A Novel Performance-Based Time Series Forecast Combination Method and Applications with Neural Networks

M. Burak Erturan
General Directorate of State Hydraulic Works, Directorate of Region 13, Antalya, Türkiye

Abstract—**Performance-based forecast combination approaches determine the weights of the individual forecasts based on the inverse average error for a past time interval. However, although the performances are calculated for a time span, the aim is mostly a one-step-ahead time-point forecast. In these classical methods, a relatively higher prediction error of a single past time-point spreads and decreases the performance value of the model, even though the model is highly successful on other time-points in the interval. In this study, a novel approach is presented where performance of each past time-point prediction is calculated separately. Instead of taking the inverse average error for a pre-determined past time interval, prediction performance is calculated for each past data point separately using the normalized inverse absolute error, then the average performances are calculated for past time interval to get the combination weights. To be able to measure the performance of the presented methodology, it is applied on three well-known time series data. Seven different models of neural networks, based on multi-layer perceptron and extreme learning machines are used to model, forecast and form the combination forecasts. Moreover, four different performance-based combination techniques, two central tendency-based benchmark combination methods and the naïve model are employed for comparison. The obtained results show that proposed methodology is a powerful and robust technique and superior to all performance-based combination techniques compared.**

Keywords—Combination forecast; performance-based combination; neural networks; multi-layer perceptron; extreme learning machine

I. INTRODUCTION

Time series forecasting has become a major research area in academics. Researchers have developed various methodologies to improve the accuracy of forecasting. From classical statistical models to machine learning [22] and heuristic models, many models have been developed. Moreover, researches showed that using more than one forecast at the same time often gives better results than a single forecast [1]. This development has led the researchers to hybridize or combine different forecasts and resulted with hybrid and combination forecasting models.

Combination forecast is done by combining multiple forecasts from different models and is usually better than a single forecast. The interested real world time series data is probably cannot be explained by a specific model, because of the consisting different processes like time-varying trends, changing seasonality and structural breaks [1]. Moreover, selecting the "best model" is somewhat problematic because of the sample, parameter and model uncertainties. Sample uncertainty is the problem that different sample sizes result different models and various parameter estimates. Parameter estimation uncertainty arises from different algorithms and various model setups. Furthermore, different model structures result with various parameter restrictions, which may cause for example the estimation problem [2]. Under these circumstances, combining multiple forecasts has been a highly preferred methodology.

The main question of the forecast combination is that, "how to combine the individual forecasts". Different answers to this question result various combination methods. An important class of combination procedures is performance-based forecast combination. That class of methods determine the weights of individual forecasts according to corresponding models' past performances. Some models use inverse past accuracy measures of the models like mean squared error (MSE), root mean squared error (RMSE) or symmetric mean absolute percentage error (sMAPE) to calculate the weights, such that the forecast of the model with a low error measure has a high weight in the forecasting pool. Another idea is, instead of using inverse error measures directly, ranking the models according to their past accuracy and employing the inverse of these rank values as weights, which is called rank-based forecast combination.

There are some characteristics of current performance-based combination methodologies arise from the mean error metrics they are based on. In inverse error based combination methods, average of the errors for some past time span is used. In these methods, the averaging procedure gives general information about the time span performance, but not enough information about the performances for past time-points individually. For example, a possible high error value of only one past prediction might cause a low performance; even other past predictions are highly accurate. On the other hand, rank-based model assigns the weights through the inverse sequence of numbers. Regardless of how much the variation between the models' past accuracies, the weights are pre-determined and constant. Therefore, rank-based combinations limit the weights to only a set of possible values [3].

Forecasting techniques are (including forecast combinations) used mostly for one-step ahead forecasting. Even when the forecast target consists of multi-steps ahead time-points, usually, forecasting is done step by step. It means that, first the next step is forecasted, and then the subsequent forecast is done using previous forecast as if it is the real value. Therefore, pointwise performance values of the forecasting models are highly important. Classical performance-based combination techniques take an average error value of the individual models for a past time-span (for the whole past training data-span, or a more recent one) and use it to assign the performance values. In this paper, a novel performance-based combination methodology is proposed, which is different from classical performance-based methods. First, the performances of each past time-point prediction of individual models are calculated separately by inverse absolute error (IAE). Then, these point performance values are normalized by min-max normalization within other models for each past time point data; such that, the best performance value is one, the worst is zero and the others are in between. The min-max normalization among different models assigns a relative performance value and holds equal performance scales ([0, 1]). Moreover, since recent data holds more information and is more significant, the weights are calculated by averaging the time-point performances of recent data by the rolling window method which is often used in the literature [1].

To test the performance of the proposed model, three well-known time series data are used for modeling and forecasting purposes. These datasets are Canadian lynx, Wolf's sunspot and GBP/USD foreign exchange series. Seven individual neural network models are constructed for combination. Three of the neural network models are based on multi-layer perceptron (MLP) and four of them are different sub-models of extreme learning machines (ELM). Obtained results from proposed combination technique are compared with the results of seven individual models, four other performance-based models, equally weighted simple mean/ median combinations and the naïve model.

The rest of the paper is organized as follows: In Section II, a review of literature related to current is presented. Section III describes all methodologies used in the research. First, MLP and ELM models are explained, which are used for individual forecasts. Then, performance-based combination models are described, which are used for comparison purposes. After that, proposed forecast combination model is introduced and explained. Lastly, the details of the data, models and the software used for application purposes are presented. Section IV presents the application results and findings. Section V concludes with discussions of the study's implications, limitations, and future work.

II. Related Work

Since the work of Bates and Granger [4], forecast combination literature has been grown substantially. From simple combination procedures to much more complex systems, researchers have developed many models and methods. However, forecast combination studies show that simpler combination methods are very successful and hard to beat [1]. The most popular one is taking the simple arithmetic mean of the individual forecasts, ignoring the past information of the models. This procedure is quite successful, robust and easy to perform [3]. Other simple and robust methodologies related to central tendency (CT) are taking the median or the mode of the forecasting pool. In contrast to the simplicity of the mean and median, mode combination needs numerous forecasts or some techniques like discretizing the data or kernel density estimation [5]. Therefore, the simple mean and the median combinations are the most referenced CT combinations, and there is still debate among the scholars whether the mean or the median is more successful [1]. Other than simple equally weighted average model, some researchers applied the trimmed or winsorized mean for combination forecasts. Results show that especially when there are high variations among the forecasts, trimmed and winsorized mean methods produce good results [5]. Simple combination rules are very successful and easy to implement. Therefore, among other and more sophisticated methods, simple combination methods are the choice of many researchers as a benchmark for testing new combination methodologies. There are other more complex methods than simple combinations, such as linear combinations, nonlinear combinations and combining by learning. A comprehensive classification of combination methods can be found in study [1].

Linear combination methods assign the weights of individual forecasts according to their accuracy in a linear combination. Optimal weights approach aims to minimize the variance by optimizing the weights [6], while in linear regression based methods this optimization is done by regression [7]. In criteria based techniques, weights of the individual forecasts are determined by using information criteria, such as Akaike's (AIC) or Bayesian (BIC) information criteria [8, 9].

Performance-based forecast combinations are in the class of linear combination methods, where each individual forecast has a weight according to its performance on past data. Differences in performance measurement are resulted with different models in the literature. In study [10], IRMSE (inverse root mean squared error) model is used with different combination models to forecast day-ahead spot electricity prices. It is found that, while IRMSE method is one of the best performer models, there is no single model dominating the others for all datasets. In the work of [11] different forecasting models are combined with weights determined by their sMAPE values. It is determined that the selection of the forecasting models is crucial for the combination forecast to be successful. Another performance-based model is rank-based model used in [12]. Rank based models determine the weights of individual models according to inverse of their rankings. This model is also used by study [13] for tourism demand forecasting and compared with other combination methods. Performance-based combinations are used to forecast various time series, such as livestock prices [14], oil prices [15], ozone concentration, and airline passengers [16].

III. Methodology

A. Multilayer Perceptron

Artificial Neural Networks (ANN) are computational models inspired by the human brain's neural networks. An ANN is composed of interconnected nodes (neurons) organized in layers; including input, hidden, and output layers. They mimic the way real neurons communicate; and are particularly useful in pattern recognition, classification, optimization, and time series prediction. ANNs can model complex nonlinear relationships and handle large datasets with numerous variables.

A multi-layer perceptron (MLP) is the most commonly used type of artificial neural network that consists of multiple layers of nodes; including an input layer, one or more hidden layers, and an output layer. The network uses a feedforward path for data processing and backpropagation for training and optimization. Training MLPs involves challenges such as entrapment in local minima, convergence speed, and sensitivity to initialization. MLPs have strong generalization capabilities but are prone to overfitting, where the network becomes too dependent on the training data. Techniques such as regularization and cross-validation are often employed to mitigate overfitting. MLPs are widely used for time series modeling. As a mathematical expression for time series y_t, the one-step ahead forecast $\hat{y}_{(t+1)}$ of the real data y_{t+1} can be calculated as [5]:

$$\hat{y}_{t+1} = \beta_0 + \sum_{l=1}^{L} \beta_l g \left(b_i + \sum_{i=1}^{I} w_{li} x_i\right) \tag{1}$$

In Eq. (1), I is the number of the inputs x_i, and L is the number of hidden nodes. β and w are the weights of the output and hidden layers respectively, where $\beta=[\beta_1, \beta_2, ..., \beta_L]$ and $w=[w_1, w_2, ..., w_L]$. β_0 and b_i are the biases acting like intercept in the regression process. Lastly, $g(.)$ is the transfer function and is usually hyperbolic tangent or logistics function.

The backpropagation algorithm is a widely used method for training MLPs by adjusting the weights of the connections to minimize the error in predictions. There are three stages of backpropagation, in the first stage the calculations are done forward through the final output by using activation function. Then, obtained error is propagated backwards, starting from the output layer. Finally, the weights are changed in order to minimize errors. These stages continue until certain conditions are reached. Backpropagation is effective but can be slow and may be stuck in local minima [17]. Another training algorithm is resilient backpropagation (RPROP), which is an advanced training algorithm designed to improve the efficiency and performance of MLPs by addressing some of the limitations of traditional backpropagation methods. RPROP is an advanced version of backpropagation that adjusts the weight updates based on the sign of the gradient rather than its magnitude, leading to faster convergence and improved performance [17, 18]. As an improvement of RPROP, a weight backtracking mechanism can be added. Weight backtracking retracts a previous update for some or all weights, while whether taking back a step or not is decided by heuristics [19].

B. Extreme Learning Machine

Extreme learning machine (ELM) is a single layer feedforward network (SLFN). Different from other neural networks, the weights of connections between input and hidden layers are selected randomly and not trained further. Therefore, instead of training all the weights in the network, only the output weights (connecting hidden and output layer) are calculated. The output weights can be optimized with the approaches like least squares or equivalents. ELM has fast learning rate with strong generalization performance. Moreover it shows high efficiency in training, particularly when data is limited [20, 21]. A representation of ELM architecture is in Fig. 1.

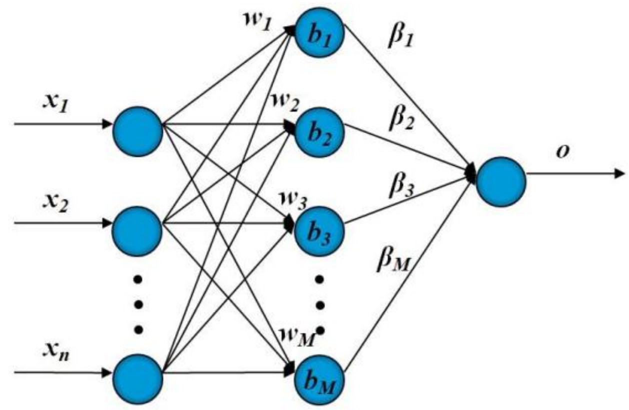


Fig. 1. Architecture of an extreme learning machine network with n input, M hidden, and 1 output nodes.

Suppose we have N arbitrary distinct training samples (x_j, t_j), each containing of n inputs and m outputs where $j = 1, 2, ..., N$. Therefore, $x_j = [x_{j1}, x_{j2}, ..., x_{jn}]^T \in Rn$ is the input vector and $tj = [tj1, tj2, ..., t_{jm}]^T \in$ Rm is the output vector. The output (o_j) of the standard SLFN model with M hidden nodes and the activation function of $g(.)$ can be modeled as [20]:

$$o_j = \sum_{i=1}^{M} \beta_i g \left(w_i x_j + b_i\right) \tag{2}$$

In Eq. (2), $w_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ is the weight vector connecting inputs to the ith hidden node. $\beta_i = [\beta_{i1}, \beta_{i2}, ..., \beta_{in}]^T$ the weight vector connecting ith hidden node to outputs, and bi is the bias (threshold) of ith hidden node. The standard SLFN with M hidden nodes with activation function $g(.)$ can approximate these N samples with zero error means that:

$$\sum_{i=1}^{M} \beta_i g \left(w_i x_j + b_i\right) = t_j \tag{3}$$

where $j = 1, 2, ..., N$. These N equations can be written in a more compact form as:

$$\boldsymbol{H}\beta = \boldsymbol{T} \tag{4}$$

where,

$$\boldsymbol{H} = \begin{bmatrix} g(w_1 x_1 + b_1) & \dots & g(w_M x_1 + b_M) \\ g(w_1 x_2 + b_1) & \dots & g(w_M x_2 + b_M) \\ \vdots & \ddots & \vdots \\ g(w_1 x_N + b_1) & \dots & g(w_M x_N + b_M) \end{bmatrix}_{NXM} \tag{5}$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \beta_2^T \\ \vdots \\ \beta_n^T \end{bmatrix}_{Mxm} \tag{6}$$

$$\boldsymbol{T} = \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_N^T \end{bmatrix}_{Nxm} \tag{7}$$

After setting the input weights and biases of the hidden layer, the output weight vector β can be obtained by a series of linear equations transformations. If the number M of hidden nodes is equal to the number N of distinct training samples, $M = N$, matrix H is square and invertible. Because the input weight vectors wi and the hidden biases b_i are randomly chosen, the SLFN can approximate these training samples with zero error. However, in most cases the number of hidden nodes is much less than the number of distinct training samples, such that $M << N$. Then, H is a not a square matrix and there may not exist w_i, b_i, β_i ($i = 1, 2, ...,M$) such that $\boldsymbol{H\beta} = \boldsymbol{T}$. Then, the smallest norm least squares solution of the above linear system is [20]:

$$\hat{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \tag{8}$$

where, $\boldsymbol{H}^{\dagger}$ represents the generalized (Moore-Penrose) inverse matrix of the output.

Different from traditional neural networks, parameters between hidden and output layer in the ELM structure can be obtained by minimizing least squares solutions with the provided training data [23, 24, 25]. Therefore, using ELM to obtain the output weights is consisted of three steps. First, randomly selecting numerical values between zero and one to set input weights and the bias of the hidden layer. Second, calculation of the output matrix $\boldsymbol{H}$. Finally, calculation of the output weights β. On the other hand, there are some possible problems with ELM networks. First, ELM can effectively approximate to any complex systems with huge numbers of hidden nodes, but in this case, the generalization might decrease. Another is that, the least square method cannot provide good estimates for weights when there exist many outliers in data. To overcome these deficiencies, researchers presents some developments like regularization techniques. Some main methods for this purpose are lasso, ridge and stepwise regression models.

Generally, in regression models a common target is keeping the model as simple as possible. Regularization models like LASSO (Least absolute shrinkage and selection operator) and Ridge regressions punish the complex model by adding a penalty term to the cost function. Ridge regression adds squared magnitude of the coefficients as penalty term, while LASSO adds absolute value of the coefficients as penalty term to the cost function. LASSO shrinks the less important variable coefficients to zero and causes a less complex structure [21]. Moreover, stepwise regression uses forward adding and backward deleting method to the variables to find the best fitting combination of independent variables for prediction [26].

C. Performance-Based Forecast Combinations

Performance-based forecast combination is a method used to improve the accuracy of predictions by linearly combining multiple individual forecasts by assigning weights to each forecast based on their past performances. This combination procedure ignores the correlations among individual forecasts. Moreover, the success of these models reconfirms that models ignoring correlations are more successful than other models, because correlations are poorly estimated in practice and should be ignored in weight calculations [1, 3].

One common branch of performance-based methodologies is inverse error technique, which uses inverse error values of the models directly as weights. Commonly used error metrics are mean squared error (MSE), root mean squared error (RMSE) and symmetrical mean absolute percentage error (sMAPE). In inverse error model, the weight of the ith model among N models with $j=[1,2,...N]$ is:

$$w_i = \frac{\varepsilon_i^{-1}}{\sum_{j=1}^{N} \varepsilon_i^{-1}} \tag{9}$$

where, ε is the error value of the model and might be MSE, RMSE or sMAPE according to which error metrics the combination is based on. Then the combined forecast for time t with N individual model and individual forecasts $\hat{y}_{tj}$ is:

$$\hat{y}_t = \sum_{j=1}^{N} \hat{y}_{tj} w_j \tag{10}$$

Note that, Eq. (9) is valid under the assumption of independence of errors. In the case of error-dependence (i.e. nested models), one of the models encompasses the other and the optimal combining weights are trivially either zero or one [27].

Another performance-based technique is combination forecasting with inverse rank based weights, which involves integrating multiple individual forecasts into a single forecast by assigning weights to each forecast based on their past performance with a ranking system. Proposed by [12], in the rank-based methodology, the weights of the individual forecasts are determined by the inverse of the ranks of the models, which are identified according to their past performance (usually MSE). The weight of the ith model among N models with $j=[1,2,...N]$ is:

$$w_i = \frac{Rank_i^{-1}}{\sum_{j=1}^{N} Rank_i^{-1}} \tag{11}$$

where, $Rank_i=[1,2,...,N]$ is the rank of the ith model's forecast according to model's past performance (MSE).

D. Proposed Combination Methodology

Performance determination with inverse error metrics uses an average value of performance for a pre-determined past time period. It is impossible to determine a point by point performance from these mean (average) error values. For example, a very bad performance with a high error value of a model in a past single point effects the whole performance no matter how successful the model in other past data points. On the other hand, rank-based methods not just use the mean error values, but also assign the performances of the models with a pre-determined values; which ignores the real performance differences between the competitive models. Furthermore, most of the forecasting researches are interested in one-step ahead forecasting for a multi-step period, including this study. Therefore, measuring the performance of each past time point separately can give more reliable values if forecasting procedure is one-step ahead.

Presented methodology takes past prediction performances into account separately for each time-point. First, inverse of absolute error (IAE) values for each past point predictions are calculated. Then, for each past data point, competitive models' IAE values are normalized within. A min-max normalization is applied such that the best prediction takes the value of one, the worst prediction is zero and the others are in between. The min-max normalization provides an equal performance scale

([0, 1]) for each past prediction. Obtained values are the performances of the models on past data points. Finally, the weights of the models are calculated by averaging the past performances. Proposed methodology takes neither the whole training set values nor just the values of test set into account. A rolling window method is used to take into account of the recent values whether they are training or test data. In this paper, a rolling window of 10 past predictions is used. Mathematical notation of the presented weight procedure is presented in Eq. (12).

$$p_{i,t} = NORM\left(AE_{i,t}^{-1}\right) = \frac{AE_{i,t}^{-1} - \min\left(AE_t^{-1}\right)}{\max\left(AE_t^{-1}\right) - \min\left(AE_t^{-1}\right)} \tag{12}$$

In Eq. (12), $p_{i,t}$ represents the performance value of the ith candidate model for time t. *NORM(.)* is the min-max normalization function, where AE^{-1} is the inverse absolute error. Then, the weights and final combination forecast for time t is:

$$w_i = \frac{\sum_{j=t-1}^{t-Z} p_{i,j}}{Z} \tag{13}$$

$$\hat{y}_t = \sum_{j=1}^{N} \hat{y}_{tj} w_j \tag{14}$$

where, Z is the number of past data to be used for weight wi of the ith model forecast, N is the number of the candidate models with individual forecasts of ytj, and j=[1,2,…N].

The usage of the proposed combination method can be summarized in three steps:

- The absolute error values for each past predictions of an individual model are calculated separately. The inverse of the AEs are normalized (min-max normalization) within other models for the same time points to get the time point performances of the models.
- For each model, time-point performances for Z past prediction are averaged for pre-determined time period to find the weights of the individual models.
- Combination forecast is found using individual forecasts and related weights.

The workflow of the proposed combination model used in this study is presented in Fig. 2.

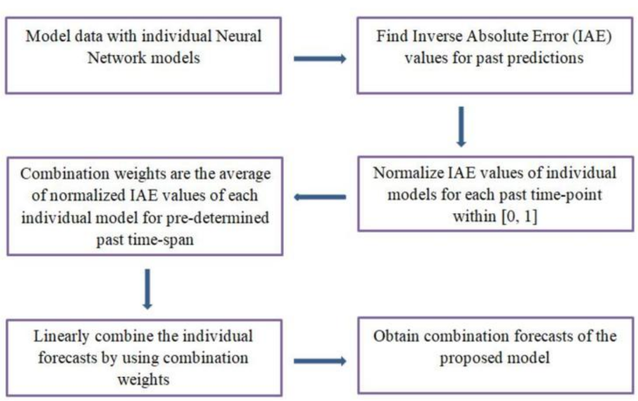


Fig. 2. The workflow of the proposed forecast combination model used in this study.

Proposed methodology has same characteristics with other performance-based models. It ignores the correlations among individual models like other performance-based models. It is a linear combination method using weights of the individual models, which are calculated by models' past performances. Proposed model's difference from other performance-based models is the calculation procedure of the past performances.

E. Data and Application

In this study, three well-known data sets in time series literature are used for the application purposes. They are Canadian lynx, Wolf's sunspot and British Pound/United States Dollar (GBP/USD) exchange rate data sets. All three datasets have various characteristics, and are used often in time series literature as benchmark datasets [28].

Canadian lynx dataset shows the number of yearly trapped lynx around North Canada Mackenzie River between 1821 and 1938. It has total 114 data points and is an important dataset in time series literature with nonlinear characteristics [29]. Wolf's sunspot data, known with nonlinear behavior, contains yearly sunspot numbers in the period of 1700-1987 and has 288 data points. Sunspot data is identified as nonlinear and non-Gaussian and used to test nonlinear models for their performances [28]. Lastly, GBP/USD dataset includes weekly values between 1821 and 1934, and is composed of 731 data points. Exchange rate, and in general, financial time series forecasting is a very difficult task. Various models are developed but few are successful to beat a random walk (naïve) model [29].

For the sake of consistency with the related literature [28-31] the logarithmic transformation with base 10 (log10) of Canadian lynx, and natural logarithmic transformation (ln) of GBP/USD exchange rate data sets are used in the application stage. Sunspot and transformed lynx datasets are stationary, while transformed exchange rate time series is nonstationary. As can be seen from the related literature [28-31], if a conventional time series model like ARIMA (Autoregressive integrated moving averages) to be used, this dataset should be integrated (differenced) with order 1 to be stationary. However, as neural networks can model nonstationary data directly, natural logarithmic transformation of the exchange rate data is used directly to be consistent with the aforementioned literature. On the other hand, successful modeling of nonstationary data is very difficult with respect to stationary data for all conventional and advanced time series models.

In Fig. 3 to 5, graphical representation of the Canadian lynx, Wolf's sunspot and GBP/USD datasets are presented respectively. In Fig. 3 and Fig. 5 the real data and log transformed data used could be seen separately. In Table I, descriptive statistics of the datasets are presented.

Seven different neural network models are used for modelling and forecasting purposes. These models are used for proposed combination forecast methodology and for comparison. Three of the models are in MLP class: MLP with classical backpropagation (MLP-BP), MLP with resilient backpropagation (MLP-RP) and MLP with resilient backpropagation and weight backtracking (MLP-RPB). Other four models are various ELM models: ELM with LASSO regression (ELM-L), ELM with ridge regression (ELM-R), ELM with stepwise regression (ELM-S) and ELM with classical linear regression (ELM-LN). Other than these seven models, six different combination models are used for comparison purposes. Two of them are benchmark combination models: Classical equally weighted simple mean (C-MEAN), and median (C-MEDIAN) models. Last four models are other performance-based combination models: Combination models with inverse MSE (C-MSE), RMSE (C-RMSE) and sMAPE (C-SMAPE) based weights, and rank-based combination model (C-RANK). In addition, the naïve model (N) is added to the pool of the models for comparison.

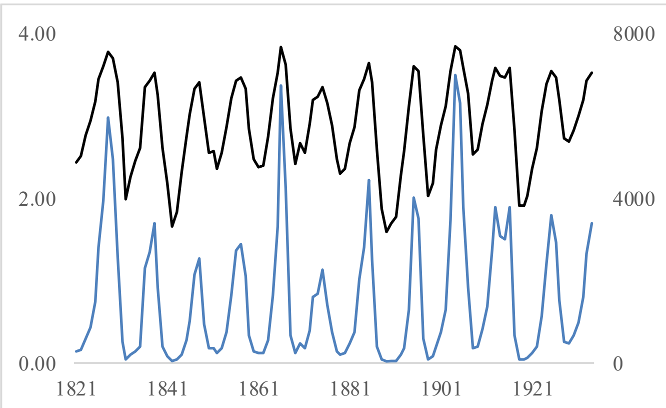


Fig. 3. Canadian lynx dataset. Real values (blue, right axis) and logarithmic transformed (log10) values (black, left axis).

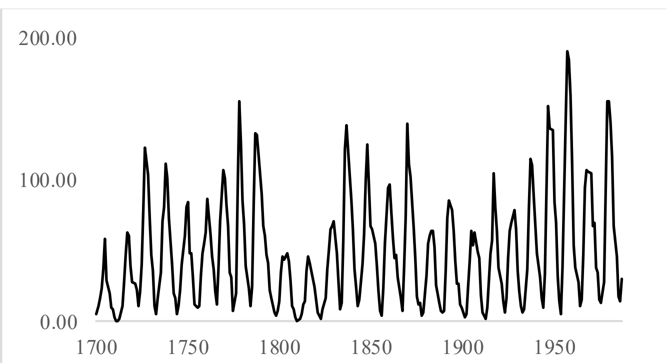


Fig. 4. Wolf's sunspot dataset.

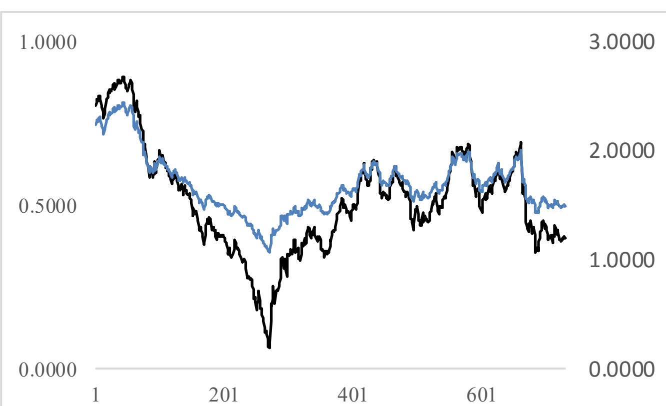


Fig. 5. GBP/USD exchange rate dataset. Real values (blue, right axis) and natural logarithmic transformed (ln) values (black, left axis).

TABLE I. DESCRIPTIVE STATISTICS OF THE DATASETS

	Canadian lynx (Log 10)	**Wolf's sunspot**	**GBP/USD (Ln)**
Data Number	114	288	731
Min	1.59	0.00	0.0628
Max	3.84	190.20	0.8909
Median	2.89	39.00	0.5043
Mean	2.90	48.43	0.5089
Std. Dev.	0.56	39.36	0.1616

Software used for this study is R Statistical Software version 4.4.1 [32]. Package nnfor [33] is used mainly with neuralnet [34] package. nnfor package is used for time series modeling and forecasting with neural networks. All seven neural networks are three layered with a single hidden layer and all activation functions are hyperbolic tangent (tanh). Since the datasets are well known and employed many times by other researchers, network architectures and training-test data compositions are determined to be compatible with the literature. MLP model architecture is 7-5-1 for lynx dataset, 4-4-1 for sunspot dataset, and 7-6-1 for GBP/USD data [28, 29, 31, 35, 36]. An exception is the number of hidden nodes in ELM networks. The nnfor package assigns the number of hidden neurons for ELM models automatically. ELMs start with a very large hidden layer (100 nodes), and then prune it as much as needed [33]. Therefore, number of the input nodes are 7, 4 and 7 for lynx, sunspot and exchange rate data sets respectively. Number of the hidden nodes are assigned by the nnfor package. Each individual neural network model is retrained 20 times and combined with mean values for a more robust forecast. Moreover, all forecasting procedures are one step forecasting. Forecast performances of the models are compared using MSE, MAD (Mean absolute deviation) and MAPE (Mean absolute percentage error) error metrics.

IV. RESULTS

Canadian lynx data is composed of 114 data points. First 100 data points are used as training set, while last 14 data points are test set to be forecasted. The architecture of all MLP models are 7-5-1. On the other hand, ELM-L/ELM-R models are both 7-91-1 and ELM-S/ELM-LN model architectures are 7-40-1. Obtained results are tabulated in Table II.

Results show that proposed model and C-RMSE model are superior to all other performance-based models. In terms of MAD and MAPE, presented model has the best results among other performance-based models. However, equally weighted mean and median combinations are more successful than proposed and other performance-based combinations. Among the individual models, ELM with LASSO regression model gives the best results. But still, in terms of MSE, C-MEAN combination model is superior to all individual neural network models. An important finding is that, all individual and combination models are better than the naïve model.

Wolf's sunspot dataset includes 288 data points with 221 of the data is allocated as training set, while last 67 data is formed as test set. MLP model architectures are 4-4-1, ELM-L/ELM-R models are 4-100-1 and ELM-S/ELM-LN models are 4-40-1. Performance results of the models are in Table III.

TABLE II. FORECASTING RESULTS OF CANADIAN LYNX DATASET

Forecasting Model		MSE ($x10^{-2}$)	MAD ($x10^{-1}$)	MAPE
Individual Models	MLP-BP	2.5993	1.2961	4.48%
	MLP-RP	2.5859	1.2501	4.31%
	MLP-RPB	2.5366	1.2963	4.49%
	ELM-L	2.3318	1.2739	4.27%
	ELM-R	4.5764	1.9828	6.44%
	ELM-S	2.5889	1.4073	4.81%
	ELM-LN	2.9211	1.4732	5.06%
CT Based Combinations	C-MEAN	2.2895	1.2918	4.37%
	C-MEDIAN	2.3900	1.2892	4.42%
Performance Based Combinations	C-MSE	2.4439	1.3430	4.59%
	C-RMSE	2.3740	1.3250	4.51%
	C-RANK	2.4741	1.4131	4.63%
	C-SMAPE	2.3921	1.3266	4.53%
	Proposed Model	**2.3907**	**1.3214**	**4.51%**
	Naïve	6.8734	2.3088	7.77%

TABLE III. FORECASTING RESULTS OF WOLF'S SUNSPOT DATASET

Forecasting Model		MSE ($x10^{2}$)	MAD ($x10^{1}$)	MAPE
Individual Models	MLP-BP	2.9186	1.3089	35.44%
	MLP-RP	2.8160	1.2843	33.60%
	MLP-RPB	2.8085	1.2909	34.59%
	ELM-L	3.2518	1.4155	40.03%
	ELM-R	3.5500	1.4817	42.90%
	ELM-S	2.7776	1.2724	34.55%
	ELM-LN	2.7846	1.2925	35.45%
CT Based Combinations	C-MEAN	2.7681	1.2935	35.57%
	C-MEDIAN	2.8138	1.2958	35.28%
Performance Based Combinations	C-MSE	2.7594	1.2888	35.35%
	C-RMSE	2.7629	1.2910	35.46%
	C-RANK	2.7832	1.2903	35.22%
	C-SMAPE	2.7689	1.2933	35.52%
	Proposed Model	**2.7353**	**1.2836**	**35.22%**
	Naïve	9.2073	2.2964	54.84%

Results presented in Table II shows that proposed combination model is the best model among all other combination models. On the other hand, in terms of MAD and MAPE error metrics, two different individual models show best performances. MLP-RP and ELM-S models seems to be the best models among individual models. Also, it can be seen that, performance-based combination methods have slightly better performances than central tendency based combinations. Naïve model has the worst performance among all competent models with significantly high error metrics.

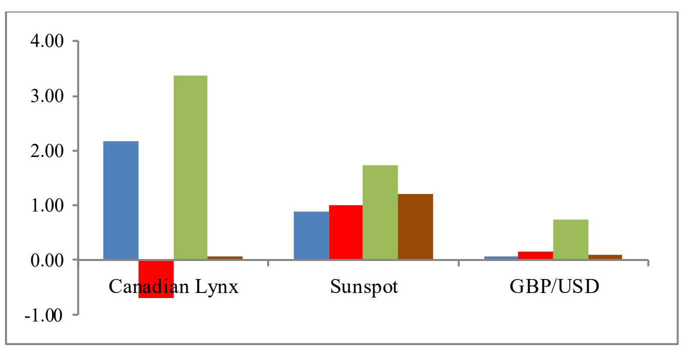
731 data points of GBP/USD exchange is divided to two as training and test sets, such that 679 data points are training and 52 data points are test data. MLP models are in 7-6-1 architecture, whereas, ELM-L/ELM-R models are 7-100-1, and ELM-S/ELM-LN models are 7-40-1. Obtained results are in Table IV.

TABLE IV. FORECASTING RESULTS OF GBP/USD DATASET

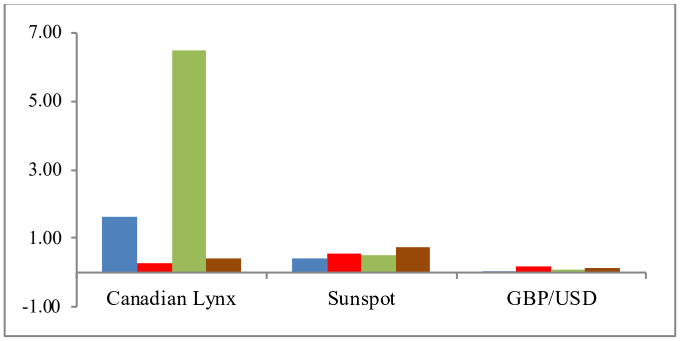
Forecasting Model		MSE ($x10^{-4}$)	MAD ($x10^{-2}$)	MAPE
Individual Models	MLP-BP	1.9434	1.1342	2.82%
	MLP-RP	1.9025	1.1216	2.79%
	MLP-RPB	1.8855	1.1028	2.74%
	ELM-L	1.8297	1.0972	2.72%
	ELM-R	1.8300	1.0973	2.72%
	ELM-S	1.8563	1.1144	2.77%
	ELM-LN	1.9003	1.1261	2.79%
CT Based Combinations	C-MEAN	1.8341	1.1020	2.74%
	C-MEDIAN	1.8391	1.1029	2.74%
Performance Based Combinations	C-MSE	1.8315	1.0995	2.73%
	C-RMSE	1.8329	1.1008	2.73%
	C-RANK	1.8436	1.0998	2.73%
	C-SMAPE	1.8321	1.1004	2.73%
	Proposed Model	**1.8301**	**1.0990**	**2.73%**
	Naïve	1.8298	1.1010	2.73%

Results show that almost all individual models and combinations show high performances. All combination models and the naïve model show very close performance values in all error metrics. Best individual models are ELM-L and ELM-R models, and they are slightly better than the forecast combinations. On the other hand, among all combination models, proposed combination methodology shows the best performance. Another important result drawing attention is the successful performance of the naïve model with respect to its less accurate results on sunspot and lynx datasets.

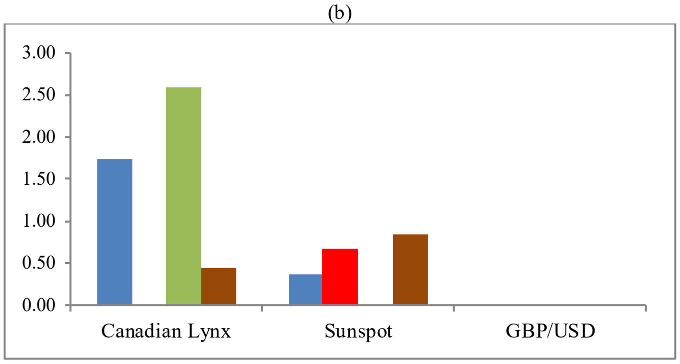
In Fig. 6, the percentage improvement by using the proposed model instead of other performance-based models in terms of error metrics is presented.



(a)



(b)



(c)

Fig. 6. Percentage improvement of the proposed model in terms of Mean Squared Error (a), Mean Absolute Deviation (b) and Mean Absolute Percentage Error (c) from other performance-based models (C-MSE (blue), C-RMSE (red), C-RANK (green) and C-SMAPE (brown).

V. Conclusion

In this study, a novel performance-based forecast combination methodology is presented. The novelty of the method is originated from the calculation of the individual models' past performances. Classical performance-based combination methods use a mean error value (MSE, MAD, sMAPE etc.) for a pre-determined past time-period. Conversely, proposed methodology measures the performance of each past time point separately, and then takes the average of these time point performances for the interested past time period. Furthermore, after computation of inverse absolute error values for each time point, a min-max normalization is applied within other individual models. That procedure causes all models' time-point performance values to be between zero and one, while the best model's is one and the worst model's becomes zero. After that, each model's performances are calculated by averaging the time point performances along the pre-determined time-period. To evaluate the performance of the proposed model, three well-known time series are used. Seven different neural network models based on MLP and ELM are trained. Moreover, simple mean and median combinations, four main performance-based combinations, and naïve model are formed for comparison purposes. Comparisons are done with MSE, MAD and MAPE error metrics.

Obtained forecasting results show that proposed combination methodology is superior to other performance-based combination methods in almost all occasions. Additionally, proposed technique is superior to central tendency based benchmark models of equally weighted mean and median combinations in sunspot and exchange rate datasets. In lynx dataset, mean and median combinations mostly show better performances. Proposed technique increases the forecasting accuracy more than other compared performance-based combination models. It beats the naïve model in lynx and sunspot dataset, whereas the naïve model is slightly better in GBP/USD time series in terms of MSE.

Considering the obtained results, it can be stated that the proposed forecast combination methodology is a very successful and robust performance-based technique. Moreover, findings show that calculating the performance value of an individual model for each past data point separately causes more accurate combination weights than computing directly from an average error value for that past time-span. This contribution to the forecast model performance metrics can be tested and used in further researches. Application to different time series data on various subjects and using different individual time series models can help the generalization of the model. Furthermore, studying with various past time periods for performance evaluation and comparing with different combination models are some other possible research extents. On the other hand, proposed model's performance evaluation is necessarily more time consuming than conventional performance-based methods. However, especially under the circumstances where recent data is used as presented study is, it is tolerable.

References

[1] X. Wang, R. J. Hyndman, F. Li, and Y. Kang, "Forecast combinations: An over 50-year review," Int. J. Forecast, vol. 39, no. 4, pp. 1518-1547, Oct.–Dec. 2023, doi: 10.1016/j.ijforecast.2022.11.005.

[2] N. Kourentzes, D. Barrow, and F. Petropoulos, "Another look at forecast selection and combination: Evidence from forecast pooling," Int. J. Prod. Econ., vol. 209, pp. 226-235, Mar. 2019, doi: 10.1016/j.ijpe.2018.05.019.

[3] V. Genre, G. Kenny, A. Meyle, and A. Timmermann, "Combining expert forecasts: Can anything beat the simple average?," Int. J. Forecast., vol. 29, pp. 108-121, Jan.-Mar. 2013, doi: 10.1016/j.ijforecast.2012.06.004.

[4] J.M. Bates, and C. W. J. Granger, "The combination of forecasts," Oper. Res. Q., vol. 20, no. 4, pp. 451-468, Dec. 1969, doi: 10.1057/jors.1969.103.

[5] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," Expert Syst. Appl., vol. 41, pp. 4235-4244, Jul. 2014, doi: 10.1016/j.eswa.2013.12.011.

[6] A. Timmermann, "Forecast combinations," in Handbook of economic forecasting, vol. I, G. Elliott, C. W. J. Granger, and A. Timmermann, Eds. Elsevier, 2006, pp. 135-196, doi: 10.1016/S1574-0706(05)01004-9.

[7] C. Conflitti, C. De Mol, and D. Giannone, "Optimal combination of survey forecasts," Int. J. Forecast., vol. 31, no. 4, pp. 1096-1103, Oct.-Dec. 2015, doi: 10.1016/j.ijforecast.2015.03.009.

[8] S. Kolassa, "Combining exponential smoothing forecasts using Akaike weights," Int. J. Forecast., vol. 27, no. 2, pp. 238-251, Apr.-Jun. 2011, doi: 10.1016/j.ijforecast.2010.04.006.

[9] F. Petropoulos, N. Kourentzes, K. Nikolopoulos, and E. Siemsen, "Judgmental selection of forecasting models," J. Oper. Manag., vol. 60, pp. 34-46, Jun. 2018, doi: 10.1016/j.jom.2018.05.005.

[10] J. Nowotarski, E. Raviv, S. Trück, and R. Weron, "An empirical comparison of alternative schemes for combining electricity spot price forecasts," Energy Econ., vol. 46, pp. 395-412, Nov. 2014, doi: 10.1016/j.eneco.2014.07.014.

[11] M. Pawlikowski and A. Chorowska, "Weighted ensemble of statistical models," Int. J. Forecast., vol. 36 no. 1, pp. 93-97, Jan.-Mar. 2020, doi: 10.1016/j.ijforecast.2019.03.019.

[12] M. Aiolfi and A. Timmermann, "Persistence in forecasting performance and conditional combination strategies," J. Econom., vol. 135, no. 1, pp. 31-53, Nov.-Dec. 2006, doi: 10.1016/j.jeconom.2005.07.015.

[13] R. R. Andrawis, A. F. Atiya, and H. El-Shishiny, "Combination of long term and short term forecasts, with application to tourism demand forecasting," Int. J. Forecast., vol. 27, no. 3, pp. 870-886, Jul.-Sep. 2011, doi: 10.1016/j.ijforecast.2010.05.019.

[14] L. Ling, D. Zhang, A. W. Mugera, S. Chen, and Q. Xia, "A Forecast Combination Framework with Multi-Time Scale for Livestock Products' Price Forecasting," Math. Probl. Eng., Article ID 8096206, Oct. 2019, doi: 10.1155/2019/8096206.

[15] C. Baumeister and L. Kilian, "Forecasting the real price of oil in a changing world: A forecast combination approach," J. Bus. Econ. Stat., vol. 33, no. 3, pp. 338-351, Jul. 2015, doi: 10.1080/07350015.2014.949342.

[16] J. P. Donate, P. Cortez, G. G. Sanchez, and A. S. De Miguel, "Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble," Neurocomputing, vol. 109, pp. 27-32, Jun. 2013, doi: 10.1016/j.neucom.2012.02.053.

[17] W. Saputra, A. P. Windarto, and A. Wanto, "Analysis of the Resilient Method in Training and Accuracy in the Backpropagation Method," Int. J. Inform. Comput. Sci., vol. 5, no. 1, pp. 52-56, Mar. 2021, doi: 10.30865/ijics.v5i1.2922

[18] M. Riedmiller and H. Braun, "A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm," Proc. IEEE Int. Conf. Neural Netw., vol. 1, pp. 586-591, Mar.-Apr. 1993, doi: 10.1109/ICNN.1993.298623

[19] C. Igel and M. Hüsken, "Empirical evaluation of the improved Rprop learning algorithms," Neurocomputing, vol. 50, pp. 105-123, Jan. 2003, doi: 10.1016/S0925-2312(01)00700-7

[20] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no. 1-3, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126

[21] D. T. Várkonyi and K. Buza, "Extreme learning machines with regularization for the classification of gene expression data," in Proceedings of the 19th Conference Information Technologies-Applications and Theory (ITAT 2019), B. Petra, H. Martin, H. Tomáš, P. Matůš, and R. Rudolf, Eds., CEUR Workshop Proceedings, 2019, pp. 99-103. https://ceur-ws.org/Vol-2473/paper11.pdf.

[22] S. Ding, X. Xu, and R. Nie, "Extreme learning machine and its applications," Neural Comput. Appl., vol. 25, pp. 549-556, Sep. 2014, doi: 10.1007/s00521-013-1522-8.

[23] G. Huang, G. B. Huang, S. Song, and K. You, "Trends in extreme learning machines: a review," Neural Netw., vol. 61, pp. 32-48, Jan. 2015, doi: 10.1016/j.neunet.2014.10.001.

[24] J. Wang, S. Lu, S. H. Wang, and Y. D. Zhang, "A review on extreme learning machine," Multimed. Tools Appl., vol. 81, pp. 41611-41660, Dec. 2022, doi: 10.1007/s11042-021-11007-7.

[25] Y. Yang, H. Zho, Y. Gao, J. Wu, Y. G. Wang, and L. Fu, "Robust penalized extreme learning machine regression with applications in wind speed forecasting," Neural Comput. Appl., vol. 34, pp. 391-407, Jan. 2022, doi: 10.1007/s00521-021-06370-3.

[26] A. Alqahtani, "Engineering the Energy Gap of Cupric Oxide Nanomaterial Using Extreme Learning Machine and Stepwise Regression Algorithms," J. Nanomater., Article ID 4797686, Oct. 2021, doi: 10.1155/2021/4797686.

[27] T. E. Clark and M. W. McCracken, "Combining Forecasts from Nested Models," Oxford B. Econ. Stat., vol. 71, no. 3, pp. 303-329, Apr. 2009, doi: 10.1111/j.1468-0084.2009.00547.x.

[28] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159-175, Jan. 2003, doi: 10.1016/S0925-2312(01)00702-0.

[29] M. Khashei and M. Bijari, "Hybridization of the probabilistic neural networks with feed-forward neural networks for forecasting," Eng. Appl. Artif. Intell., vol. 25, pp. 1277-1288, Sep. 2012, doi: 10.1016/j.engappai.2012.01.019.

[30] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data," Appl. Soft Comput., vol. 23, pp. 27-38, Oct. 2014, doi: 10.1016/j.asoc.2014.05.028.

[31] A. Atesongun and M. A. Gulsen, "Hybrid Forecasting Structure Based on Arima and Artificial Neural Network Models," Appl Sci., vol. 14, Article ID 7122, Aug. 2024, doi: 10.3390/app14167122.

[32] R Core Team, "R: A language and environment for statistical computing v. 4.4.1 [software]," R Foundation for Statistical Computing, 2020, https://www.R-project.org.

[33] N. Kourentzes, "nnfor: Time Series Forecasting with Neural Networks v. 0.9.9 [software package]," R, Nov. 2023, doi: 10.32614/CRAN.package.nnfor.

[34] S. Fritsch, F. Guenther, M. N. Wright, M. Suling, and S.M. Mueller, "neuralnet: Training of Neural Networks v. 1.44.2 [software package]," R, Feb. 2019, doi: 10.32614/CRAN.package.neuralnet.

[35] Z. Hajirahimi and M. Khashei, "Sequence in Hybridization of Statistical and Intelligent Models in Time Series Forecasting," Neural Process. Lett., vol. 54, Oct. 2022, doi: 10.1007/s11063-020-10294-9

[36] R. Adhikari and R. K. Agrawal, "A Homogeneous Ensemble of Artificial Neural Networks for Time Series Forecasting," Int. J. Comput. Appl., vol. 32, no. 7, pp. 1-8, Oct. 2011, doi: 10.48550/arXiv.1302.6210