dataset are better behaved. Considering the bagging results, *NN-Median* consistently outperforms the statistical benchmarks, while the same is true for *NN-Mode*, once there is an adequate number of members in the ensemble (again 30 or more). *NN-Mode* is the most accurate model with the lowest mean and median MASE. This is followed closely by *NN-Median*. On the other hand, *NN-Mean* often fails to outperform the benchmark *ETS*, although it is always better than the *Naive*.

Looking at the accuracy of the training ensembles *NN-Mode* is overall more accurate for mean MASE, *NN-Median* is the most accurate for median MASE. Although all NN models outperform the *Naive* benchmark, the differences between either *NN-Mode* or *NN-Median* and *ETS* are very small. *NN-Mean* is worse than *ETS* in terms of mean MASE, while occasionally it is marginally better in terms of median MASE. Comparing accuracies between bagging and training ensembles there are differences in favour of the former when looking at *NN-Median* and *NN-Mode*, while the accuracy for *NN-Mean* is almost identical for both types of ensembles.

Across both datasets *NN-Mode* and *NN-Median* are the most accurate models. *NN-Mode* seems to perform better when the size of ensemble is large enough. *NN-Median* has slightly lower accuracy. While large ensembles benefit *NN-Median*, it can perform well for small ensembles too. Both these models are on average more accurate than the statistical benchmarks *ETS* and *Naive*. On

Table 4
Average computational time comparison.

Ensemble operator	FRED			Retail		
	Ensemble size	Mean time (s)	Difference (%)	Ensemble size	Mean time (s)	Difference (%)
Mean	100	9.74	+25.0	70	5.33	+133.3
Median	100	9.74	+25.0	90	6.85	+200.0
Mode	80	7.79	–	30	2.28	–

the other hand, *NN-Mean* provides mixed results. In both datasets it outperforms *Naive*, but not always *ETS*. It is substantially outperformed by both *NN-Mode* and *NN-Median*.

7. Discussion

The value of ensembles for NNs has been argued theoretically and demonstrated empirically. The combination of the models has often involved some type of mean operator. The empirical evaluation in this paper found that the less commonly used median operator and the proposed mode operator are more accurate and thus preferable. The size of the ensemble was found to be important for the accuracy of all operators. Both mode and median, for the two datasets investigated here, seemed on average to converge for ensembles of 60 or more members, with any additional members offering minimal changes in the forecasting performance. In particular, the mode, due to its reliance on kernel density estimation, required at least 30 members. However, after that point it was found to be on average the most accurate ensemble operator. This is illustrated in Figs. 4 and 5 that present the mean MASE for different number of ensemble members for the FRED and the Retail datasets, respectively. The results for the different type of ensembles have been pooled together, since they had only small differences. Note that there is little evidence that the mean ensembles had converged even with 100 members. Even larger ensembles were not calculated due to the substantial computational resources required, especially when the objective is to forecast a large number of time series, which is common in supply chain and retailing forecasting applications.

In order to assess whether these differences are significant or not, we employ the testing methodology suggested by Koning, Franses, Hibon, and Stekler (2005) that is appropriate for comparing forecasts from multiple models. The comparison is done across all different ensemble sizes to highlight if an operator is consistently statistically different. First, a Freidman test is used to assess whether the accuracy of any model is significantly different from the rest. Subsequently, the MCB test is used to reveal the exact ordering of the different operators, and any significant differences between them. For both datasets the mode operator was significantly better than the median, which in turn was significantly different than the mean, at 5% significance level.

At this point, it is useful to comment on the associated computational cost of the NN ensembles. The main cost comes from training the networks. Therefore, the more ensemble members that need to be trained, the less scalable forecasting with NNs becomes, and the operator that achieves good forecasting performance with the least amount of members is preferable. Table 4 provides an overview of the average time required for forecasting across all series, for each dataset. As a different number of hidden nodes are used for each time series, the complexity of NN training changes, requiring different amount of time. To keep the presentation of the values simple, we summarise the training time over different series into the reported average time. The ensemble size that gave the minimum error for each operator in Figs. 4 and 5 is used as reference for the comparison. The average time in seconds, as well as the percentage difference over the time needed for the mode ensembles, are provided in the table. The networks were trained in parallel on an i7-3930 K CPU clocked at 4.5 GHz with 12 logical cores.

The mode operator needed the least number of ensemble members, requiring from 25% up to 200% less time than the mean or median operators across both datasets. Therefore, apart from the significant gains in forecasting accuracy, the proposed ensemble operator required the least computational resources. In particular for the retailing dataset, the run-time was more than halved.

In Figs. 4 and 5 it is clear that similar performance is achieved for a large range of ensemble sizes for the median operator. This allows exchanging marginal differences in accuracy for smaller runtimes, thus improving its scalability as well. On the other hand, this is not the case with the mean operator, the accuracy of which improves with bigger ensembles.

In the experiments, two types of ensembles were considered, bagging and training ensembles. Each one tackles a different type of parameter uncertainty. We examined whether the performance of the operators was affected by the type of ensemble. Again median and mode had very similar performance, favouring bagging. For the mean this behaviour was not evident. We attribute this different behaviour to the sensitivity of the mean to extreme values, which both median and mode are designed to avoid, albeit with different success.

8. Conclusions

This paper evaluates different fundamental ensemble operators. The well known mean and the less commonly used median were compared, together with a proposed mode operator that is based on kernel density estimation. All three operators attempt to describe the location of the distribution of the forecasts of the members of an ensemble. However, they deal with outlying extreme values differently, with the mean being the most sensitive and the mode the least. Furthermore, distributional asymmetries can affect both the mean and the median, while the mode is immune.

The findings in this paper suggest that both median and mode are very useful operators as they provided better accuracy than mean ensembles consistently across both datasets. The mode was found to be the most accurate, followed by the median. Based on this finding, we recommend investigating the use of the mode and median operators further in ensembles research and applications, which have been largely overlooked in the literature that has mainly focused on the mean.

Furthermore, this work demonstrated that mode ensembles can robustly and accurately forecast automatically a large number of time series with neural networks, while the commonly used mean ensembles were often outperformed by exponential smoothing forecasts. Moreover, mean ensembles required a very large number of members, which neither mode or median needed, with apparent implications for computational costs. In particular, the mode operator was found to require the least computation resources, due to the relatively small number of ensemble members that needed to be trained.

We have already mentioned a number of applications that can benefit from improved NN ensemble forecasts, ranging from

economic and business forecasting to climate modelling. Most of these applications are characterised by forecasting a few, yet important, time series. The improved scalability of mode ensembles over the commonly used mean ensembles allows applying NNs to areas that routinely require large scale automatic forecasting, which can benefit from the nonlinear modelling capabilities of NNs. One such example is retailing, where one has to forecast a large number of products, the sales of which are affected by multiple factor that interact in a nonlinear fashion, such as pricing, promotional and temperature effects. The improved scalability of mode ensembles, compounded with the ever increasing computing capabilities provides opportunities for novel important forecasting applications of NN ensembles. This paper found significant savings in computing time from the proposed operator, which over the complete number of time series accounts for several hours of computations. Such reduction will also help using NN ensembles in high frequency forecasting cycles, where the computational speed has been a limiting factor. Future work should explore these potentials.

The empirical evaluation, in this paper, focused on the unweighted version of all these operators, trying to assess their fundamental properties. Although their differences are attributed to their robustness to extreme values, future research should extend this work to weighted versions of the operators. This will allow considering their use on further ensemble types, such as boosting.

References

Adya, M., & Collopy, F. (1998). How effective are neural networks at forecasting and prediction? A review and evaluation. *Journal of Forecasting, 17*(5–6), 481–495. http://dx.doi.org/10.1002/(SICI)1099-131X(1998090)17:5/6<481::AID-FOR709>3.0.CO;2-Q.

Agnew, C. E. (1985). Bayesian consensus forecasts of macroeconomic variables. *Journal of Forecasting*, 1099-131X, *4*(4), 363–376. http://dx.doi.org/10.1002/for.398004040.

Araújo, M. B., & New, M. (2007). Ensemble forecasting of species distributions. *Trends in Ecology & Evolution, 22*(1), 42–47.

Barrow, D., Crone, S., & Kourentzes, N. (2010). An evaluation of neural network ensembles and model selection for time series prediction. In *The 2010 international joint conference on neural networks (IJCNN)*, 1098-7576 (pp. 1–8). IEEE. http://dx.doi.org/10.1109/IJCNN.2010.5596686.

Bates, J. M., & Granger, C. W. J. (1969). The combination of forecasts. *Operational Research Quarterly, 20*(4), 451–468.

Ben Taieb, S., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications, 39*(8), 7067–7083.

Bishop, C. M. (1996). *Neural networks for pattern recognition* (1st ed.). 9780198538646. USA: Oxford University Press.

Bodyanskiy, Y., & Popov, S. (2006). Neural network approach to forecasting of quasiperiodic financial time series. *European Journal of Operational Research, 175*(3), 1357–1366.

Botev, Z. I., Grotowski, J. F., & Kroese, D. P. (2010). Kernel density estimation via diffusion. *The Annals of Statistics, 38*(5), 2916–2957. http://dx.doi.org/10.1214/10-AOS799.

Breiman, L. (1996a). Heuristics of instability and stabilization in model selection. *The Annals of Statistics, 24*(6), 2350–2383.

Breiman, L. (1996b). Bagging predictors. *Machine Learning, 24*(2), 123–140.

Campolo, M., Andreussi, P., & Soldati, A. (1999). River flood forecasting with a neural network model. *Water Resources Research, 35*(4), 1191–1197.

Chen, A.-S., & Leung, M. T. (2004). Regression neural network for error correction in foreign exchange forecasting and trading. *Computers & Operations Research, 31*(7), 1049–1068.

Chen, T., & Ren, J. (2009). Bagging for Gaussian process regression. *Neurocomputing, 72*(7), 1605–1610.

Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE Transactions on Neural Networks, 5*, 240–254.

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting, 27*(3), 635–660. http://dx.doi.org/10.1016/j.ijforecast.2011.04.001.

Crone, S. F., & Kourentzes, N. (2010). Feature selection for time series prediction – A combined filter and wrapper approach for neural networks. *Neurocomputing, 73*(10–12), 1923–1936. http://dx.doi.org/10.1016/j.neucom.2010.01.017.

Dawson, C., & Wilby, R. (2001). Hydrological modelling using artificial neural networks. *Progress in Physical Geography, 25*(1), 80–108.

Efendigil, T., Önüt, S., & Kahraman, C. (2009). A decision support system for demand forecasting with artificial neural networks and neuro-fuzzy models: A comparative analysis. *Expert Systems with Applications, 36*(3), 6697–6707.

Elliott, G., & Timmermann, A. (2004). Optimal forecast combinations under general loss functions and forecast error distributions. *Journal of Econometrics, 122*(1), 47–79. http://dx.doi.org/10.1016/j.jeconom.2003.10.019.

Fildes, R., & Kourentzes, N. (2011). Validation and forecasting accuracy in models of climate change. *International Journal of Forecasting, 27*(4), 968–995.

Gardner, E. S. (2006). Exponential smoothing: The state of the art – Part II. *International Journal of Forecasting, 22*(4), 637–666.

Hagan, M. T., Demuth, H. B., & Beale, M. H. (1996). *Neural network design*. MA, Boston: PWS Publishing.

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(10), 993–1001. http://dx.doi.org/10.1109/34.58871.

Haykin, S. (2009). *Neural networks and learning machines*. Pearson Education, Inc..

Hillebrand, E., & Medeiros, M. C. (2010). The benefits of bagging for forecast models of realized volatility. *Econometric Reviews, 29*(5–6), 571–593.

Hippert, H. S., Pedreira, C. E., & Souza, R. C. (2001). Neural networks for short-term load forecasting: A review and evaluation. *IEEE Transactions on Power Systems, 16*(1), 44–55.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks, 4*(2), 251–257. http://dx.doi.org/10.1016/0893-6080(91)90009-T.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks, 2*(5), 359–366. http://dx.doi.org/10.1016/0893-6080(89)90020-8.

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting, 22*, 679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K., & Snyder, R. D. (2008). *Forecasting with exponential smoothing: The state space approach*. Springer.

Hyndman, R. J., Koehler, A. B., Snyder, R. D., & Grose, S. (2002). A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting, 18*(3), 439–454.

Inoue, A., & Kilian, L. (2008). How useful is bagging in forecasting economic time series? A case study of US consumer price inflation. *Journal of the American Statistical Association, 103*(482), 511–522.

Jose, V. R. R., & Winkler, R. L. (2008). Simple robust averages of forecasts: Some empirical results. *International Journal of Forecasting, 24*, 163–169.

Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with Applications, 37*(1), 479–489.

Koning, A. J., Franses, P. H., Hibon, M., & Stekler, H. O. (2005). The M3 competition: Statistical tests of the results. *International Journal of Forecasting, 21*(3), 397–409.

Kourentzes, N., & Crone, S. F. (2010). Frequency independent automatic input variable selection for neural networks for forecasting. In *The 2010 international joint conference on neural networks (IJCNN)*, 1098-7576 (pp. 1–8). IEEE. http://dx.doi.org/10.1109/IJCNN.2010.559663.

Kourentzes, N., Petropoulos, F., & Trapero, J. R. (2014). Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting, 30*(2), 291–302. http://dx.doi.org/10.1016/j.ijforecast.2013.09.006.

Kunsch, H. R. (1989). The jackknife and the bootstrap for general stationary observations. *The Annals of Statistics, 17*(3), 1217–1241.

Langella, G., Basile, A., Bonfante, A., & Terribile, F. (2010). High-resolution space–time rainfall analysis using integrated ANN inference systems. *Journal of Hydrology, 387*(3), 328–342.

Lee, T.-H., & Yang, Y. (2006). Bagging binary and quantile predictors for time series. *Journal of Econometrics, 135*(1), 465–497.

Makridakis, S., Andersen, A., Carbone, R., Fildes, R., Hibon, M., Lewandowski, R., et al. (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting, 1*(2), 111–153. http://dx.doi.org/10.1002/for.3980010202.

Makridakis, S., & Hibon, M. (2000). The M3-competition: Results, conclusions and implications. *International Journal of Forecasting, 16*(4), 451–476.

Makridakis, S., & Winkler, R. L. (1983). Averages of forecasts: Some empirical results. *Management Science, 29*(9), 987–996. http://dx.doi.org/10.1287/mnsc.29.9.987.

Mallows, C. (1991). Another comment on ocinneide. *The American Statistician, 45*, 257.

McAdam, P., & McNelis, P. (2005). Forecasting inflation with thick models and neural networks. *Economic Modelling, 22*(5), 848–867.

McNees, S. K. (1992). The uses and abuses of 'consensus' forecasts. *Journal of Forecasting, 11*, 703–711.

Ming Shi, S., Da Xu, L., & Liu, B. (1999). Improving the accuracy of nonlinear combined forecasting using neural networks. *Expert Systems with Applications, 16*(1), 49–54.

Naftaly, U., Intrator, N., & Horn, D. (1997). Optimal ensemble averaging of neural networks. *Network: Computation in Neural Systems, 8*, 283–296.

Newbold, P., & Granger, C. W. J. (1974). Experience with forecasting univariate time series and the combination of forecasts. *Journal of the Royal Statistical Society. Series A (General), 137*(2), 131–165.

Pattie, D. C., & Snyder, J. (1996). Using a neural network to forecast visitor behavior. *Annals of Tourism Research, 23*(1), 151–164.

Roebber, P. J., Butt, M. R., Reinke, S. J., & Grafenauer, T. J. (2007). Real-time forecasting of snowfall using a neural network. *Weather and Forecasting, 22*(3), 676–684.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland

(Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Learning internal representations by error propagation* (Vol. 1. 0-262-68053-X, pp. 318–362). Cambridge, MA, USA: MIT Press.

Silverman, B. W. (1981). Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society. Series B (Methodological), 43*(1), 97–99.

Silverman, B. (1998). *Density estimation for statistics and data analysis*. London: Chapman & Hall/CRC.

Stock, J. H., & Watson, M. W. (2004). Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting, 23*(6), 405–430. http://dx.doi.org/10.1002/for.928.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting, 16*(4), 437–450. http://dx.doi.org/10.1016/S0169-2070(00)00065-0.

Taylor, J. W., & Buizza, R. (2002). Neural network load forecasting with weather ensemble predictions. *IEEE Transactions on Power Systems, 17*(3), 626–632.

Trapero, J. R., Kourentzes, N., & Fildes, R. (2012). Impact of information exchange on supplier forecasting performance. *Omega, 40*(6), 738–747.

Versace, M., Bhatt, R., Hinds, O., & Shiffer, M. (2004). Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks. *Expert Systems with Applications, 27*(3), 417–425.

Wand, M. P., & Jones, M. C. (1995). *Kernel smoothing*. Chapman & Hall.

Werbos, P. J. (1990). Backpropagation through time – What it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550–1560.

Yu, L., Wang, S., & Lai, K. K. (2008). Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics, 30*(5), 2623–2635.

Zhang, G. P. (2001). An investigation of neural networks for linear time-series forecasting. *Computers and Operations Research, 28*(12), 1183–1202.

Zhang, G. P., & Berardi, V. L. (2001). Time series forecasting with neural network ensembles: An application for exchange rate prediction. *Journal of the Operational Research Society, 52*, 652–664.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35–62. http://dx.doi.org/10.1016/S0169-2070(97)00044-7.

Zhang, G. P., Patuwo, B. E., & Hu, M. Y. (2001). A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers and Operations Research, 28*(4), 381–396.

Zhang, G. P., & Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research, 160*(2), 501–514.