

Received July 8, 2021, accepted July 27, 2021, date of publication July 30, 2021, date of current version August 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101741

A Dynamic Predictor Selection Method Based on Recent Temporal Windows for Time Series Forecasting

ERAYLSON G. SILVA¹, (Member, IEEE), PAULO S. G. DE MATTOS NETO¹, (Member, IEEE), AND GEORGE D. C. CAVALCANTI¹, (Senior Member, IEEE)

Centro de Informática, Universidade Federal de Pernambuco, Recife 50670-901, Brazil

Corresponding author: Eraylson G. Silva (egs@cin.ufpe.br)

This work was supported in part by the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) (Brazilian Agency), and in part by the Fundação de Amparo a Ciência e Tecnologia de Pernambuco (Brazilian Agency).

ABSTRACT The development of accurate forecasting systems for real-world time series modeling is a challenging task. Due to the presence of temporal patterns that change over time, the adoption of a single model can lead to underperformed forecasts. In this scenario, Multiple Predictor Systems (MPS) emerge as an alternative to adopting single models since they struggle to learn in the presence of temporal patterns that change over time. Dynamic prediction/ensemble selection is a special case of MPS where each model is an expert in the time series's specific patterns. In dynamic selection, instead of combining all models, the most competent models per test pattern are selected. A criterion commonly used is to evaluate the models' performance in the region of competence, formed by the patterns present in the in-sample set (training or validation sets) more similar to the test pattern. Thus, the region of competence's quality is a key factor in the precision of the MPS. However, adequately defining the similarity criterion and the size of the region of competence is challenging and problem-dependent. Furthermore, there is no guarantee that similar data exist in the in-sample set. This paper proposes a dynamic selection approach entitled Dynamic Selection based on the Nearest Windows (DSNAW) that selects one or more competent models according to their performance in the region of competence composed of the nearest antecedent windows to the new target time window. This strategy assumes that the temporal windows closer to a test pattern have a behavior more similar to the target than in-sample data. The experimental study using ten well-known time series showed that the DSNAW outperforms the literature approaches.

INDEX TERMS Time series forecasting, multiple predictor systems, dynamic predictor selection, dynamic ensemble selection.

I. INTRODUCTION

Time series forecasting is a central task in many application areas, such as Economy [1], Seismology [2], Meteorology [3], and Astronomy [4], Hydrology [5], Engineering [6]. The development of accurate forecasting systems has been a central goal in the time series modeling area. However, due to the presence of different temporal patterns in real-world time series, accurate forecasting systems' construction is a challenge. In this sense, the classical approaches focus on seeking the best single model to forecast the whole time series have some serious drawbacks, such as the incorrect specification of the model's parameters [7]. Among the alter-

native approaches that have emerged, multiple predictor systems (MPS) have gained attention because they employ an ensemble of models instead of using only one model to improve the system's accuracy [8]. Such an ensemble contains many models that likely capture different behavior of the time series. MPS are composed of three phases [9]: Generation, Selection, and Integration or Combination. In the Generation phase, a pool of models is generated using a training set. After, one or more models are selected using some criterion in the Selection phase. In the last phase, the prediction is obtained by combining the forecasts of the selected models.

The Selection phase can be performed during the system's training (offline) or in the generalization (online). In either part, online or offline, this phase has a crucial role because it is related to the accuracy [10] and the computational

The associate editor coordinating the review of this manuscript and approving it for publication was Justin Zhang¹.

complexity [11] of the MPS. Dynamic selection approaches aim to select a subset of the pool containing the most suitable models to predict each new test pattern [9], [12]–[14]. Thus, instead of using the same models for all new patterns, these approaches select a different subset of models per pattern, which is a suitable strategy in modeling real-world time series that exhibit dynamic behavior that changes over time [15]. However, selecting the most suitable subset of models for a given test pattern is complex since there is no well-established criterion [16].

Dynamic selection approaches commonly use the performance of the models in the Region of Competence (RoC) as a criterion to select the most competent ones [9], [12], [13], [17]–[19]. The RoC is composed of the k patterns in the in-sample (training or validation sets) [12], which are more similar to the test pattern according to some measure such as the Euclidean distance [20]. This strategy for populating the RoC is applied for different tasks such as classification [21], [22], regression [23], [24] and time series forecasting [9], [12]–[14], [18].

The accuracy of dynamic selection approaches is directly related to the region of competence's quality, commonly defined using the k -nearest neighbors algorithm (k -NN) to find the k most similar patterns to the test pattern. However, to guarantee the optimum similarity measure is being adopted for a given time series is a challenging task [25] because the parameters setting of the k -NN is problem-dependent [18], [26]. Moreover, it is unlikely to assure the existence of patterns having similar behavior to the new test patterns (time windows) in the RoC due to noise, the available amount of data, or the absence of similar data [27]. In the time series context, the data distribution can change over time [28], [29]; consequently, the most suitable model to forecast a new test pattern can also change [30].

In this paper, we claim that the temporal windows closer to the new test pattern tend to have more similar behavior to the target than the traditional approach that used the k -NN to define the RoC. To validate such a hypothesis, the Kolmogorov Smirnov (KS) [31] test is applied to compare the data distributions defined by the proposal against the traditional approach. Based on this claim, we propose a new dynamic selection approach, entitled Dynamic Selection based on the Nearest Windows (DSNAW), that defines the region of competence with the nearest windows to the new test sample. DSNAW consists of four phases: i) definition of region of competence using the nearest antecedent windows to the new time window; ii) evaluation of each model in the region of competence using a forecasting measure; iii) construction of a ranking of the model by their performance; iv) forecasting of the point using the selected models.

The proposed approach is evaluated in the one-step ahead forecast using ten real-world time series. The proposal's performance is compared to traditional and state-of-the-art approaches in terms of seven well-known forecasting error measures: Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute

Percentage Error (SMAPE), Normalized Root Mean Square Error (NRMSE), Root Mean Square Error (RMSE), Average Relative Prediction Error Variation (ARV), and Mean Absolute Error (MAE) [6], [32]. The Dynamic Selection based on Nearest Antecedent Windows (DSNAW) is proposed based on the assumption that the temporal windows closer to the new test pattern are promising candidates to compose the region of competence. The main advantages of the proposed approach are:

- The creation of the region of competence is based on the assumption that the nearest temporal windows close to the new test pattern are promising candidates to compose the region of competence;
- DSNAW automatically finds the number (k) of time windows that compose the region of competence (RoC) per data set;
- DSNAW also automatically defines the number (n) of forecasting models used in the test phase. So, DSNAW can choose the best approach between Dynamic Predictor Selection or Dynamic Ensemble Selection for each data set;
- DSNAW selects the most suitable way to combine the forecasting models' outputs when more than one model ($n > 1$) is selected.

The remainder of the paper is organized as follows. Section II formulates the problem, and an experiment is performed to analyze the distribution of the region of competence generate by the patterns of the in-sample set. Section III introduces the proposed approach. Section IV describes the methodology that is used in the experiments. Section V presents a comparative study involving ten real-world time series and the respective discussion. Section VI shows the concluding remarks and future work.

II. PROBLEM DEFINITION

Dynamic selection approaches [9], [12], [13], [17] for time series forecasting have been proposed aiming to improve the accuracy of MPS. These approaches are based on [23] that seminally proposed a dynamic selection approach for regression tasks named herein to as Dynamic Selection by Local Accuracy (DS-LA). In the time series forecasting area, the dynamic selection approaches supposes that the most suitable model to forecast the future value z_{t+1} given the new time window $w_t = \{z_t, z_{t-1}, z_{t-2}, \dots, z_{t-n-1}\}$ with n lags (previous records), is the model with the best performance in the neighbourhood of w_t . This neighbourhood called the region of competence (RoC) is composed of k time windows from the training or validation set most similar to w_t .

The definition of the RoC plays an important role since its quality can be a limiting factor to the algorithm's performance. Likewise to classification problems, it is not possible to guarantee the existence of patterns really similar to the test sample in the region of competence since it depends on the distribution of the used dataset [27]. Besides, the most suitable similarity measure to define the region of competence for a given temporal behavior is unknown.

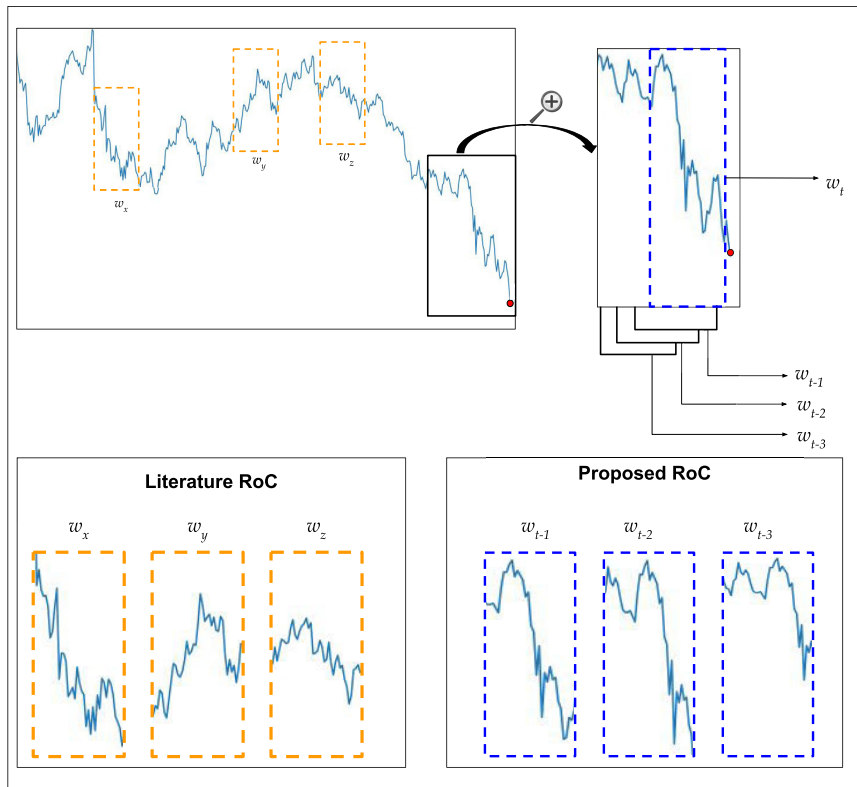


FIGURE 1. An example showing the region of competence defined by the Literature and by the proposed approach.

Figure 1 shows a new time window w_t and its RoC composed of three temporal windows w_x , w_y , and w_z . This RoC was defined using the assumption adopted by literature approaches, named “Literature RoC”. According to a given distance measure, the Literature RoC searches for the temporal patterns in the training and/or validation sets most similar to the new time window. We claim that this strategy is not appropriate for the time series forecasting since it may select a weak model to predict z_{t+1} . In contrast, we suppose that selecting the temporal windows just before w_t can produce better results; this new region of competence is composed of w_{t-1} , w_{t-2} , and w_{t-3} , as shown in Figure 1 - “Proposed RoC”

It is reasonable to consider that the best model to forecast z_{t+1} changes over time according to the data distribution. Based on this, we hypothesize that the k previous temporal windows (w_{t-1} , w_{t-2} , \dots , w_{t-k}) and w_t have a high probability of belonging to the same distribution. So, the best forecasting model for the k closest previous windows is a promising candidate to forecast z_{t+1} , since the behavior of a time series can be described through its probabilistic distribution function (PDF) over time [33]. PDF, estimated here as $F(w_t)$, is a function that describes the probability that an observation of the time series falls into a range of values. So, any change in $F(w_t)$ affects $F(z_{t+1}|w_t)$, since w_t and z_{t+1} are described using the same distribution. Hence, a model that attains a high performance for w_t is a promising candidate to forecast z_{t+1} . Moreover, this candidate changes depending on

the w_t , which is a desirable strategy since it is expected that the data distribution of real-world time series changes over time [28].

For evaluation of the proposed assumption, an experiment was performed to compare the data distribution of the new patterns with the Literature RoC and Proposed RoC. Ten time series¹ with different behaviors (described in Table 1) are used in this evaluation. Each time series was divided into three sets: the first 50% of the points for the training set, the next 25% for the validation set, and the last 25% for the test set. For this experiment, the training set is defined as in-sample, the validation set is used as out-of-sample, and the test is not considered.

For the experiment, each window (w_t) contains 20 records from the out-of-sample, i.e., 20 time lags that are used to forecast z_{t+1} . So, in a dataset with 100 records, 80 windows are generated. It is important to stress that between two neighboring windows (w_t and w_{t+1}), the difference is of only one record. For each w_t of the out-of-sample, two regions of competence Δ_t^{LT} and Δ_t^{PR} are generated. Each region of competence is composed of 10 windows ($k = 10$), where $\Delta_t^{LT} = (w_{m1}, w_{n2}, \dots, w_{ik})$ represent the windows from in-sample selected by the literature assumption [9], [12], [13], [17] (using Euclidean distance), and $\Delta_t^{PR} = (w_{t-1}, \dots, w_{t-k})$ are the k closest windows to w_t selected by the proposed assumption.

¹https://github.com/Eraysongaldino/dataset_time_series

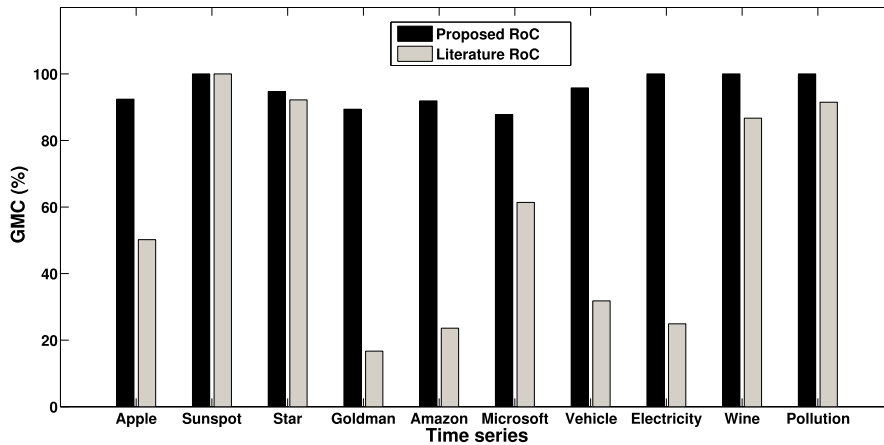


FIGURE 2. Percentual of selected windows (GMC) presents in the RoC that has the same distribution of the out-of-sample pattern w_t . The black and grey columns correspond to the “Proposed RoC” and “Literature RoC” that were generated by proposed and literature approaches, respectively.

The Kolmogorov Smirnov (KS) test [31] was chosen to compare the data distribution. The KS test is used to identify similar patterns on data and for detecting differences in position, dispersion, or shape of the distribution of the two samples [34]. This test is appropriate to analyze samples with sizes varying from 10 to 50 records [35]. The null hypothesis indicates that both samples have the same distribution, regardless of the distribution.

The KS test is applied to compare w_t with both regions of competence (Δ_t^{LT} and Δ_t^{PR}) separately. For example, given the test pattern w_t and the region of competence Δ_t^{LT} , w_t is compared with each pattern in Δ_t^{LT} and it is calculated how many times (out of k , which is the size of the region of competence) they have the same distribution. This formulation is defined in Equation 1.

$$GMC(w_t, \Delta_t) = \frac{100}{k} \sum_{w_j \in \Delta_t} KS(w_t, w_j), \quad (1)$$

where Δ_t can be Δ_t^{LT} or Δ_t^{PR} , k is the number of pattern in Δ_t , and the function $KS(w_t, w_j)$ outputs 1 if w_t and w_j have the same distribution, otherwise the output is 0.

Figure 2 shows the experimental results where the higher the Grand Mean of Occurrence in percent (GMC), the more selected windows have the same distribution as w_t . For most of the time series, the RoC defined by the proposed assumption obtained better results. In other words, the proposed hypothesis was able to select more windows with the same distribution as w_t in the in-sample set than the literature assumption. Besides selecting more promising windows than literature assumption, the proposal is computational cheaper because the search for the region of competence is straightforward.

III. THE PROPOSED APPROACH

Figure 3 shows the proposed architecture composed of three phases: generation, dynamic selection, and combination. The first phase generates a pool of forecasting models P that

is trained using the training dataset (ϕ). After, the dynamic selection phase chooses a relevant subset of models ($P' \subset P$) for each test pattern ($w_t \in \gamma$). If more than one predictor is selected, their outputs $\hat{z}_{t+1}^1, \dots, \hat{z}_{t+1}^n$ are combined in the last phase; otherwise, whether only one predictor is selected, the prediction of w_t is given by this predictor. These three phases are detailed in the next sections.

A. GENERATION

The generation phase’s objective is to create a diverse pool of forecasters that can model different time series behaviors. This diversity can be generated using two main strategies [36]: employing different samples for training each model or using different models trained from the same training sample.

Herein we decided to generate a homogeneous pool where all the predictors are trained using the same learning algorithm using Bagging [37]. Since Bagging performs a random sampling with replacement for populating each training data set, the diversity is achieved using different samples to train the models.

B. DYNAMIC SELECTION

The dynamic selection phase consists of selecting a single model or an ensemble of models per new time window (w_t). Algorithm 1 and Figure 4 show the proposed dynamic selection steps of the Nearest Antecedent Windows (DSNAW) algorithm.

The first step of the DSNAW is the definition of the region of competence Δ_t^{PW} that is composed of the k previous windows $\Delta_t^{PW} = (w_{t-1}, \dots, w_{t-k})$ closest to w_t (“Region of Competence Definition” module in Figure 4 and line 1 in Algorithm 1). This strategy of selecting the closest windows is based on the hypothesis presented in Section II that shows these windows represent a better choice than choosing the most similar windows, as performed by the DS-LA algorithm.

In the second step (“Models Selection” module in Figure 4 and lines 2 - 9 in Algorithm 1), all the models in are evaluated

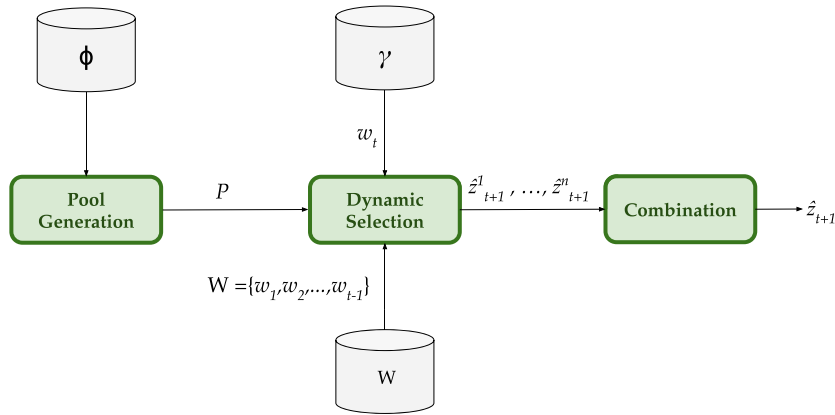


FIGURE 3. The flowchart of the proposed dynamic predictor selection system. P is the pool of predictors; w_t is the new time window in out-of-sample γ ; W is the all previous time window; $\hat{z}_{t+1}^1, \dots, \hat{z}_{t+1}^n$ are the forecasting of each model selected by the DSNAW; and, \hat{z}_{t+1} is the combined forecasting.

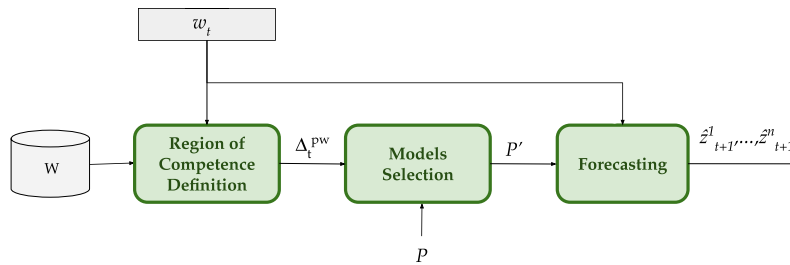


FIGURE 4. The flowchart of the dynamic selection phase performed by the proposed DSNAW algorithm. W is the all previous time window; w_t is the new time window; Δ_t^{PW} is the region of competence; P is the pool of predictors and P' is the set of models selected; $\hat{z}_{t+1}^1, \dots, \hat{z}_{t+1}^n$ are the forecasting of each model selected by the DSNAW.

in the region of competence. Thus, each model $p \in P$ is employed to predict the patterns in the region of competence Δ_t^{PW} . Each model’s accuracy is calculated using the Sum of Absolute Errors (SAE) [38] metric, which was chosen based on its robustness and reliability [39]. A ranking in ascending order of the models using the SAE is returned (line 8 – Algorithm 1). The lower the SAE value, the better the model’s accuracy.

The selection of the n best models, i.e., the n models having lower SAE (line 9 in Algorithm 1), can work in two different ways: dynamic predictor selection, in which only the best model is selected (in this case, $n = 1$), or dynamic ensemble selection, in which a set ($n > 1$) with the best models of the ranking is chosen.

In the last step (“Forecasting” module in Figure 4 and lines 10 - 16 in Algorithm 1), each one of the n selected models in P' is applied to predicted the value of w_t , generating the following forecasting $\hat{z}_{t+1}^1, \dots, \hat{z}_{t+1}^n$.

C. COMBINATION

In the last phase, the final forecasting \hat{z}_{t+1} is generated. Whether an ensemble ($n > 1$) was dynamically selected in the selection phase, the final prediction is given by the

combination of the output of all models in the ensemble. The combination can be performed using different strategies, such as average or median [8], [29]. If only one single model is selected ($n = 1$), the forecasting of this model is the final forecasting.

IV. EXPERIMENTAL PROTOCOL

An experimental evaluation of the proposed approach was conducted with a set of ten univariate time series (described in Table 1): Goldman Sachs, Sunspot, Star Brightness, Amazon, Apple, Microsoft, Vehicle, Wine, Pollution, and Electricity [40], [41].² These time series were chosen because they are widely used in the literature and have distinct behaviors regarding the presence or absence of seasonality, stationary, and trend [16]. Figure 5 shows the plot of each time series used in the experimental evaluation.

Each time series was normalized into the interval [0, 1]. After, the observations are organized into time sliding windows, composed of a maximum of 20 lags [42], where these lags are selected using the Auto-correlation function (ACF) [43]. Each time series was split into three sequential

²https://github.com/EraysnGaldino/dataset_time_series

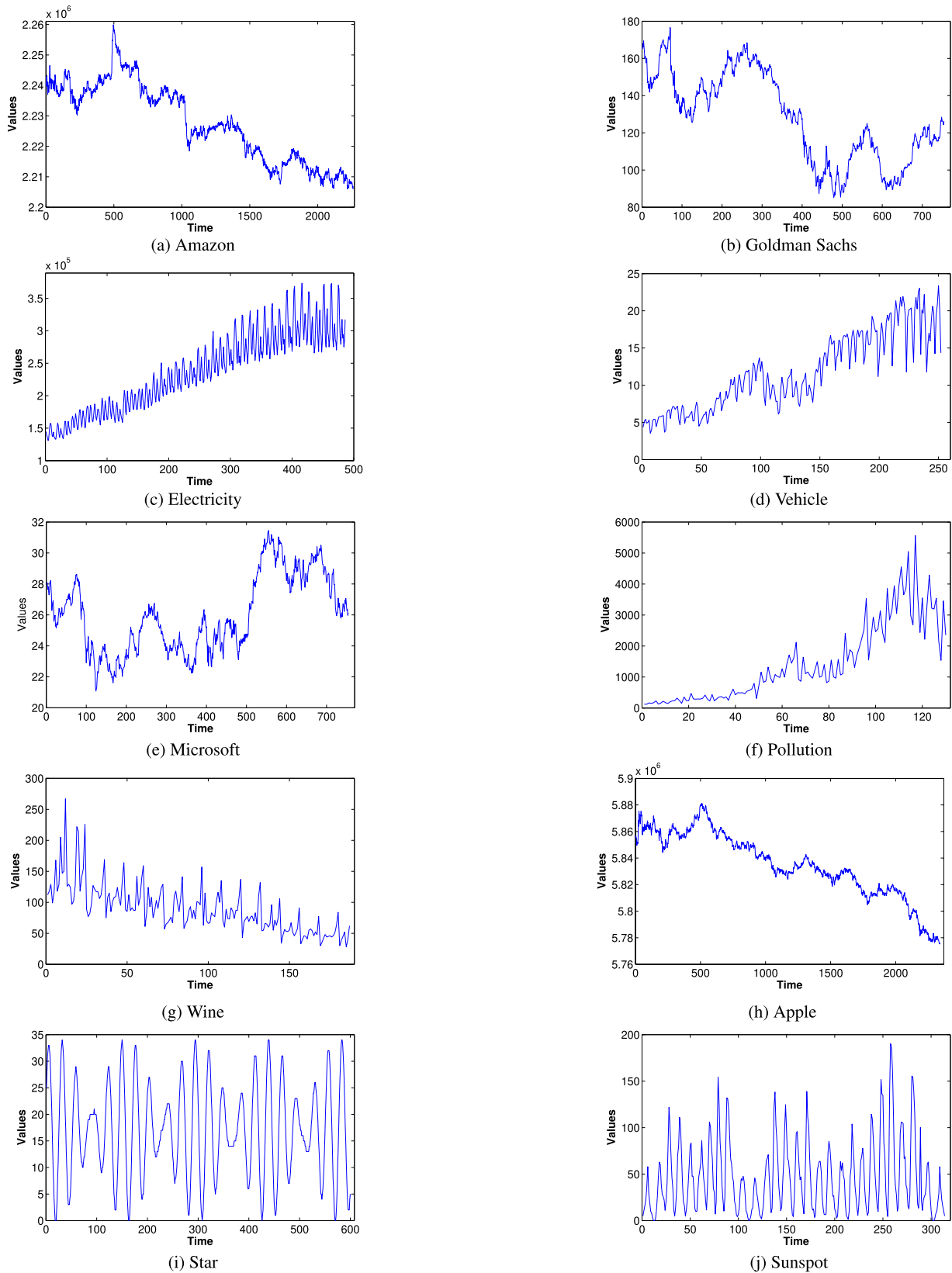


FIGURE 5. Time series used in the experimental evaluation.

samples with the following proportion: 50% for training, 25% for validation, and the last 25% observations for testing.

The bagging algorithm [37] that performs sampling with replacement was used to generate a diverse pool. Each bag

TABLE 1. Description of the data sets.

Time Series	Size	Description	Frequency	Period
Goldman S.	754	Adjusted close price of Goldman Sachs stock	Daily	01/04/2010-12/31/2012
Electricity	486	Australian electricity production	Monthly	01/1956-12/1996
Amazon	2261	Amazon ultra-high-frequency	Nanoseconds	-
Vehicle	252	Sales of vehicles in USA	Monthly	1971-1991
Microsoft	754	Adjusted close price of Microsoft stock	Daily	2010-2012
Pollution	130	Shipment of pollution equipment	Monthly	1986-1996
Wine	187	Australian wine sales	Monthly	1980-1994
Apple	2340	Apple ultra-high-frequency	Nanoseconds	-
Star	600	Brightness of a variable star on midnights	Daily	-
Sunspot	314	Wolfs Sunspot Numbers	Annual	1700-2014

Algorithm 1: Dynamic Selection From Nearest Antecedent Windows

```

// Input: All previous windows (W), new time
// window (wt), pool of predictors (P), size of
// the region of competence (k), number of
// models to be selected (n)
Input: W, P, k, n
Output:  $\hat{z}_{t+1}^1, \dots, \hat{z}_{t+1}^n$ 
// Region of competence definition composed of
// the k previous windows
1  $\Delta_t^{PW}$  = region-of-competence-definition(W, k)
// Evaluation of each model in the pool P
2 errors = []
3 for each predictor p ∈ P do
4    $\hat{y} = p(\Delta_t^{PW})$ 
5   e = error-measure(target,  $\hat{y}$ )
6   errors = errors ∪ {e}
7 end
// Selection of the n best forecasting models
8 ranking = ascending-order(errors)
9 P' = select-best-models(ranking, n)
// Applying the n selected models to the new
// time window wt
10 forecasting = []
11 for i = 1 to n do
12   p = P'(i)
13    $\hat{z}_{t+1}^i = p(w_t)$ 
14   forecasting = forecasting ∪ { $\hat{z}_{t+1}^i$ }
15 end
16 return forecasting

```

has the same size as the training sample, and it was split into 67% for training the model and 33% to validate the model.

The forecasting model pool is composed of 100 Support Vector Regressors (SVRs) based in [9]. SVR is a stable and robust model [44], which has attained significant performance accuracy in the forecasting task [40], [45]. A grid-search approach was employed for selecting the best combination of the hyper-parameters for each model.

TABLE 2. Values of the parameters for SVR.

Parameters	Values
Kernel	Radial basis function and Sigmoid
Gamma	0.5, 1, 10, 20, ..., 100, 200, ..., 1000
Cost	0.1, 1, 100, 1000, 10000
Epsilon	1, 1E-1, 1E-2, 1E-3, 1E-4, 1E-5, 1E-6

TABLE 3. Parameters of the proposed approach selected in the validation sample.

Time Series	Parameters		
	k	n	Combination approach
Amazon	20	14	Median
Apple	7	11	Average
Electricity	2	16	Median
Goldman Sachs	10	11	Median
Microsoft	12	10	Average
Pollution	12	3	Median
Star	10	16	Median
Sunspot	9	5	Median
Vehicle	18	10	Median
Wine	8	11	Median

The DSNAW has two parameters: *k* (size of the region of competence) and *n* (number of selected models). For both *k* and *n* parameters, the values are defined in the range [1, 20]. These values for *k* and *n* expand the interval evaluated in literature [46], [47]. When *n* = 1, one forecaster model is selected (Dynamic Predictor Selection task) and when *n* > 1, an ensemble of models is selected (Dynamic Ensemble Selection task). For the latter case (*n* > 1), the average and median operators are employed in the combination phase. These combination approaches are commonly employed because it is fairly simple and obtains accurate forecasting [8], [29]. The parameters *k*, *n*, and the combination operator are defined and selected through the validation set for each time series (Table 3).

The DSNAW was compared against 12 literature approaches. These approaches can be grouped into four main categories, described in the next paragraphs.

Single Models group is composed of the traditional statistical models and Support Vector Regression (SVR) model [48]:

- ARIMA [49]: the ARIMA parameters were estimated for each data set using the auto ARIMA python library [50];
- Exponential Smoothing (ETS) [51]: the ETS parameters were estimated using the Statsmodel library of the Python [52];
- Random Walk Forecasting (RW) [53]: RW model was modeled using the ARIMA(0,1,0). Its parameters were estimated through the Statsmodel library of the Python [52];
- SVR: named herein of Monolithic. The hyperparameters were selected using a grid-search (values in Table 2).

Dynamic Predictor Selection class is comprised of the following approaches:

- DS-LA (1): dynamic selection approach that employs parameters of the literature [46], [47]. The model is selected based on its performance in the region of competence composed of the time windows from in-sample, which are more similar to the test pattern;
- Temporal-window Framework (TWF) [54]: approach that trains the models using specific partitions of the time series. Given a test pattern, TWF selects the model trained in the partition more similar (calculated via the dynamic time warping algorithm [55]) to this new pattern. The TWF parameters were defined based on [54].

Full Pool group is formed of the approaches that combine all forecasting of the Pool generated by Bagging approach:

- $Bagg_A$ [37]: Bagging of SVRs combined by average;
- $Bagg_M$ [37]: Bagging of SVRs combined by median.

Dynamic Ensemble Selection class is composed of the approaches that selects an ensemble and combines them:

- DS-LA: dynamic selection approach [46], [47] employing parameters values selected in the validation sample for each data set;
- DES_A : Dynamic Ensemble Selection combined by mean using parameters defined in the literature [23];
- DES_M : Dynamic Ensemble Selection combined by median using parameters defined in the literature [23];
- DES-PALR [9]: DES - Predictor Accuracy over Local Region selects a set of forecasting models with the higher performance in the cluster (region of competence) that have the center closer to the new test pattern.

All models were assessed in the one step ahead forecasting using the test sample. The evaluation of the results was performed using seven traditional performance measures: Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Symmetric Mean Absolute Percentage Error (SMAPE), Normalized Root Mean Square Error (NRMSE), Root Mean Square Error (RMSE), Average Relative Prediction Error Variation (ARV), and Mean Absolute Error (MAE) [6], [32].

The following nomenclature is used in the definition of the performance metric (Equation 2 to 8): N is the number of observations, $output$ is the predicted value by the model, $target$ is the actual value of the time series, and t corresponds to the respective time (t) at the sample. The average, maximum and minimum values of the sample are represented by \overline{target} , $target_{max}$, and $target_{min}$, respectively.

MSE is a measure commonly used in the literature [32] to evaluate forecasting models. MSE is defined by the following equation:

$$MSE = \frac{1}{N} \sum_{t=1}^N (target_t - output_t)^2. \quad (2)$$

RMSE is the square root of the MSE value. The RMSE results are in the time series interval and hence can be more interpretable than MSE. The RMSE is defined as:

$$RMSE = \sqrt{MSE}. \quad (3)$$

The normalized RMSE (NRMSE) is used to compare the performance of models in the time series forecasting with different range values. The NRMSE is defined by the following equation:

$$NRMSE = \frac{RMSE}{((target_{max}) - (target_{min}))}. \quad (4)$$

MAPE computes the average forecast error percentage regardless of the scale of the values and indicates the forecast error margin in percentage. The following equation defines the MAPE:

$$MAPE = \frac{100}{N} \sum_{t=1}^N \left| \frac{target_t - output_t}{target_t} \right|. \quad (5)$$

The SMAPE measure evaluates the percentage of average absolute error independent of the scale of the values. The following equation defines the SMAPE:

$$SMAPE = \frac{100}{N} \sum_{t=1}^n \frac{|target_t - output_t|}{(|target_t| + |output_t|)/2}. \quad (6)$$

ARV is a relative metric that compares the model's forecast with the forecasting of time series using the mean. If the ARV value is less than 1, the model's accuracy is better than the use of mean; otherwise, the model is worse than the forecasting by the mean. If the ARV value is equal to 1, the forecasting model is equivalent to the mean. The ARV is defined by the following equation:

$$ARV = \frac{\sum_{t=1}^N (target_t - output_t)^2}{\sum_{t=1}^N (\overline{target} - output_t)^2}. \quad (7)$$

MAE is commonly applied to measure the error of the model on average. The following equation defines the MAE:

$$MAE = \frac{1}{N} \sum_{t=1}^N |target_t - output_t|. \quad (8)$$

The Diebold-Mariano (DM) [56] statistical test was applied to verify whether the proposed approach has a performance statistically equal ($pValue > 0.05$) to the literature approaches. If the performances are statistically different ($pValue \leq 0.05$), the best model is the one with the lowest MSE value.

Equation 9 was used to measure the performance percentage difference between the proposed approach and the literature approaches. ε_a is the performance of the base method, and ε_b is the performance of the proposed approach. The proposed approach is better than the base method when the ratio value is positive, which means the error of the base method (ε_a) is higher than the error of the proposed approach (ε_b). This measure is interesting because it allows quantifying the difference of performance between two approaches [16].

$$ratio = \frac{(\varepsilon_a - \varepsilon_b)}{\varepsilon_a} \times 100, \quad (9)$$

The source-code was implemented in the Python programming language using the Sklearn [57] library and performed on a computer with an Intel Core i7-7500 CPU and 20 GB RAM.

V. EXPERIMENTAL RESULTS

Table 4 shows the results of the proposal (DSNAW) and literature approaches using seven well-known performance metrics: MSE, MAPE, ARV, MAE, RMSE, NRMSE, and SMAPE. The DSNAW achieved better values in most performance metrics than single models in 6 out of 10 data sets. For Apple, Pollution, Sunspot, Wine series, the proposed method attained lower performance than at least one single model. In the Apple series, RW obtained better performance than DSNAW. For other data sets, RW, monolithic, and ARIMA models attained better measure values for Pollution, Sunspot, and Wine, respectively.

Compared with Dynamic Predictor Selection approaches (DS-LA (1) and TWF), DSNAW reached the best overall results in 9 out of 10 series. DS-LA (1) attained better error metrics values only for the Star series. TWF attained better values only for specific metrics in some data sets, such as ARV in Apple, Pollution, and Vehicle.

Regarding Ensemble approaches, the proposed method reached smaller measure values in 9 out of 10 data sets. DSNAW obtained a worse overall performance than two ensemble approaches, $Bagg_A$ and $Bagg_M$, for the Star series.

Concerning the four Dynamic Ensemble Selection approaches employed in the comparison, the DSNAW attained the best general performance in 9 out of 10 time series. The DS-LA, DES_A , DES_M , and DES-PALR approaches reached the best accuracy in terms of the set of measures for the Star series.

Table 5 shows the percentage difference in performance between the DSNAW and other approaches in terms of MSE using Equation 9. The proposed approach attained a positive gain compared with the other approaches in most time

series evaluated. The proposed approach reached a highlighted performance in several cases. For instance, DSNAW attained a percentage gain greater than 80% regarding most of the literature approaches for the Amazon, Apple, Electricity, Goldman, Microsoft, and Wine series. For the Pollution and Vehicle sets, the proposed approach reached a gain greater than 50% regarding most literature approaches. In the Star and Sunspot series, poor performance was reached, where the DSNAW loss was up to 28%. We can associate the performance percentage difference of the proposed approach to selecting windows with the same distribution in the region of competence. In most time series, where the literature dynamic selection approaches have the GMC value smaller than 70% (see Figure 2), the DSNAW has a performance 50% greater than other methods. Based on these results, we can correlate the high performance of the proposed approach with generating a region of competence composed of the high occurrence of windows with the same behavior.

Table 6 shows the statistical comparison between DSNAW and literature approaches using the DM hypothesis test. The symbols “+”, “-” and “~” mean that DSNAW attained better, worse, and equal accuracy than the concurrency, respectively. The last three rows show the summary of the comparison, where *Win* represents the number of time series the proposed approach obtained statistically significant results (less than 0.05), *Loss* represents the number of times the proposed approach obtained worse results than the literature approach, and *Tie* shows in how many time series the performance of proposed approach and the literature approach was similar. The proposed approach attained a statistically better or equal MSE value than literature approaches in 8 out of 10 data sets. For the Apple series, the DSNAW obtained a lower performance only regarding the RW model. For the Star series, $Bagg_A$, $Bagg_M$, DES_A , and DES_M approaches reached a better MSE value than the proposed approach. So, considering the 120 direct (10 data sets x 12 approaches) comparisons performed with literature approaches, DSNAW obtains 91 wins, 24 draws, and 4 losses. This result shows a significant performance attained by the proposed approach.

Figures 6 and 7 show the forecast of the DSNAW and the two best literature approaches for the Star and Amazon series, respectively. Although there are differences in the performance metrics for the Star series, Figure 6 shows that the forecasts of the selected models are very close. Indeed, Table 4 shows that the forecasting approaches based on ML models attain similar results. On the other hand, Figure 7 shows a significant difference in the forecast of the models for the Amazon series. It is possible to verify that the approaches generate distinct forecasts for the test set of the Amazon series.

A. DISCUSSION

In the previous section, DSNAW presented promising results in terms of accuracy regarding several literature approaches.

TABLE 5. Percentage difference (%) between DSNAW and the literature approaches using Equation 9 in terms of MSE.

Time Series	ARIMA	ETS	RW	Monolithic	DS-LA (1)	TWF	<i>Bagg_A</i>	<i>Bagg_M</i>	DS-LA	<i>DES_A</i>	<i>DES_M</i>	DES-PALR
Amazon	97.87	97.85	98.92	99.61	98.60	21.97	94.50	88.01	98.20	98.59	98.20	99.40
Apple	30.58	30.41	-66.35	86.19	88.85	72.42	84.92	87.25	88.29	87.60	88.46	88.89
Electricity	82.40	75.19	86.14	98.03	92.14	38.68	81.75	76.07	89.96	87.97	89.65	94.32
Goldman	98.58	97.84	98.56	99.06	88.87	34.61	42.89	25.68	82.83	81.59	83.74	88.67
Microsoft	96.69	96.69	97.51	91.35	87.26	84.75	62.13	67.14	87.26	82.88	85.43	88.86
Pollution	41.91	6.26	-4.00	64.56	55.15	61.54	2.06	58.85	58.28	56.73	57.71	61.06
Star	99.96	99.92	99.92	21.05	-8.47	4.44	-28.55	-28.81	-27.96	-28.55	-28.81	1.04
Sunspot	47.22	57.20	57.30	-24.69	19.66	-5.03	-1.65	8.30	5.79	-1.65	8.30	16.49
Vehicle	38.45	24.21	55.23	93.67	79.91	71.48	68.06	68.23	67.83	64.07	67.40	51.39
Wine	-9.52	6.71	46.99	36.02	83.18	62.65	88.03	92.72	92.25	88.60	92.23	92.62

TABLE 6. DM statistical test (critical value $\alpha = 0.05$) results comparing the DSNAW performance with literature approaches regarding to MSE values. The symbols "+", "-" and "~" correspond to the better, worse and equal accuracy attained by the DSNAW, respectively.

Time Series	ARIMA	ETS	RW	Monolithic	DS-LA (1)	TWF	<i>Bagg_A</i>	<i>Bagg_M</i>	DS-LA	<i>DES_A</i>	<i>DES_M</i>	DES-PALR
Amazon	+	+	+	+	+	+	+	+	+	+	+	+
Apple	+	+	-	+	+	+	+	+	+	+	+	+
Electricity	+	+	+	+	+	+	+	+	+	+	+	+
Goldman	+	+	+	+	+	+	+	+	+	+	+	+
Microsoft	+	+	+	+	+	+	+	+	+	+	+	+
Pollution	~	~	~	+	+	+	~	+	+	+	+	+
Star	+	+	+	~	~	~	-	-	~	-	-	~
Sunspot	+	+	+	~	~	~	~	~	~	~	~	~
Vehicle	~	~	+	+	+	+	+	+	+	+	+	+
Wine	~	~	~	~	+	+	+	+	+	+	+	+
Wins	7	7	7	7	8	8	7	8	8	8	8	8
Tie	3	3	2	3	2	2	2	1	2	1	1	2
Loss	0	0	1	0	0	0	1	1	0	1	1	0

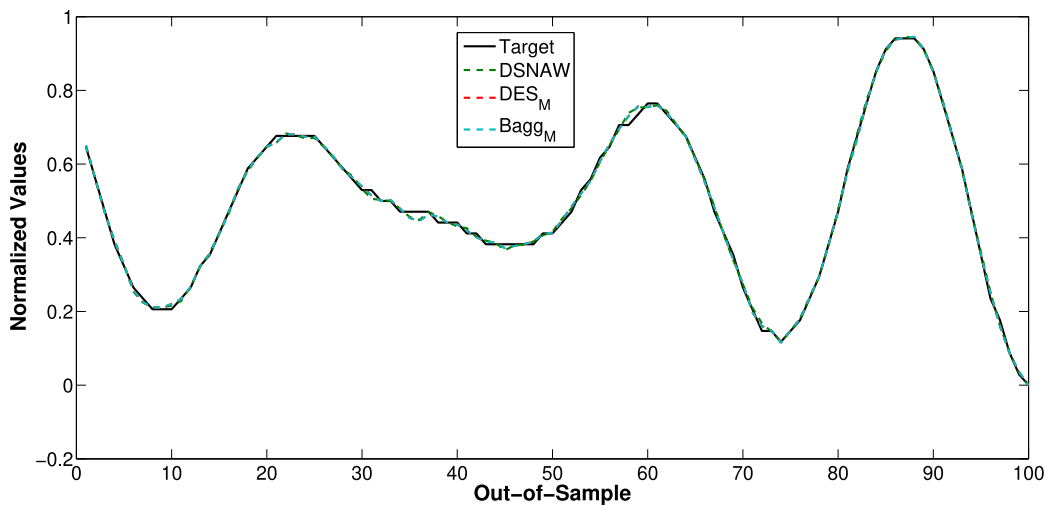


FIGURE 6. Star time series prediction values for the proposed approach and the best literature approaches in the first 100 points of the test set.

From the analysis of the accuracy of the MPS, it is possible to verify that DPS and DES approaches present higher variability in the accuracy than Ensembles. As all MPS used in this work employed the same pool, it can be inferred that this variability is related to how the forecasting models are chosen. This selection is performed from RoC, and its definition is closely related to system accuracy, as highlighted in the literature [9], [12], [13], [17].

The literature approaches create the RoC using temporal windows of the training and/or validation data according to some similarity criterion. However, there is no guarantee the existence of patterns really similar to the new test patterns since the data can be noisy, or the generator phenomenon of the data can have changed over time. On the other hand, DSNAW creates the RoC using the available temporal records closer to the test pattern. The objective is to increase the

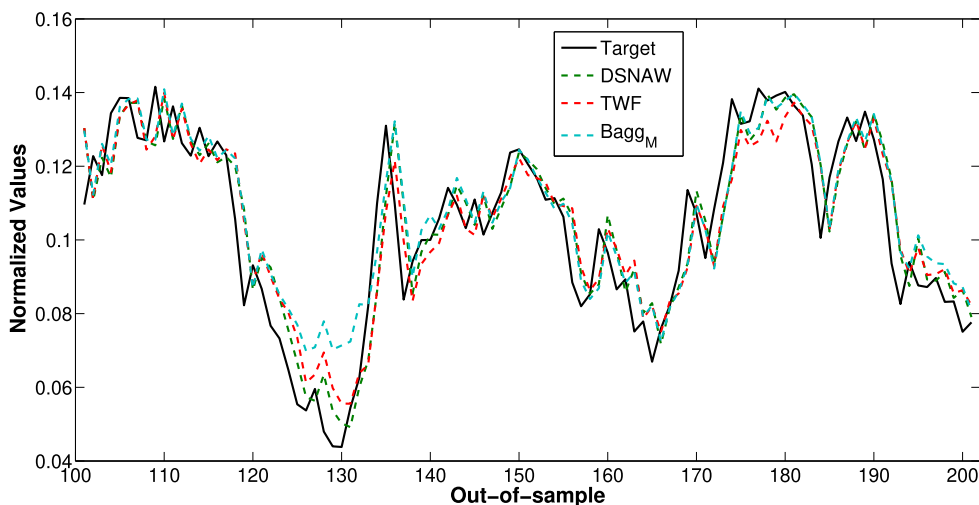


FIGURE 7. Amazon time series prediction values for the proposed approach and the two best literature approaches interval 100 to 200 points in the test set.

chance of using more similar temporal windows to the test pattern. So, an accurate forecasting model in the defined RoC is a promising candidate to forecast the test pattern. This strategy can overcome the issues mentioned earlier once the RoC is generated from the latest temporal data.

Moreover, the DSNAW an adaptable proposal since it defines which dynamic selection approach is the most promising between DPS and DES from their performance in the validation set.

VI. CONCLUSION

This paper presented a novel dynamic selection algorithm for time series forecasting called Dynamic selection based on the nearest windows, DSNAW for short. DSNAW is a multiple predictor system that selects the best models per query instance instead of combining all the models.

One key point in dynamic selection algorithms is the definition of the region of competence, a set composed of time windows that belong to the validation or training set. This region of competence is employed to select the best models per test pattern. A common alternative to select the time windows for the region of competence is to search for the patterns that have a minimum distance to the test pattern, as used in [23]. DSNAW is based on the assumption that the time windows close to the test pattern are more likely to have similar behavior to the test pattern than the previous strategy. Thus, a model that performs well in the region of competence formed by the closest windows to the test pattern is a promising candidate to forecast it.

DSNAW employs this new way to define the region of competence jointly by choosing the most suitable approach between Dynamic Predictor Selection or Dynamic Ensemble Selection for each data set under study. The experimental evaluation was carried out using ten real-world time series. The proposed approach reached better performance values than literature models (single models, ensembles, and dynamic selection approaches) in most time series evaluated.

These promising results obtained by DSNAW show that it selected more competent models per test pattern than the concurrence since it better defines the composition of the region of competence.

As future work, given that choosing between the two contexts of the DSNAW (DPS or DES) is time-dependent, we intend to propose a meta-test that aims to select one context or another depending on the test pattern under analysis.

REFERENCES

- [1] S. C. Nayak, B. B. Misra, and H. S. Behera, "On developing and performance evaluation of adaptive second order neural network with ga-based training (ASONN-GA) for financial time series prediction," in *Advancements in Applied Metaheuristic Computing*. Hershey, PA, USA: IGI Global, 2018, pp. 231–263.
- [2] M. Y. Khan and S. Mittnik, "Nonlinear time series modeling and forecasting the seismic data of the Hindu Kush region," *J. Seismol.*, vol. 22, no. 1, pp. 353–376, 2017.
- [3] G. Sun, C. Jiang, P. Cheng, Y. Liu, X. Wang, Y. Fu, and Y. He, "Short-term wind power forecasts by a synthetical similar time series data mining method," *Renew. Energy*, vol. 115, pp. 575–584, Jan. 2018.
- [4] S. L. D. Junior, J. R. Cecatto, M. M. Fernandes, and M. X. Ribeiro, "An optimized data mining method to support solar flare forecast," in *Information Technology-New Generations*. Cham, Switzerland: Springer, 2018, pp. 467–474.
- [5] G. Liu, F. Xiao, C.-T. Lin, and Z. Cao, "A fuzzy interval time-series energy and financial forecasting model using network-based multiple time-frequency spaces and the induced-ordered weighted averaging aggregation operation," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2677–2690, Nov. 2020.
- [6] S. Mao and F. Xiao, "A novel method for forecasting construction cost index based on complex network," *Phys. A, Stat. Mech. Appl.*, vol. 527, Aug. 2019, Art. no. 121306.
- [7] I. Ginzburg and D. Horn, "Combined neural networks for time series analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 1994, p. 224.
- [8] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," *Expert Syst. Appl.*, vol. 41, no. 9, pp. 4235–4244, Jul. 2014.
- [9] C. Yao, Q. Dai, and G. Song, "Several novel dynamic ensemble selection algorithms for time series prediction," *Neural Process. Lett.*, vol. 50, no. 2, pp. 1789–1829, 2018.
- [10] R. Liang and P. Jia, "Coal mine gas concentration prediction model based on selective ensemble learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 3789–3793.

- [11] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using Bayesian model averaging to calibrate forecast ensembles," *Monthly Weather Rev.*, vol. 133, no. 5, pp. 1155–1174, 2005.
- [12] D. Yankov, D. DeCoste, and E. Keogh, "Ensembles of nearest neighbor forecasts," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, 2006, pp. 545–556.
- [13] A. Widodo and I. Budi, "Model selection for time series forecasting using similarity measure," in *Proc. Int. Conf. Adv. Comput. Sci. Inf. Syst.*, Dec. 2011, pp. 221–226.
- [14] I. Budi, R. F. Aji, and A. Widodo, "Prediction of research topics using ensemble of best predictors from similar dataset," *Int. J. Comput., Electr., Autom., Control Inf. Eng.*, vol. 8, no. 1, pp. 86–92, 2014.
- [15] L. Cao and Q. Gu, "Dynamic support vector machines for non-stationary time series forecasting," *Intell. Data Anal.*, vol. 6, no. 1, pp. 67–83, Apr. 2002.
- [16] L. Wang, Z. Wang, H. Qu, and S. Liu, "Optimal forecast combination based on neural networks for time series forecasting," *Appl. Soft Comput.*, vol. 66, pp. 1–17, May 2018.
- [17] F. Nadeem, D. Alghazzawi, A. Mashat, K. Faqeeh, and A. Almalaise, "Using machine learning ensemble methods to predict execution time of e-science workflows in heterogeneous distributed systems," *IEEE Access*, vol. 7, pp. 25138–25149, 2019.
- [18] J. F. L. de Oliveira, E. G. Silva, and P. S. G. de Mattos Neto, "A hybrid system based on dynamic selection for time series forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 29, 2021, doi: 10.1109/TNNLS.2021.3051384.
- [19] G. Zhu and Q. Dai, "Ens^{KDE} & IncL^{KDE}: A hybrid time series prediction algorithm integrating dynamic ensemble pruning, incremental learning, and kernel density estimation," *Appl. Intell.*, vol. 51, no. 2, pp. 617–645, 2021.
- [20] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining Knowl. Discovery*, vol. 26, no. 2, pp. 275–309, 2013.
- [21] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 405–410, Apr. 1997.
- [22] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Inf. Fusion*, vol. 41, pp. 195–216, Sep. 2018.
- [23] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," in *Proc. Int. Workshop Multiple Classifier Syst.* Berlin, Germany: Springer, 2004, pp. 164–173.
- [24] T. J. M. Moura, G. D. C. Cavalcanti, and L. S. Oliveira, "MINE: A framework for dynamic regressor selection," *Inf. Sci.*, vol. 543, pp. 157–179, Jan. 2021.
- [25] R. M. O. Cruz, R. Sabourin, G. D. C. Cavalcanti, and T. Ing Ren, "META-DES: A dynamic ensemble selection framework using meta-learning," *Pattern Recognit.*, vol. 48, no. 5, pp. 1925–1935, May 2015.
- [26] E. G. Silva, G. D. C. Cavalcanti, J. F. L. de Oliveira, and P. S. G. de Mattos Neto, "On the evaluation of dynamic selection parameters for time series forecasting," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–7.
- [27] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Prototype selection for dynamic classifier and ensemble selection," *Neural Comput. Appl.*, vol. 29, no. 2, pp. 447–457, 2018.
- [28] R. Elwell and R. Polikar, "Incremental learning of concept drift in non-stationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.
- [29] R. Adhikari, "A neural network based linear ensemble framework for time series forecasting," *Neurocomputing*, vol. 157, pp. 231–242, Jun. 2015.
- [30] N. Meade, "Evidence for the selection of forecasting methods," *J. Forecasting*, vol. 19, no. 6, pp. 515–535, 2000.
- [31] A. Hong-Zhi and C. Bing, "A Kolmogorov-Smirnov type statistic with application to test for nonlinearity in time series," *Int. Stat. Rev./Revue Internationale de Statistique*, vol. 59, pp. 287–307, Dec. 1991.
- [32] P. S. G. de Mattos Neto, F. Madeiro, T. A. E. Ferreira, and G. D. C. Cavalcanti, "Hybrid intelligent system for air quality forecasting using phase adjustment," *Eng. Appl. Artif. Intell.*, vol. 32, pp. 185–191, Jun. 2014.
- [33] A. M. Kowalski, M. T. Martin, A. Plastino, and G. Judge, "On extracting probability distribution information from time series," *Entropy*, vol. 14, no. 10, pp. 1829–1841, Sep. 2012.
- [34] L. Mora-López and J. Mora, "An adaptive algorithm for clustering cumulative probability distribution functions using the Kolmogorov–Smirnov two-sample test," *Expert Syst. Appl.*, vol. 42, no. 8, pp. 4016–4021, May 2015.
- [35] H. W. Lilliefors, "On the Kolmogorov-Smirnov test for the exponential distribution with mean unknown," *J. Amer. Stat. Assoc.*, vol. 64, no. 325, pp. 387–389, Mar. 1969.
- [36] M. M. Rahman, M. M. Islam, K. Murase, and X. Yao, "Layered ensemble architecture for time series forecasting," *IEEE Trans. Cybern.*, vol. 46, no. 1, pp. 270–283, Jan. 2016.
- [37] F. Moretti, S. Pizzuti, S. Panziera, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, Nov. 2015.
- [38] T. S. Hu, K. C. Lam, and S. T. Ng, "River flow time series prediction with a range-dependent neural network," *Hydrolog. Sci. J.*, vol. 46, no. 5, pp. 729–745, Oct. 2001.
- [39] J. Sun, Z. Shen, H. Li, and Y. Shen, "Clustering via local regression," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Berlin, Germany: Springer, 2008, pp. 456–471.
- [40] E. G. Silva, D. S. de O. Junior, G. D. C. Cavalcanti, and P. S. G. de Mattos Neto, "Improving the accuracy of intelligent forecasting models using the perturbation theory," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.
- [41] V. M. Landassuri-Moreno and J. A. Bullinaria, "Neural network ensembles for time series forecasting," in *Proc. 11th Annu. Conf. Genetic Evol. Comput. (GECCO)*, Jul. 2009, pp. 1235–1242.
- [42] G. Zhu and Q. Dai, "Ens^{KDE} & IncL^{KDE}: A hybrid time series prediction algorithm integrating dynamic ensemble pruning, incremental learning, and kernel density estimation," *Appl. Intell.*, pp. 1–29, 2020.
- [43] M. C. A. Neto, G. Tavares, V. M. O. Alves, G. D. C. Cavalcanti, and T. Ing Ren, "Improving financial time series prediction using exogenous series and neural networks committees," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2010, pp. 1–8.
- [44] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. Adv. Neural Inf. Process. Syst.*, pp. 155–161, 1997.
- [45] J. F. L. de Oliveira and T. B. Ludermir, "A hybrid evolutionary decomposition system for time series forecasting," *Neurocomputing*, vol. 180, pp. 27–34, Mar. 2016.
- [46] J. Mendes-Moreira, A. M. Jorge, C. Soares, and J. F. de Sousa, "Ensemble learning: A study on different variants of the dynamic selection approach," in *Proc. Int. Workshop Mach. Learn. Data Mining Pattern Recognit.* Berlin, Germany: Springer, 2009, pp. 191–205.
- [47] T. J. M. Moura, G. D. C. Cavalcanti, and L. S. Oliveira, "Evaluating competence measures for dynamic regressor selection," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.
- [48] A. Mariette and R. Khanna, *Support Vector Regression. Efficient Learning Machines*. Berkeley, CA, USA: Springer, 2015, pp. 67–80.
- [49] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.
- [50] T. G. Smith, "Pmdarima: ARIMA estimators for Python," *Retrieved From*, vol. 309, Aug. 2017. [Online]. Available: <https://alkaline-ml.com/pmdarima/1.1.1/>
- [51] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*. Melbourne, VIC, Australia: OTexts, 2018.
- [52] S. Seabold and J. Perktold, "Statsmodels: Econometric and statistical modeling with Python," in *Proc. 9th Python Sci. Conf.*, vol. 57, 2010, p. 61.
- [53] S. Khanderwal and D. Mohanty, "Stock price prediction using ARIMA model," *Int. J. Marketing Hum. Resource Res.*, vol. 2, no. 2, pp. 98–107, 2021.
- [54] P. S. G. de Mattos Neto, G. D. C. Cavalcanti, P. R. A. Firmino, E. G. Silva, and S. R. P. V. N. Filho, "A temporal-window framework for modelling and forecasting time series," *Knowl.-Based Syst.*, vol. 193, Apr. 2020, Art. no. 105476.
- [55] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-26, no. 1, pp. 43–49, Feb. 1978.
- [56] F. Diebold and R. Mariano, "Comparing predictive accuracy," *J. Bus. Econ. Statist.*, vol. 20, no. 1, pp. 134–144, Jan. 2002.
- [57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, and J. Vanderplas, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.



ERAYLSON G. SILVA (Member, IEEE) received the degree in computer from the University of Pernambuco, in 2014, and the M.Sc. degree in computational intelligence from the Federal University of Pernambuco, in 2017, where he is currently pursuing the Ph.D. degree in computational intelligence. His research interests include artificial neural networks, time series forecasting, and ensemble.



GEORGE D. C. CAVALCANTI (Senior Member, IEEE) received the Ph.D. degree in computer science from the Universidade Federal de Pernambuco (UFPE), Brazil, in 2005. He joined the Centro de Informática (CIn)—UFPE, as a Professor of computer science, in 2005. He has been an Associate Member of the Laboratory of Imaging, Vision and Artificial Intelligence (LIVIA) at the École de Technologie Supérieure (ETS), Montreal, Canada, since 2011. His current research interests include machine learning, computer vision, pattern recognition, multiple classifier systems, imbalanced learning, instance reduction, meta-learning, text mining, neural networks, handwriting recognition, document understanding, and image processing.

...



PAULO S. G. DE MATTOS NETO (Member, IEEE) received the Ph.D. degree in computer science from the Center of Informatics, Federal University of Pernambuco, in 2012. He is a Researcher at the Brazilian National Research Council (CNPq). He is currently a Professor at the Center of Informatics, Federal University of Pernambuco. His current research interests include time series modeling, artificial neural networks, and hybrid systems. He is a member of International Neural Network Society (INNS) and the Brazilian Computer Science Society.