On the evaluation of dynamic selection parameters for time series forecasting

Eraylson. G. Silva*, George D. C. Cavalcanti*, João Fausto L. de Oliveira† and Paulo S. G. de Mattos Neto*

*Centro de Informática , Universidade Federal de Pernambuco, Brazil

†Escola Politécnica de Pernambuco, Universidade de Pernambuco, Brazil

Email: *{egs, gdcc, psgmn}@cin.ufpe.br, †jflo@poli.br

***Abstract*—Dynamic predictor selection has been applied to time series context to improve the accuracy to forecast. A crucial step in dynamic selection methods if the definition of the region of competence, which is composed of the most similar patterns to a test pattern, because the predictor that attains the best performance in this region is selected to forecast this test pattern. The performance of dynamic selection methods depends on two main parameters, the size of the region of competence and the similarity measure (also called of distance measure). This work evaluates the influence of these parameters on six real-world time series to forecasting one step. In the experiments, Bagging is adopted to generate a pool of predictors, where the best predictor is selected per query pattern based on its performance on the region of competence. The results show that the choice of an appropriate distance measure, as well as the size of the region of competence, is mandatory to boost the performance of the prediction system. Moreover, the results reinforce the importance of using a dynamic selection approach to improve forecasting accuracy when compared to the monolithic models, also called of single models.**

I. Introduction

Time series forecasting has been applied in several areas, such as Energy [1], Economy [2], Financial Market [3], Traffic Control [4], among others [5], [6]. In recent years, several Machine Learning (ML) models have been used for this task, e.g., Multilayer Perceptron Neural Network (MLP) [7], Support Vector Regression (SVR) [8], Radial Basis Function Network (RBF) [9], and Long Short-Term Memory (LSTM) [10]. ML models have been used to time series forecasting because they are non-parametric, data-driven, and perform nonlinear modeling [11]. Each ML model has specific characteristics and, consequently, can present different performances in the forecasting of a given time series. Furthermore, according to the no-free-lunch theorem [12], no technique is better than others in all possible cases. Thus, the selection of the most suitable ML model to forecast a specific time series is a relevant and challenging task [13], [14].

Multiple Predictor System (MPS), also named Ensemble, has been developed to deal with uncertainties inherent to the choice of model [15] aiming to improve the accuracy [16] of the whole system. The idea is to combine the strengths of different techniques with the objective to obtain a more robust system. Several studies have used MPSs to the forecasting task [1], [12], [17], [18]. In time series context, an ensemble of forecasting models generates the final forecast of two manners: combining the forecasts of two or more models or choosing one model. An MPS generally is composed of three steps [19]: (I) Generation, (II) Selection, (III) Integration. In the first step, a pool of forecasting models (set of models) is generated. In the second step, a single model or a subset of models is selected. In the last step, the final forecast is obtained combining the chosen models in the previous step.

The selection is a crucial step because it can improve the accuracy of the system [17] and reduce the computational complexity [20]. However, the selection step is a complex task since there is no certainty that the more appropriate model is being chosen for a specific problem [21]. Several methods have been proposed to select the best forecast model [12], [22], [23], [24], [25]. Most of these methods employ static selection, using the same model for forecasting all out-samples. However, due to the dynamic behavior of the time series, the dynamic selection approach has attained better results [12], [26], [27].

Dynamic selection methods aim at selecting one or more models according to some criterion [17], based on the assumption that a unique forecaster is not enough to adequately model local patterns that may be present in the time series [28]. In this way, different forecasting models may be experts in local regions of the time series [12]. So, the critical issue in the dynamic selection approach is to define the region of competence more adequate to evaluate the performance of the models in a determined task [29]. Dynamic selection methods developed for time series forecasting were inspired by the Local Accuracy (OLA) [30], which is a Dynamic Classifier Selection method, and by the Dynamic Selection (DS) method [31], which was initially proposed to regression problems and posteriorly applied to time series forecasting [32], [33], [34]. DS consists of selecting a set of patterns (region of competence) in the training or validation sets [35], which are more similar to the test pattern. After, the model with the lowest forecasting error in this region of competence is selected to predict the test pattern.

Regarding time series classification task, an empirical comparison of different similarity measures was carried out using the nearest neighbor method with a fixed value of $k = 1$ in [36]. The influence of the DS method parameters, distance measure, and neighborhood size, was evaluated in the classification task [37]. However, to the best of the authors' knowledge, the impact of the parameters of the DS in the time series forecasting context was not investigated.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

In time series forecasting, methods that use DS have outperformed single best predictors and static combinations of predictors in several studies [34], [38], [11]. However, the influence of the DS parameters in the overall performance of the system is still an open question.

In this paper, it is investigated the influence of the two main DS parameters: the k that defines the size of the region of competence and the similarity measure (distance measure) that determines the shape of the decision space selecting the patterns based on the similarity between the test pattern and the in-sample patterns. The study is performed to evaluate the sensitivity of the DS algorithm to the parameters. Experiments were performed employing ten similarity measure, and two values for k, to evaluated the one step ahead task on six real-world time series using the Mean Square Error (MSE) metric. Experimental results showed that the accuracy of the DS algorithm increases with the correct choice of the parameters, attaining better results than a single model in all evaluated cases.

The remainder of this paper is structured as follows: Section II presents the proposed methodology to analyze the influence of DS parameters in time series forecasting. The experimental setup and results are described in Section III and the conclusions are presented in Section IV.

II. Proposed Methodology

The proposed methodology aims to evaluate the influence of two parameters, the similarity measure and the number of patterns in the region of competence, in the performance of the dynamic selection step of an MPS. Section II-A and II-B describe the proposed methodology and the evaluated similarity measures, respectively.

A. Multiple Predictor Systems

Multiple Predictor Systems (MPSs) are composed of a set of forecasting models. The main idea behind this approach is to generate diversity among the individual ensemble members and use it for the improvement of the accuracy of the whole forecasting system. The diversity can be generated through different approaches, such as manipulation of the training set, handling of the model parameters and employing different models. The former approach is often employed in homogeneous forecasting systems, where the same type of model is trained on subsets of the original data set. Model parameters can also be changed to produce different models, such as changing neural networks weights.

MPSs are composed of three steps: (I) Generation, (II) Selection, (III) Integration. The first step is responsible for generating a pool of forecasting models $M = \{M_1, M_2 \ldots, M_N\}$. The pool should contain accurate and diverse models. The second step is responsible for selecting (statically, or dynamically) a single model, or a subset of models. In the static approach, the selection is performed in the in-sample patterns through some chosen selection criterion, e.g., Meta-Learning [25] and Ranking [23]. After, the selected model(s) is applied to forecast all out-of-sample patterns. In the dynamic selection step, one or more models can be selected for each new pattern in out-of-sample. The main approach used in the dynamic selection of forecasters is the DS [31]. DS selects the set of time windows (in the training or validation sets) more similar to a new pattern of the test sample and, after that, the model with the lowest prediction error in the region of competence is used to forecast the test pattern. In the integration step, the selected model are fused to return the predicted value of the test pattern. Combination approaches such as mean or median are commonly applied to produce the final forecasting. If only one model is selected, no integration is required.

Figure 1 shows the architecture of a typical MPS with two phases: (I) Generation and (II) Dynamic Selection. In the first phase, given a training data set of a time series Z_t, the Bagging method is used to generate a pool with N models trained in different subsets (Tr). The subsets are composed of patterns selected randomly of the training data set, taking into account the temporal order of the series. So, for application of the Bagging approach in the time series context is necessary to organize the data into a pattern (input, target). The input is created from a time sliding window $(z_t, z_{t-1}, \ldots, z_{t-m-1})$ where m is the size of the window. The target is the future point that will be forecast (z_{t+1}), preserving the temporal ordering. For each Tr_i, an M_i model is trained, resulting in a pool with N models.

In the second phase, the MPS is applied in the test set. For each new time window (TS_q) an M_q model is selected to forecast the pattern z_{t+1}. The DS algorithm is employed to select the M_q model from the generated pool in phase (I). This process consists of choosing the M_i model with the higher performance in the region of competence (R_q). R_q is defined as a set of k patterns that belengs to the training set (VI), which are more similar to the pattern TS_q, taking into account a similarity measure.

B. Similarity Measures

Given two time windows $X = (x_1, x_2, \ldots, x_d)$ and $Y = (y_1, y_2, \ldots, y_d)$, different metrics can be used to measure the similarity between them.

DS algorithm uses a similarity measure to select the k time windows in the in-sample set with behavior more similar to the new pattern of the test set. The k windows compose the region of competence, which is used to evaluate each model of the pool. It is fundamental to create a region of competence with patterns similar to the target pattern that will be predicted. In this context, it is crucial to use a similarity measure able to select the time window really more similar.

In the time series context, several distance measures are applied to calculate the similarity. In this study, the experiments are carried out using ten measures that can be organized into distances metric (Euclidean, Manhattan, Cosine, Correlation, Chebyshev, Hellinger, and Gower) and algorithms of distance (DTW, ShapeDTW, and EDRS). For a better comprehension, we use the following terms: dist(X,Y) is the distance between

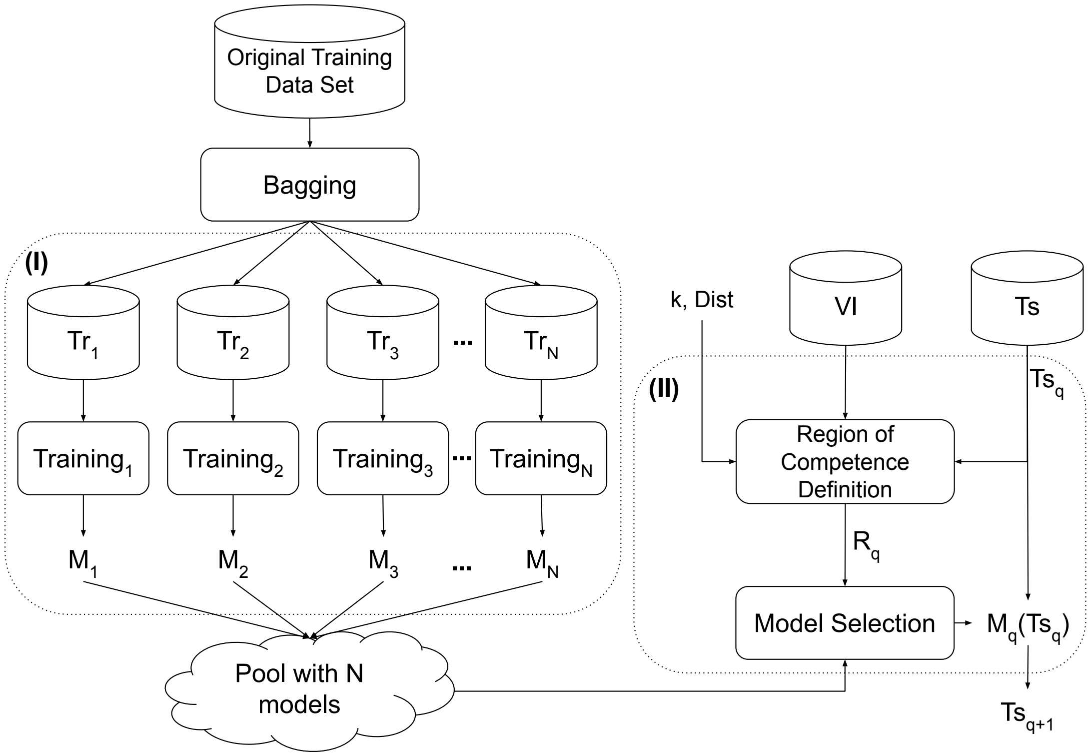

Fig. 1: Overview of the MPS proposed with the DS approach. It is divided into two steps: (I) Generation, where Tr_N subsets are generated through the Bagging approach, and M_N are predictor model of pool trained using each subset. (II) Dynamic Selection, where k and Dist are parameters of the DS, k to define the size of Region of Competence (R_q), and Dist to define the distance/similarity measure, IV is the training sample, (TS_q) is the pattern of out-of-sample (TS), and M_q is the model selected.

.

the windows X and Y, d is the window size, X' and Y' are the windows X and Y without the first value, respectively.

Euclidean. The most popular distance, with no configurable parameters and easy interpretation. The main drawbacks are sensibility to noise and unable to deal with local time shifts [39]. The Euclidean distance is defined as follows:

$$\mathrm{dist}(X, Y) = \sqrt{\sum_{i=1}^{d} (X_i - Y_i)^2} \quad (1)$$

Dynamic Time Warping (DTW) is considered the most successful similarity measure in time series classification context, but it has a high computational cost [40]. The dynamic programming is applied to determine the best alignment that will result in the DTW distance. DTW is defined as:

$$\mathrm{dist}(X, Y) = \sqrt{(X_1 - Y_1)^2} + \min(\mathrm{dist}(X', Y'), \mathrm{dist}(X', Y), \mathrm{dist}(X, Y')) \quad (2)$$

ShapeDTW is an algorithm that applies the DTW in similarly-shaped structures. The algorithm is composed of two steps: (I) transform each temporal point through a shape descriptor, resulting in a sequence of descriptors. (II) apply the DTW to align two sequences of descriptors and result in the distance of them [41].

Manhattan is also known as city block distance and has the simplicity in computation and it is more robust to the influence of outliers in comparison with other distance measures [42]. The Manhattan distance is computed as following:

$$\mathrm{dist}(X, Y) = \sum_{i=1}^{d} |x_i - y_i| \quad (3)$$

Cosine is applied to compute to the angular distance ignoring the scale of values of the time window [43]. The cosine distance following the equation:

$$\mathrm{dist}(X, Y) = 1 - \frac{\sum_{i=1}^{d} x_i \times y_i}{\sqrt{\sum_{i=1}^{d} x_i^2} \times \sqrt{\sum_{i=1}^{d} y_i^2}} \quad (4)$$

Correlation or Pearson distance is based Pearson's product-momentum correlation coefficient of two time windows [43]. The equation of Correlation distance is defined as:

$$\mathrm{dist}(X, Y) = 1 - \frac{\sum_{i=1}^{d} (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{d} (x_i - \overline{x})^2} \sqrt{\sum_{i=1}^{d} (y_i - \overline{y})^2}}. \quad (5)$$

Chebyshev computes the greatest difference along between each point in the time windows [44].The equation is defined as:

$$\mathrm{dist}(X, Y) = \max |x_i - y_i| \quad (6)$$

Hellinger compute the distance between two discrete probability distributions, measuring how to distribution are close to each other [45]. The equation of Hellinger distance is defined as:

$$\mathrm{dist}(X, Y) = 2\sqrt{1 - \sum_{i=1}^{d} \sqrt{x_i \times y_i}} \quad (7)$$

Gower calculate distance values in mixed data types [46]. The equation is defined as:

$$\mathrm{dist}(X, Y) = \frac{1}{d} \sum_{i=1}^{d} |x_i - y_i| \quad (8)$$

Edit Distance on Real Sequences (EDRS) is able to account misalignment between time windows, and reduce the effect of noise [47]. The EDRS is defined by:

$$\mathrm{dist}(X, Y) = \min\{\mathrm{dist}(X', Y') + \mathrm{cost}, \mathrm{dist}(X', Y) + 1, \mathrm{dist}(X, Y') + 1\}, \quad (9)$$

where the cost is equal to 0 when the points have the same values, 1 otherwise.

III. Simulation and Results

A. Experimental setup

An experimental study is conducted in the scenario of one step ahead forecasting. Six real-world time series are used: Goldman Sachs (Goldman), Star Brightness (Star) , Microsoft (MSFT), Vehicle, Red Wine, and Pollution[1]. Goldman Sachs series is composed of daily values of the adjusted close price of Goldman Sachs stock from 01/04/2010 to 12/31/2012, resulting in 754 points. Star Brightness series is composed of daily values of the brightness of a variable star at midnight, resulting in 600 points. Microsoft series corresponds to daily records of the adjusted close price of Microsoft stock from 01/04/2010 to 12/31/2012, totaling 754 points. Vehicle series is the collection of monthly values of sales of vehicles in the USA from 1971 to 1991, resulting in 252 points. Red Wine series is composed of monthly records of Australian wine sales from 1980 to 1994, totaling 187 points. These time series have different behaviors (seasonality, nonstationary, and trend) and are widely used in the literature. Each time series was normalized into the interval [0, 1] and divided as follows: the first 75% points of time series for training and validation, and the last 25% for testing.

The experimental simulation was performed using a pool with 100 Support Vector Regression (SVR) models. The pool was created with Bagging [4], as shown in Figure 1. SVR was chosen as the base model to composed the pool due to its good performance regarding accuracy in the forecasting task [48]. The pool was trained using a grid search approach to find the best configuration parameters per model (Table I).

[1]Available in https://github.com/EraylsonGaldino/dataset_time_series

The training set, which represents 75% of the time series, was resampled using Bagging. This process resulted in two sets: training (in-sample patterns) and validation (out-of-sample patterns) per model. These sets were organized into time sliding windows having a maximum size of 20 lags selected using the autocorrelation function (ACF) [49]. The training set (in-sample) was used to train each model, and the validation set (out-of-sample) was used to adjust the parameters of the model.

TABLE I: SVR parameters values.

Parameters	Values
Kernel	Radial basis function, Sigmoid
Gamma	$0.5,\ 1,\ 10,\ 20, \cdots, 100, 200, \cdots, 1000$
Cost	0.1,1,100,1000,10000
Epsilon	1, 0.1,0.001, 0.0001, 0.00001, 0.000001

The performance of the different configurations of the MPS was evaluated using the Mean Square Error (MSE) which is a widely used measure to evaluate forecasting models [50]. The MSE is defined in Equation 10.

$$\mathrm{MSE} = \frac{1}{N} \sum_{i=1}^{N} \left(z_i - \hat{z}_i\right)^2, \quad (10)$$

where z_i and $\hat{z}_i$ are the actual value of the time series and its forecast in time i, respectively.

Another way to assess the performance is to calculate the percentage difference between two forecasting approaches. Equation 11 shows the ratio (percentage difference).

$$\mathrm{ratio} = \frac{(\delta_A - \delta_B)}{\delta_A} \times 100, \quad (11)$$

where δ_A and δ_B are the performance in terms of MSE of the approaches A and B, respectively.

In this work, the performance of the proposed MPS is compared to the Oracle [29] and the Monolithic model. The Oracle [29] is a hypothetic approach that selects the best model for each point in the test sample. This approach is the best performance of a specific pool. The Monolithic model consists of only one SVR. Table I shows the parameters of the SVR defined using a grid search procedure.

B. Results

Table II shows the MSE values obtaining varying the parameter k and the distance measure. The best result achieved in each data set for a specific k value is underlined. The lowest MSE for each series is highlighted in bold. For the majority of the study cases, the MSE varies one order of magnitude for different distance measures employing the same k value. For the same distance measure, different error values also are attained for different k values.

For k = 1, the best distances were Euclidean and Shape DTW, two datasets for each one. When k = 10, Chebyshev

TABLE II: Performance (MSE) comparison varying the value of k and the similarity measure.

Distance	Star		MSFT		Red Wine		Pollution		Vehicle		Goldman	
	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10
DTW	1.01E-04	4.48E-05	2.41E-03	2.47E-03	4.56E-03	3.16E-03	5.36E-02	4.02E-02	3.45E-02	2.27E-02	6.37E-04	7.00E-04
Chebyshev	1.06E-04	4.93E-05	2.42E-03	2.41E-03	5.47E-03	**2.97E-03**	4.55E-02	**3.90E-02**	5.50E-02	2.53E-02	7.24E-04	6.29E-04
Manhattan	1.01E-04	4.48E-05	2.41E-03	2.47E-03	4.56E-03	3.16E-03	5.36E-02	4.02E-02	3.45E-02	2.27E-02	6.37E-04	7.00E-04
Correlation	1.09E-04	9.31E-05	1.48E-02	8.80E-03	2.92E-02	2.05E-02	5.03E-01	6.29E-02	8.17E-02	4.69E-02	2.30E-03	1.36E-03
Euclidean	1.07E-04	5.13E-05	2.11E-03	1.88E-03	3.32E-03	3.00E-03	5.36E-02	4.06E-02	3.58E-02	2.20E-02	6.47E-04	6.11E-04
Cosine	1.00E-04	5.21E-05	6.23E-03	3.08E-03	3.04E-02	2.05E-02	5.02E-01	4.96E-02	8.58E-02	3.23E-02	1.93E-03	1.78E-03
Gower	1.06E-04	**4.41E-05**	2.41E-03	2.47E-03	4.56E-03	3.16E-03	5.36E-02	4.02E-02	3.45E-02	2.27E-02	6.37E-04	7.00E-04
Hellinger	9.52E-05	6.12E-05	2.13E-03	**1.84E-03**	3.33E-03	3.12E-03	4.68E-02	4.06E-02	3.58E-02	2.21E-02	6.47E-04	7.20E-04
EDRS	1.04E-04	1.12E-04	1.84E-02	2.36E-02	4.01E-02	3.06E-03	9.87E-02	8.45E-02	1.51E-01	2.56E-02	3.55E-03	1.81E-03
Shape DTW	9.97E-05	8.24E-05	2.20E-03	2.29E-03	5.88E-03	3.01E-03	9.89E-02	5.22E-02	2.75E-02	**1.98E-02**	**6.06E-04**	9.48E-04

TABLE III: Comparing the best results in Table II against the results of the single model and the Oracle.

Distance	Star		MSFT		Red Wine		Pollution		Vehicle		Goldman	
	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10
DTW	1.01E-04	4.48E	2.41E-03	2.47E-03	4.56E-03	3.16E-03	5.36E-02	4.02E-02	3.45E-02	2.27E-02	6.37E-04	7.00E-04
Euclidean	1.07E-04	5.13E-05	2.11E-03	1.88E-03	3.32E-03	3.00E-03	5.36E-02	4.06E-02	3.58E-02	3.58E-02	6.47E-04	6.11E-04
Best Distance	9.52E-05	**4.41E-05**	2.11E-03	**1.84E-03**	3.32E-03	**2.97E-03**	4.55E-02	**3.90E-02**	2.75E-02	**1.98E-02**	**6.06E-04**	6.11E-04
Worst Distance	1.09E-04	1.12E-04	1.84E-02	2.36E-02	4.01E-02	2.05E-02	5.03E-01	8.45E-02	1.51E-01	4.69E-02	3.55E-03	1.81E-03
Monolithic	8.52E-05		2.30E-02		1.24E-02		1.26E-01		2.86E-01		1.26E-01	
Oracle	1.05E-06		2.29E-05		2.74E-05		6.45E-04		1.18E-03		9.39E-06	

distance attained the best accuracy in 2 out of 6 cases. From the similarity measures point of view, the best performances were obtained for Chebyshev and Shape DTW, each one won in two cases, while, Gower and Hellinger won in one case each. In 5 out of 6 data sets, the best MSE value was achieved employing a k value equal to 10. This result shows that the k value has a fundamental role in the performance of the DS algorithm.

Table III shows the comparison of the accuracy obtained by best configuration attained in Table II against the distances commonly used in the literature (Euclidean and DTW), the Monolithic approach, and the Oracle. The best configuration found in Table II attained higher performance than the Monolithic model in all data sets. The difference was one order of magnitude in 4 data sets (MSFT, Red Wine, Pollutions, Vehicle) and three orders of magnitude in one series (Goldman).

Comparing the Oracle with the best distance, the Oracle reached lower MSE by one order of magnitude in 2 series (Star, Vehicle), two orders of magnitude in 4 data sets (MSFT, Red Wine, Pollution, Goldman). Since Oracle results represent a hypothetical scenario (the best performance of the pool in the selection of a single model), this analysis shows us that there is plenty of room for improvement.

Table IV shows the percentage difference ratio (Equation 11) between the configurations of the MPS and single model for all data sets. A percentage equals to zero means that there was no difference between the single model and MPS. If the single model achieves lower error than the MPS, a negative ratio is achieved. Otherwise, a positive value is obtained if the opposite situation occurs. Furthermore, higher absolute ratio values represent a higher difference between the performances of the single model and of the MPS. For Goldman and Vehicle data sets, all configurations of the MPS attained the lower MSE than the single model. In these series, the highest percentage differences were reached. For other sets, the performance of the MPS was more sensitive in relation to the variation of the parameters distance measures and k values. For MSFT set, the choice of the distance measure and k value was critical for the performance of MPS. For example, the MPS using EDRS distance with k = 10 performed worse -2.64% concerning the single model, but for configuration Hellinger and k = 10 achieved a percentage gain of 91.99%. Table III also shows that for the same measure distance, different results can be reached with the variation of the k value. For Red Wine, MPS with EDRS distance employing k = 1 and k = 10 obtained -223.05% and 75.35%, respectively. For Pollution series, MPS with Correlation distance employing k = 1 and k = 10 obtained -298.92% and 50.07%, respectively. In general, the worst performances were achieved with k = 1.

Table III also shows that for the same k value, changing the distance measure generally leads to significant differences. For example, in the Pollution series, the MPS with k = 1 reached a percentage difference when compared with the single model of -298.92% and 63.87% with Correlation and Chebyshev distances, respectively.

C. Discussion

This work investigates the influence of the parameters of the DS algorithm for time series forecasting task. Ten different similarity measures were employed, varying the number of patterns within the region of competence (k) between two values (1 and 10), totaling twenty configurations. The results show that the combination of the parameters distance measure and k leads to different performances of the DS algorithm.

Among six time series evaluated, configurations of the DS employing the distance measures Chebyshev and Shape DTW achieved the best results into two data sets each one. So, 4 out of 10 distance measures reached the best results: Chebyshev for Red Wine and Pollution series, Shape DTW for Vehicle and Goldman series, Gower for Star set, and Hellinger for MSFT data set. From these results, two important points may be highlighted: the measure more used in the literature [34], [51], Euclidean distance, did not lead the DS to the best performance in any data base, and there is not a distance

TABLE IV: Percentage difference ratio between the proposal and the monolithic predictor.

Distance	Star		MSFT		Red Wine		Pollution		Vehicle		Goldman	
	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10	k = 1	k = 10
DTW	-18.34	47.47	89.52	89.28	63.24	74.56	57.47	68.07	87.94	92.07	99.49	99.44
Chebyshev	-23.95	42.17	89.48	89.51	55.91	76.03	63.87	69.02	80.78	91.15	99.43	99.50
Manhattan	-18.34	47.47	89.52	89.28	63.24	74.56	57.47	68.07	87.94	92.07	99.49	99.44
Correlation	-27.48	-9.26	35.57	61.74	-135.39	-65.51	-298.92	50.07	71.44	83.61	98.17	98.92
Euclidean	-25.89	39.81	90.82	91.84	73.22	75.81	57.47	67.80	87.48	92.31	99.49	99.51
Cosine	-17.89	38.87	72.93	86.62	-145.03	-65.51	-298.27	60.62	70.01	88.69	98.47	98.59
Gower	-24.42	48.28	89.52	89.28	63.24	74.56	57.47	68.07	87.94	92.07	99.49	99.44
Hellinger	-11.71	28.18	90.74	91.99	73.11	74.83	62.85	67.80	87.49	92.29	99.49	99.43
EDRS	-22.56	-31.43	19.98	-2.64	-223.05	75.35	21.67	32.96	47.12	91.06	97.18	98.56
Shape DTW	-17.04	3.27	90.41	90.03	52.54	75.70	21.50	58.58	90.39	93.07	99.52	99.25

measure most suitable for all cases. The first point shows that the investigation of other distance measures is a crucial research question to obtain accurate forecasts using the DS. The second issue is related to the difficulty to guarantee which a given temporal behavior be better identified using a specific distance measure. So, the definition of the best similarity measure for a particular time series is a complex task.

Table III shows that in 5 out of 6 series the best and worst accuracies can differ to one order of magnitude, changing only the k value. In 5 out of 6 data sets, the best accuracy was attained with k $= 10$. This result infers that through a region of competence composed of ten patterns (k $= 10$) is possible to select models more accurate. It can occur because of two reasons: First, the region of competence with more than one pattern similar to the new pattern of out-of-sample reduce the uncertainty to select the model suitable to predict the new pattern. Second, the best size of the region of competence can be influenced by the performance measure selected. In this work, the MSE was the performance measure applied to evaluate the models of the pool on patterns of the region of competence, this measure evaluates the model through than mean error, and for that is important applied in the region of competence with more than one pattern.

In Table IV, the results achieved show that the parameters could influence the MPS performance compared to the monolithic approach (an SVR model). In the Star series, the choice of k $= 1$ resulted in negative performance regardless of the distance measure. On the other hand, in the Goldman series the choice of parameters resulted in small variation performance. When using a k $= 10$, the DTW, Chebyshev, Manhattan, Euclidean, Gower, Hellinger and Shape DTW measures, achieve better performance than the monolithic in all time series evaluated. This result reinforces the importance of the use of MPS with dynamic selection compared to use of the monolithic model to time series forecasting.

IV. Conclusions

In this work, we investigated the influence of the similarity measure along with the number of patterns k in the region of competence on the performance of time series forecasting systems. In the proposed methodology, Bagging was employed to generate a homogeneous pool composed of SVRs. After, possible regions of competence are evaluated on different parameter configurations. In the end, the model with the best performance in the region of competence is selected to forecast the test pattern.

The experiments were performed on six publicly available data sets (Star, MSFT, Red Wine, Pollutions, Vehicle, and Goldman) considering ten distance metrics (DTW, Chebyshev, Manhattan, Correlation, Euclidean, Cosine, Gower, Hellinger, EDRS, and Shape DTW) and two values of k (1 and 10). The results showed that the selection of a proper distance metric and k value influence the results, and promising results can be achieved if the appropriate parameter values are selected. However, there is no best configuration of parameters for all the analyzed data sets; these parameters should be defined in a data set-dependent fashion. Moreover, the results reinforce the importance of the use of the dynamic selection approach to improve forecasting accuracy when compared to the monolithic model.

In future works, two research issues should be addressed: how to select the best similarity measure for a particular time series, and analyzing if there is some relation between the performance measure and the size of the region of decision. Furthermore, it is interesting to explore the tradeoffs of having a heterogeneous pool of predictors.

Acknowledgment

This work was partially supported by Brazilian agencies: CNPq and Facepe.

References

[1] J. Xiao, Y. Li, L. Xie, D. Liu, and J. Huang, "A hybrid model based on selective ensemble for energy consumption forecasting in china," *Energy*, vol. 159, pp. 534–546, 2018.

[2] 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," *Advancements in Applied Metaheuristic Computing*, pp. 231–263, 2018.

[3] S. Lahmiri, "Intraday stock price forecasting based on variational mode decomposition," *Journal of Computational Science*, vol. 12, pp. 23–27, 2016.

[4] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," *Neurocomputing*, vol. 167, pp. 3–7, 2015.

[5] J. P. González, A. M. San Roque, and E. A. Pérez, "Forecasting functional time series with a new hilbertian ARMAX model: Application to electricity price forecasting," *IEEE Transactions on Power Systems*, vol. 33, no. 1, pp. 545–556, 2018.

[6] C. Sudheer, R. Maheswaran, B. K. Panigrahi, and S. Mathur, "A hybrid SVM-PSO model for forecasting monthly streamflow," *Neural Computing and Applications*, vol. 24, no. 6, pp. 1381–1389, 2014.

[7] S. Samara and E. Natsheh, "Modeling the output power of heterogeneous photovoltaic panels based on artificial neural networks using low cost microcontrollers," *Heliyon*, vol. 4, no. 11, p. e00972, 2018.

[8] R. M. Ehsan, S. P. Simon, and P. Venkateswaran, "Day-ahead forecasting of solar photovoltaic output power using multilayer perceptron," *Neural Computing and Applications*, vol. 28, no. 12, pp. 3981–3992, 2017.

[9] J. de Jesus Rubio, I. Elias, D. R. Cruz, and J. Pacheco, "Uniform stable radial basis function neural network for the prediction in two mechatronic processes," *Neurocomputing*, vol. 227, pp. 122–130, 2017.

[10] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[11] R. Adhikari, "A neural network based linear ensemble framework for time series forecasting," *Neurocomputing*, vol. 157, pp. 231–242, 2015.

[12] C. Yao, Q. Dai, and G. Song, "Several novel dynamic ensemble selection algorithms for time series prediction," *Neural Processing Letters*, pp. 1–41, 2018.

[13] M. Ma, B. Jin, S. Luo, S. Guo, and H. Huang, "A novel dynamic integration approach for multiple load forecasts based on q-learning algorithm," *International Transactions on Electrical Energy Systems*, p. e12146, 2019.

[14] M. Hibon and T. Evgeniou, "To combine or not to combine: selecting among forecasts and their combinations," *International Journal of Forecasting*, vol. 21, no. 1, pp. 15–24, 2005.

[15] N. Kourentzes, D. K. Barrow, and S. F. Crone, "Neural network ensemble operators for time series forecasting," *Expert Systems with Applications*, vol. 41, no. 9, pp. 4235–4244, 2014.

[16] X. Qiu, Y. Ren, P. N. Suganthan, and G. A. Amaratunga, "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," *Applied Soft Computing*, vol. 54, pp. 246–255, 2017.

[17] R. Liang and P. Jia, "Coal mine gas concentration prediction model based on selective ensemble learning," in *2018 Chinese Control And Decision Conference (CCDC)*. IEEE, 2018, pp. 3789–3793.

[18] D. K. Barrow and S. F. Crone, "A comparison of adaboost algorithms for time series forecast combination," *International Journal of Forecasting*, vol. 32, no. 4, pp. 1103–1119, 2016.

[19] A. S. Britto Jr, R. Sabourin, and L. E. Oliveira, "Dynamic selection of classifiers—a comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014.

[20] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski, "Using bayesian model averaging to calibrate forecast ensembles," *Monthly weather review*, vol. 133, no. 5, pp. 1155–1174, 2005.

[21] C. Lemke and B. Gabrys, "Meta-learning for time series forecasting and forecast combination," *Neurocomputing*, vol. 73, no. 10-12, pp. 2006–2016, 2010.

[22] Z. Ma and Q. Dai, "Selected an stacking elms for time series prediction," *Neural Processing Letters*, vol. 44, no. 3, pp. 831–856, 2016.

[23] R. Adhikari, G. Verma, and I. Khandelwal, "A model ranking based selective ensemble approach for time series forecasting," *Procedia Computer Science*, vol. 48, pp. 14–21, 2015.

[24] A. Inoue and L. Kilian, "On the selection of forecasting models," *Journal of Econometrics*, vol. 130, no. 2, pp. 273–306, 2006.

[25] R. B. Prudêncio and T. B. Ludermir, "Meta-learning approaches to selecting time series models," *Neurocomputing*, vol. 61, pp. 121–137, 2004.

[26] A. V. Krikunov and S. V. Kovalchuk, "Dynamic selection of ensemble members in multi-model hydrometeorological ensemble forecasting," *Procedia Computer Science*, vol. 66, pp. 220–227, 2015.

[27] R. Adhikari and G. Verma, "Time series forecasting through a dynamic weighted ensemble approach," in *Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics*. Springer, 2016, pp. 455–465.

[28] P. S. de Mattos Neto, G. D. Cavalcanti, P. R. Firmino, E. G. Silva, and S. R. V. Nova Filho, "A temporal-window framework for modeling and forecasting time series," *Knowledge-Based Systems*, p. 105476, 2020.

[29] R. M. Cruz, R. Sabourin, G. D. Cavalcanti, and T. I. Ren, "Meta-des: A dynamic ensemble selection framework using meta-learning," *Pattern recognition*, vol. 48, no. 5, pp. 1925–1935, 2015.

[30] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 4, pp. 405–410, 1997.

[31] N. Rooney, D. Patterson, S. Anand, and A. Tsymbal, "Dynamic integration of regression models," in *International Workshop on Multiple Classifier Systems*. Springer, 2004, pp. 164–173.

[32] I. Budi, R. F. Aji, and A. Widodo, "Prediction of research topics using ensemble of best predictors from similar dataset," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 8, no. 1, 2014.

[33] D. Yankov, D. DeCoste, and E. Keogh, "Ensembles of nearest neighbor forecasts," in *European Conference on Machine Learning*. Springer, 2006, pp. 545–556.

[34] A. Widodo and I. Budi, "Model selection for time series forecasting using similarity measure," in *2011 International Conference on Advanced Computer Science and Information Systems*. IEEE, 2011, pp. 221–226.

[35] R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195–216, 2018.

[36] R. Giusti and G. E. Batista, "An empirical comparison of dissimilarity measures for time series classification," in *2013 Brazilian Conference on Intelligent Systems*. IEEE, 2013, pp. 82–88.

[37] L. Didaci and G. Giacinto, "Dynamic classifier selection by adaptive k-nearest-neighbourhood rule," in *International Workshop on Multiple Classifier Systems*. Springer, 2004, pp. 174–183.

[38] A. T. Sergio, T. P. de Lima, and T. B. Ludermir, "Dynamic selection of forecast combiners," *Neurocomputing*, vol. 218, pp. 37–50, 2016.

[39] 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 and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[40] S. Yoshida and B. Chakraborty, "A comparative study of similarity measures for time series classification," in *JSAI International Symposium on Artificial Intelligence*. Springer, 2015, pp. 397–408.

[41] J. Zhao and L. Itti, "Shapedtw: shape dynamic time warping," *Pattern Recognition*, vol. 74, pp. 171–184, 2018.

[42] S. Pratiher, S. Mukhopadhyay, R. Barman, S. Pratiher, A. Pradhan, N. Ghosh, and P. K. Panigrahi, "Covariance weighted distance metrics for optical diagnosis of cancer," in *2016 International Conference on Signal Processing and Communication (ICSC)*. IEEE, 2016, pp. 340–345.

[43] R. C. Cavalcante, L. L. Minku, and A. L. Oliveira, "Fedd: Feature extraction for explicit concept drift detection in time series," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 740–747.

[44] D. Shang, P. Shang, and L. Liu, "Multidimensional scaling method for complex time series feature classification based on generalized complexity-invariant distance," *Nonlinear Dynamics*, vol. 95, no. 4, pp. 2875–2892, 2019.

[45] M. D. Vamvakaris, A. A. Pantelous, and K. M. Zuev, "Time series analysis of s&p 500 index: A horizontal visibility graph approach," *Physica A: Statistical Mechanics and its Applications*, vol. 497, pp. 41–51, 2018.

[46] M. Voß, A. Haja, and S. Albayrak, "Adjusted feature-aware k-nearest neighbors: Utilizing local permutation-based error for short-term residential building load forecasting," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 2018, pp. 1–6.

[47] S. Mauceri, J. Sweeney, and J. McDermott, "Dissimilarity-based representations for one-class classification on time series," *Pattern Recognition*, vol. 100, p. 107122, 2020.

[48] E. G. Silva, S. d. O. Domingos, G. D. Cavalcanti, and P. S. de Mattos Neto, "Improving the accuracy of intelligent forecasting models using the perturbation theory," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–7.

[49] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[50] P. S. de Mattos Neto, F. Madeiro, T. A. Ferreira, and G. D. Cavalcanti, "Hybrid intelligent system for air quality forecasting using phase adjustment," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 185–191, 2014.

[51] 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. 25 138–25 149, 2019.