Hybrid systems using residual modeling for sea surface temperature forecasting

Paulo S. G. de Mattos Neto\textsuperscript{1,2}, George D. C. Cavalcanti\textsuperscript{1,2}, Domingos S. de O. Santos Júnior\textsuperscript{1,2} & Eraylson G. Silva\textsuperscript{1,2}

The sea surface temperature (SST) is an environmental indicator closely related to climate, weather, and atmospheric events worldwide. Its forecasting is essential for supporting the decision of governments and environmental organizations. Literature has shown that single machine learning (ML) models are generally more accurate than traditional statistical models for SST time series modeling. However, the parameters tuning of these ML models is a challenging task, mainly when complex phenomena, such as SST forecasting, are addressed. Issues related to misspecification, overfitting, or underfitting of the ML models can lead to underperforming forecasts. This work proposes using hybrid systems (HS) that combine (ML) models using residual forecasting as an alternative to enhance the performance of SST forecasting. In this context, two types of combinations are evaluated using two ML models: support vector regression (SVR) and long short-term memory (LSTM). The experimental evaluation was performed on three datasets from different regions of the Atlantic Ocean using three well-known measures: mean square error (MSE), mean absolute percentage error (MAPE), and mean absolute error (MAE). The best HS based on SVR improved the MSE value for each analyzed series by 82.26\%, 98.93\%, and 65.03\% compared to its respective single model. The HS employing the LSTM improved 92.15\%, 98.69\%, and 32.41\% concerning the single LSTM model. Compared to literature approaches, at least one version of HS attained higher accuracy than statistical and ML models in all study cases. In particular, the nonlinear combination of the ML models obtained the best performance among the proposed HS versions.

Climate change is one of humanity’s most critical global challenges since it is harmful to all living beings on Earth. Therefore, it is a crucial subject for all countries, independent of geographical, social, or economic characteristics. Global climate change directly affects the environment and has already had observable effects, such as glacier shrinkage, more intense heat waves, and sea-level rise. According to the Intergovernmental Panel on Climate Change, several regions will be affected in diverse manners over time, leading to significant societal and environmental systems changes. The Sea Surface Temperature (SST) is one of the most significant variables for monitoring the global climate system\textsuperscript{1}. SST is related to ocean heat content, which directly affects global warming. The SST records are collected using satellites, which read these values mainly of moored and drifting buoys and can be considered the best-known ocean parameter on global scales\textsuperscript{2}.

The variability of the SST is correlated with many natural phenomena\textsuperscript{1,3–6}. For instance, the El Niño/Southern Oscillation and Indian Ocean Dipole relate to the warming or cooling of SST in predefined areas of the Pacific and Indian oceans, respectively\textsuperscript{1}. Furthermore, the SST from the Atlantic ocean is connected with the quantity of rain and cloudiness in South America, droughts in Northeastern Brazil, and climate change on the Amazon vegetation\textsuperscript{6,7}. In this way, SST forecasting can support the decision in many operational applications, such as rainfall monitoring, turtles tracking, tourism, fishing management, and coral bleaching evaluation\textsuperscript{8}.

In the time series forecasting literature, statistical and machine learning (ML) models have been widely employed in various domains\textsuperscript{8–11}. Among statistical models, linear methods, such as Autoregressive (AR), Moving Average (MA), and ARIMA models are the most popular due to their simplicity, adaptability, and the Box and Jenkins methodology\textsuperscript{12}, which provides a well-established design process for time series modeling. The Box & Jenkins methodology used in the design of the linear models guarantees that the linear patterns are properly modeled. However, this class of models is not able to properly model temporal phenomena that present nonlinear
patterns. ML models have been highlighted due to their performance, flexibility, nonlinearity and because they are data-driven techniques, allowing temporal modeling without making any a priori assumption. Among the ML models, multilayer perceptron neural networks (MLP), support vector regression (SVR), and long short-term memory (LSTM) are examples of techniques that have reached promising results.

We can highlight the following works that used linear statistical models and nonlinear ML models for SST forecasting. Lins et al. employed an SVR to the daily forecast of one year ahead of two different locations of the tropical Atlantic ocean. Salles et al. applied the ARIMA model to analyze the temporal aggregation of seventeen datasets of SST located in the tropical Atlantic ocean. Tripathi et al. analyzed the MLP and linear regression techniques in the monthly forecasting of SST located in the Indian Ocean. Mahongo and Deo showed that the Nonlinear Autoregressive with Exogenous Input neural network reached an accurate performance in the SST forecast located in the African shoreline of the Indian ocean. Garcia-Gorriz and Garcia-Sanchez employed a system based on MLP to the monthly forecasting of the SST in the western Mediterranean Sea.

One of the primary objectives in time series analysis and forecasting is to develop accurate systems. Among the approaches that use error series methods, hybrid systems that combine models from the error series modeling have reached promising results in many applications. Residuals or error series are obtained from the difference between the time series and its forecasting. Such hybrid systems use the residuals modeling to correct biased forecasts that can occur due to the overfitting, underfitting, or misspecification of models. Hybrid systems commonly design the time series as a combination of a linear statistical model with a nonlinear ML model or as the combination of ML models. The former aims to model the linear and nonlinear patterns of the time series separately. The latter employs ML models for error series modeling intending to improve the accuracy of an initial nonlinear ML model.

To the best of our knowledge, hybrid systems that perform the residuals modeling were not proposed or evaluated for SST time series forecasting. In SST forecasting, the works proposed in the literature commonly use a simple method to model the time series under analysis. These approaches employ mainly linear statistical models or nonlinear ML models for this task. To fulfill this gap, we perform an empirical evaluation of hybrid systems that use error series modeling in the context of SST time series forecasting. This experimental analysis is of crucial importance because of the adoption of hybrid systems: (i) it generally leads to more accurate results than single models in complex time series modeling; (ii) it is an efficient way of dealing with the problem of model selection with little extra effort; and (iii) it is an effective manner to correct biased and/or misspecified forecasters. However, the best combination approach between the forecasters of time series and the residuals still is an open question and not yet investigated in the context of SST forecasting. This work evaluates the performance of hybrid systems combining ML models for SST time series forecasting unprecedently. In this sense, the objective of this paper is threefold: (a) evaluate whether the residual modeling is an advantageous approach to increasing ML models’ accuracy for SST time series; (b) evaluate two well-known forms of combination (linear and nonlinear) of the literature employing ML models; (c) analyze, for each SST series, which combination is most suitable.

The main contributions of this work can be summarized as:

- Proposal of a hybrid system methodology to improve the accuracy of ML models in the SST forecasting;
- The performance evaluation of two hybrid systems in the 1-day ahead forecasting SST using three well-known measures: mean square error (MSE), mean absolute percentage error (MAPE) and mean absolute error (MAE);
- The development of two versions of each analyzed hybrid system using well-known ML models: SVR and LSTM;
- The hybrid systems employing the SVR achieved on average a percentage gain compared to its respective single model of 80.27%, 61.72%, and 60.21% for MSE, MAPE, and MAE, respectively;
- The hybrid systems using the LSTM attained an average percentage gain concerning its respective single model of 73.16%, 57.90%, and 56.98% for MSE, MAPE, and MAE, respectively;
- The results show that, in general, the developed hybrid systems overcame literature statistical and ML models in the SST forecasting context.

The remainder of the paper is organized as follows. “Related works” section shows the related works of hybrid systems that deal with residual series modeling. This section describes the evaluated hybrid systems in the SST forecasting: Perturbative approach ("The perturbative approach" section) and NoLiC ("The NoLiC method" section). “PIRATA data set” section presents the data set extracted from the PIRATA project website. “Experimental protocol” section shows the experimental protocol used in this work. In “Simulations and experimental results” section, the results and discussions are presented. Finally, “Discussion” section shows the concluding remarks and suggestions for future works.

**Related works**

Combining models is one of the most common alternatives to enhance the accuracy of forecasting systems. In the literature, there are two well-established approaches: ensembles and hybrid systems that perform residuals modeling. Both theoretical and empirical results indicate that the latter approach is an interesting strategy to increase the robustness and accuracy of the forecasts.

The general architecture of a hybrid system that performs the residuals modeling can be divided into three main steps: time series forecasting, error series forecasting, and the combination of the two first steps. Equation (1) shows a general view of this architecture, where the final output of the hybrid system $\tilde{y}_t$ is given by a function $f(.)$ that combines the forecast of the time series $\hat{P}_0$ with the forecast of the residuals $P_1$ to estimate $y_t$. 

$$\tilde{y}_t = f(\hat{P}_0, P_1)$$
$P_0$ is the forecast of the time series given by the $M_0$ model (Eq. 2), and $P_1$ is the forecast of the residual series ($e_{t-1}, \ldots, e_{t-n}$) given by the $M_1$ model (Eq. 3). The residual series is calculated as the difference between the predicted and the actual values.

$$\hat{y}_t = f(P_0, P_1),$$

where

$$P_0 = M_0(y_{t-1}, \ldots, y_{t-m}),$$

and

$$P_1 = M_1(e_{t-1}, \ldots, e_{t-n}),$$

where $m$ and $n$ are the time lags used as input to the $M_0$ and $M_1$ models, respectively. The time lags can be defined using the auto-correlation function (ACF), partial auto-correlation function (PACF), or some searching algorithm. 

Based on the general architecture described by the $f$ function in Eq. (1), two classes of hybrid systems have been studied for real-world time series modeling: a combination of linear statistical methods with nonlinear Machine Learning (ML) models and a combination of ML models. For simplicity, the first class is denominated as a hybrid system and the second as a combination of ML models. The hybrid system class is described in “Hybrid systems—combining linear and nonlinear models” section. Techniques that combine ML models are presented in “Combining nonlinear models” section. This section also describes two recent techniques: the perturbative approach and NoLiC.

### Hybrid systems—combining linear and nonlinear models.

Linear statistical models have been combined with nonlinear ML models based on the assumption that real-world time series generally present linear and nonlinear patterns. Thus, in this hybrid system class, statistical models are used as $M_0$, and ML models are employed as $M_1$ intended to deal with linear and nonlinear patterns separately.

The $f$ function that is responsible for combining $P_0$ with $P_1$ can be either linear or nonlinear. The linear combination, which is more commonly used in the literature, consists of a non-trainable rule, such as the sum. This combination has been successfully used in several applications, for instance: financial indexes, wind speed, groundwater level fluctuations, the prevalence of schistosomiasis in humans, particulate matter, and water quality.

Zhang showed the linear combination:

$$\hat{y}_t = P_0 + P_1,$$

where $P_0$ is the forecasting of a linear statistical model ($M_0$), $P_1$ is the forecasting of an ML model ($M_1$) applied to the residual of the time series, and $\hat{y}_t$ is the final prediction of the hybrid system performed by the linear combination. In his experiments, $M_0$ was defined as an ARIMA and $M_1$ as an MLP neural network.

Despite being widely used in the literature, the linear combination of the forecasts $P_0$ and $P_1$ can underestimate, or degenerate the accuracy of the initial model ($M_0$), since there may be no additive relationship between linear and nonlinear forecasts.

Based on this assumption, Khashei and Bijari proposed a nonlinear combination of the forecasts to overcome the limitations of the linear combination. In their hybrid systems, the $f$ function (Eq. 1) is defined as an ML that receives as input $P_0$, the residual ($e_{t-1}, \ldots, e_{t-n}$), and the time series ($y_{t-1}, \ldots, y_{t-m}$), as shown in Eq. (5). They employed an ARIMA model for time series modeling as $P_0$, and an MLP as $P_1$.

$$\hat{y}_t = f(P_0, e_{t-1}, \ldots, e_{t-n}, y_{t-1}, \ldots, y_{t-m}).$$

In general, the nonlinear combination of the forecasts $P_0$ and $P_1$ reaches better results than the linear approach. However, there is no guarantee that the nonlinear combination is the most appropriate for modeling any temporal phenomena. Therefore, the best combination function of the forecasts of the time series (linear component) and the residual series (nonlinear component) is unknown, being still a research challenge in the hybrid systems research field.

### Combining nonlinear models.

Nonlinear ML models have been combined based on the assumption that adopting only one single model can be inadequate to real-world time series forecasting. The underperforming of a single ML model can occur due to problems caused by overfitting, underfitting, or misspecification.

In Ginzburg and Horn, two MLPs are combined linearly following the same idea shown in Eq. (4). Thus, the time series forecast ($P_0$) is performed by the first MLP ($M_0$), and its residuals are modeled by the second MLP ($M_1$), generating the forecast of the residuals ($P_1$). In this sense, the $M_1$ model is employed to uncover and model temporal correlations found in the residuals of $M_0$, thus correcting the original forecast ($P_0$). This premise is based on biological systems that commonly deal with complex tasks through subsystems. Later stages of consecutive subsystems (networks) refine the response of earlier ones, improving the performance of the entire biological system. This principle was also successfully employed in atmospheric pollution forecasting.

The next two subsections show two recent approaches that combine nonlinear models: the perturbative approach and NoLiC.
The perturbative approach. The linear combination of ML models proposed in28 was generalized in22, which employed the perturbation theory concept that was previously applied in many areas, such as physics, chemistry, and mathematics46,47. The idea is to initiate the forecasting of a time series using a first estimation (forecast \(P_0\)). Then, \(p\) new forecasts \((P_1 + P_2 + \cdots + P_p)\) are added to make a partial forecast ever closer to the real solution \(P\). Mathematically,

\[
P = P_0 + P_1 + P_2 + \cdots + P_p,
\]

where \(P\) is the desired solution (perfect forecasting), \(P_0\) is the series forecast, and the term of the major contribution to \(P\), and \(P_1, P_2, \ldots, P_p\) are \(p\) higher-order terms (residual forecasts). Then, \(P_1\) is the forecast of the residuals of \(M_0\), \(P_2\) is the forecast of the residuals of \(M_0 + M_1\), \(P_3\) is the forecast of the residuals of \(M_0 + M_1 + M_2\), and \(P_p\) is the forecast of the residuals of \(M_0 + M_1 + M_2 + \cdots + M_p\). Theoretically, the corrections generated by the residual forecasts \((P_1, P_2, \ldots, P_p)\) decrease since, at each perturbation \(i\), the residual series \(E_i\) present values closer to zero. In practice, the contribution of the residual forecasts \((P_1, P_2, \ldots, P_p)\) depends on the specification and training of the model \(M_i\).

Algorithms 1 and 2 show the training and the testing phases of the perturbative approach22, respectively.

Algorithm 1: Training phase of the perturbative approach

1: Input: \(Y = [y_0, y_1, \ldots, y_r]\)
2: Input: \(pMax\)
3: Output: \(M_0, M_1, M_2, \ldots, M_p\)
4: Output: \(E_1, E_2, \ldots, E_p\)
5: \(M_0 \leftarrow \text{modelTraining}(Y)\)
6: \(P_0 \leftarrow \text{forecast}(M_0, Y)\)
7: \(P \leftarrow P_0\)
8: \(i \leftarrow 1\)
9: \(E_i \leftarrow (Y - P)\)
10: while \(i \leq pMax\) do
11: \(M_i \leftarrow \text{modelTraining}(E_i)\)
12: \(P_i \leftarrow \text{forecast}(M_i, E_i)\)
13: \(P \leftarrow P + P_i\)
14: if errorVal(\(P_0 + \cdots + P_i\)) > errorVal(\(P_0 + \cdots + P_{i-1}\)) then
15: \(\text{break}\)
16: \(E_i \leftarrow (Y - P)\)
17: \(P \leftarrow i\)
18: \(i \leftarrow i + 1\)
19: end while

The training phase is divided into two general steps: training the time series forecasting model (lines 5–9) and training the correction models based on the residual series (lines 10–19). The algorithm’s input is the training set of the time series, and the outputs are \(p\) residuals (error series \(E\)) and \(p + 1\) trained models \((\{M_0, M_1, \ldots, M_p\})\). The training phase has two stop criteria: the maximum number of perturbations \((pMax)\) or an increase in the error value in the validation set concerning antecedent perturbation (lines 14 to 15).

In line 4, the initial model \((M_0)\) is trained using the training set \(Y\), generating the time series forecast \(P_0\) in line 5. \(P_0\) is the main contributor to the final solution \(P\). In line 8, the first error series \((E_1)\) is generated. The error series consists of the difference between the actual series and the estimated values provided by the perturbative approach \((P)\).

After, the perturbative terms are generated. Each \(M_i\) model is trained to forecast \(E_i\) (line 10), which is the difference between \(Y\) and \(P\) (line 14). At each iteration of the loop (lines 9–15), a new perturbative term is generated (lines 10–11) and added to the final solution (line 12). At the end of the training phase, \(P\) is the sum of the \(p + 1\) forecasts \((P_0, P_1, P_2, \ldots, P_p)\) of the \(p + 1\) models \((M_0 + M_1 + M_2 \cdots + M_p)\). Lines 10 to 14 are executed until the stopping criterion is reached.
The testing phase (Algorithm 2) is divided into two steps: forecast of the time series (line 5) and forecast of the perturbative terms (lines 8–12). Lines 5 and 6 show the generating of $P_0$ and its inclusion in the final output $P$ of the perturbative approach. Lines 8 to 12 show the second part, where the perturbative terms are generated for the test point (observation of the test sample). So, each model ($M_1, M_2, \ldots, M_p$) generates the forecasting of its respective error series ($E_1, E_2, \ldots, E_p$). This loop is repeated ($p$) times, which is the number of perturbations defined in the training phase. After, each perturbative term is forecasted (line 9) and added to the solution $P$ using a linear combination (line 10), generating the final forecasting.

**Algorithm 2** Testing phase of the perturbative approach

1. **Input:** $Y_q$  
   \(\triangleright \) inputs for a test pattern ($M_0$ model)
2. **Input:** $E_1, E_2, \ldots, E_p$  
   \(\triangleright \) inputs for perturbative models ($M_1, \ldots, M_p$)
3. **Input:** $M_0, M_1, M_2, \ldots, M_p$  
   \(\triangleright \) $p + 1$ models
4. **Output:** $P$  
   \(\triangleright \) final forecast
5. $P_0 \leftarrow M_0(Y_q)$
6. $P \leftarrow P_0$
7. $i \leftarrow 1$
8. **while** $i \leq p$ **do**
   9. $P_t \leftarrow M_t(E_i)$
   10. $P \leftarrow P + P_t$
   11. $i \leftarrow i + 1$
12. **end while**

The testing phase (Algorithm 2) is divided into two steps: forecast of the time series (line 5) and forecast of the perturbative terms (lines 8–12). Lines 5 and 6 show the generating of $P_0$ and its inclusion in the final output $P$ of the perturbative approach. Lines 8 to 12 show the second part, where the perturbative terms are generated for the test point (observation of the test sample). So, each model ($M_1, M_2, \ldots, M_p$) generates the forecasting of its respective error series ($E_1, E_2, \ldots, E_p$). This loop is repeated ($p$) times, which is the number of perturbations defined in the training phase. After, each perturbative term is forecasted (line 9) and added to the solution $P$ using a linear combination (line 10), generating the final forecasting.

**The NoLiC method.** The NoLiC method\(^{23}\) employs an adaptive combination of ML models with other techniques using the residual series. This combination method does not presuppose a linear combination as other works\(^{16,22,28}\). The idea is to find a combination function between $P_0$ and $P_1$ using an ML model that is flexible, capable of performing linear and nonlinear modeling.

The Nonlinear Combination (NoLiC) method is composed of three steps: forecast of the time series ($P_0$), forecast of the residuals ($P_1$), and the combination $f(.)$ of $P_0$ and $P_1$. Figure 1 shows the training and testing phases of the NoLiC method.

The training phase receives as input the training set and generates three trained models ($M_0, M_1, \text{and } M_C$) as outputs. Similarly to other works\(^{15,28}\), the models $M_0$ and $M_1$ are employed to forecast the time series and the
The selected locations can be seen in Fig. 2, represented by red points. Table 1 shows the characteristics of data for modeling with ML models. These data sets do not present interruptions and are located in different regions. The forecast of the error series established kernel function in the time series forecasting area. RBF kernel also was successfully employed in a set of hyperparameters. To the kernel SVR, the Radial Basis Function (RBF) was selected because it is a well-established kernel function in the time series forecasting area. However, there is no guarantee that the performance depends on adjusting the parameters and training of the M_C model, which is a complex task since it is related to forecasts of M_0 and M_1. Thus, investigating how to combine the forecasts of the time series and its error series is a crucial issue in the definition of the hybrid system since it is closely related to its accuracy.

**Remarks.** The combination methods described in “The perturbative approach” and “The NoLiC method” sections have different characteristics. The perturbative method can extract information from more than one residual series. However, this method supposes that the models should be combined using a simple sum rule. In contrast, the NoLiC method supposes a nonlinear combination between the forecasts and the residuals. The NoLiC method employs an ML model aiming to find a combination more suitable than a simple sum. However, there is no guarantee that the M_C model leads to the best accuracy of the hybrid system. The optimum performance depends on adjusting the parameters and training of the M_C model, which is a complex task since it is related to forecasts of M_0 and M_1. Thus, investigating how to combine the forecasts of the time series and its error series is a crucial issue in the definition of the hybrid system since it is closely related to its accuracy.

**PIRATA data set**
The Pilot Research Moored Array in the Tropical Atlantic (PIRATA) was developed by a network of observatories composed of many countries, such as Brazil, France, and the United States. This project has the objective to improve the knowledge about atmospheric variations in the tropical Atlantic Ocean. The climatic variations in this area can influence the development of droughts, floods, severe storms, and even hurricanes, affecting millions of people in South America and Africa.

The Pilot Research Moored Array in the Tropical Atlantic (PIRATA) was developed by a network of observatories composed of many countries, such as Brazil, France, and the United States. This project has the objective to improve the knowledge about atmospheric variations in the tropical Atlantic Ocean. The climatic variations in this area can influence the development of droughts, floods, severe storms, and even hurricanes, affecting millions of people in South America and Africa.

The PIRATA project has buoys in the ocean, where meteorological variables are collected, such as shortwave radiation, relative humidity, air temperature, and ocean surface temperature. All data are gathered and transmitted by satellite and are available on the project web page (https://www.pmel.noaa.gov/gtmba/pirata).

We aim to perform the forecast of the sea surface temperature of three regions: 8° N 38° W (Fig. 3a), 10° S 10° W (Fig. 3b), and 4° N 23° W (Fig. 3c). These locations were selected because they have an appropriate amount of data for modeling with ML models. These data sets do not present interruptions and are located in different regions. The selected locations can be seen in Fig. 2, represented by red points. Table 1 shows the characteristics of each time series used in this work.

**Experimental protocol**
The experiments evaluate two machine learning models: support vector machines (SVR) and long short-term memory (LSTM). These models were chosen because they reached relevant results regarding accuracy for the SST forecasting task. SVR is an interesting choice because it employs a quadratic optimization procedure to solve a convex constrained problem, with a single solution. Therefore, in contrast to methods such as neural networks where several local minima can be achieved, the uniqueness of the solution of SVR is obtained given a set of hyperparameters. To the kernel SVR, the Radial Basis Function (RBF) was selected because it is a well-established kernel function in the time series forecasting area. RBF kernel also was successfully employed in the SST forecasting and has been widely used in hybrid systems. Besides, the RBF is considered the default SVR kernel in the Sklearn library, which is now the most popular package for creating SVR models in Python. The RBF’s popularity can be explained by its finite and localized responses across the entire range of the x-axis, so it does not need previous assumptions about the data and adds few parameters to the SVR model (Cost and Gamma).
LSTM was selected because it is one of the state-of-the-art ML models in time series forecasting. It has outperformed traditional neural networks in several applications\textsuperscript{56}. Its ability to deal with short or long-term temporal dependencies can be promising in the SST time series modeling\textsuperscript{57}.

For the combination approach, the same model (SVR or LSTM) are employed for all stages. For the perturbative approach, the best number of perturbations (or corrections) was selected based on the MSE value in the validation set having a upper limit of four perturbations. For all models, a grid search approach was performed for

---

**Figure 3.** Sea surface temperature time series.
selecting the best configuration based on the MSE value in the validation set. The data used in the experimental simulations were scaled into the interval [0.1, 0.9], similar to 3.

Table 2 shows the set of parameters investigated for each model in the 1-day-ahead forecasting scenario. The number of input lags used in the grid search was selected based on PACF. For S1, the lags 1, 3, 4, 11, and 15 presented significant linear correlations. For S2, PACF selected the lags 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 28, and 30, and for S3, the lags 1, 2, 3, 5, 17, and 18 presented relevant correlation.

Table 3 shows the values selected for single and combination approaches for 1 day ahead SST forecasting for each study case (S1, S2, and S3 series). It is important to highlight that all combination approaches use the same M0, and the NoLiC method employs the same M0 and M1 of the perturbative approach. So, it is possible to compare the performance of the combination approaches directly. For all series, the perturbation approach employed three perturbations for the SVR model and two perturbations for the LSTM model.

The performance of the approaches are evaluated using three performance metrics applied to the context of sea surface temperature forecast3,4: mean square error (MSE), mean absolute percentage error (MAPE) and mean absolute error (MAE). Equations (6), (7) and (8) show the MSE, MAPE and MAE metrics, respectively.

\[
\text{MSE} = \frac{1}{N} \sum_{t=1}^{N} (y_t - \hat{y}_t)^2, \tag{6}
\]

\[
\text{MAPE} = \frac{100}{N} \sum_{t=1}^{N} \left| \frac{y_t - \hat{y}_t}{y_t} \right|, \tag{7}
\]

\[
\text{MAE} = \frac{1}{N} \sum_{t=1}^{N} |y_t - \hat{y}_t|, \tag{8}
\]

where \(N\) represents the time series length, \(y_t\) the true value at time \(t\) and \(\hat{y}_t\) is the forecast at time \(t\). For all metrics, the lower the values, the better the results. A percentage gain/loss measure (Eq. 9) is used to compare the combination approaches with the single models.

\[
\text{PC} = \frac{\text{Metric}_{\text{sm}} - \text{Metric}_{\text{comb}}}{\text{Metric}_{\text{sm}}} \times 100, \tag{9}
\]

where \(\text{Metric}_{\text{sm}}\) and \(\text{Metric}_{\text{comb}}\) represent the MSE values reached by single models and combination approaches, respectively. In this way, the higher the PC, the better the performance of the combination approach in relation to the single model.

---

**Table 1.** Properties of the SST time series used in this work.

| Adopted acronym | S1       | S2       | S3       |
|-----------------|----------|----------|----------|
| Time series     |          |          |          |
| Localization    | 8° N 38° W | 10° S 10° W | 4° N 23° W |
| Time unit       | Daily    | Daily    | Daily    |
| Start date      | Oct 08, 00 | Sep 09, 99 | May 10, 07 |
| Start date (test set) | Oct 01, 06 | Sep 07, 10 | Jun 30, 16 |
| End date        | Oct 07, 07 | Sep 08, 11 | Jul 02, 17 |
| Total size      | 2545     | 4376     | 3699     |
| Training sample size | 1815     | 3646     | 2969     |
| Validation sample size | 365      | 365      | 365      |
| Testing sample size | 365      | 365      | 365      |

**Table 2.** Values of the parameters of the SVR and LSTM models.

| Model | Parameters            | Values                          |
|-------|-----------------------|---------------------------------|
| SVR   | Gamma                 | [0.001, 1]                      |
|       | Cost                  | [0.1, 1, 100]                   |
|       | Tolerance             | [0.001, 0.01, 0.1]              |
|       | Kernel                | Radial basis function \(^3\)   |
| LSTM  | Units in hidden layer | [2, 5, 10]                      |
|       | Algorithm             | Adam \(^{19,58}\)               |
The SVR and LSTM models were implemented in the Python programming language using the Sklearn\textsuperscript{54} and Keras\textsuperscript{59} libraries. The experimental simulations were performed in a computer with a single Intel Core i7-7500 CPU and 20 GB RAM.

The experimental comparison was carried out among the single models, the hybrid approaches, and the following literature models: Exponential Smoothing (ETS)\textsuperscript{60}, Convolution LSTM (ConvLSTM)\textsuperscript{11,50}, and the Nonlinear Autoregressive Exogenous (NARX)\textsuperscript{1}.

Exponential Smoothing (ETS) is a traditional statistical method employed in time series forecasting\textsuperscript{60,61}. It is a versatile method due to its ability to model time series with/without trend and seasonality components. However, the ETS can reach a limited performance in forecasting time series that present nonlinear patterns\textsuperscript{62}. The experiments with ETS were carried out using the Statsmodel library of Python\textsuperscript{63}.

The Convolutional Long Short-Term Memory (ConvLSTM)\textsuperscript{11,50} is a Deep Learning technique able to model spatiotemporal correlations. The ConvLSTM models spatial and temporal patterns using convolution and LSTM layers. This technique attained higher accuracy than other ML models to SST time series forecasting tasks\textsuperscript{11,50}. On the other hand, its training can be costly computationally due to the number of hyper-parameters that must be adjusted\textsuperscript{64}. In this work, the employed ConvLSTM used the configuration suggested in the SST forecasting works\textsuperscript{11,50}.

Nonlinear Autoregressive with Exogenous Input neural network (NARX) was proposed to model the nonlinear and autoregressive behaviors\textsuperscript{65}. NARX model was successfully used to predict SST anomalies in the western Indian Ocean region\textsuperscript{1}. Despite being able to forecast seasonal anomaly trends, the NARX performance is highly sensitive to parameters specification\textsuperscript{1}.

**Simulations and experimental results**

Table 4 shows the results regarding MSE, MAPE and MAE for the test set of the S1, S2, and S3 series, for 1-day ahead forecasting. In that table is possible to compare the performance of the hybrid systems with single and literature models.

For the S1 time series, all hybrid systems improved the accuracy of their respective single model, reaching better MSE, MAPE, and MAE values than literature models. In particular, the NoLiC employing the LSTM model attained the best result in all considered metrics. Regarding MSE, the hybrid system versions obtained an error of one order of magnitude smaller than their respective single models, for instance, 3.97E−04 for NoLiC+LSTM and 6.89E−04 for LSTM.

The S2 and S3 series follow the same behavior: all hybrid systems versions improved the performance of their respective single models for the evaluated metrics. For the S2 time series, the perturbation approach using the SVR model attained the best performance in terms of MSE, MAPE, and MAE. This hybrid system version, which employed three perturbations ($P_0 + P_1 + P_2 + P_3$), improved the MSE value in two orders of magnitude regarding the single SVR.

Hybrid systems that use the LSTM model deserve special attention for the S3 time series. In this case, the NoLiC model attained the best MSE value, while the perturbative approach ($P_0 + P_1 + P_2$) obtained the smallest MAPE.
and MAE. Both hybrid system versions of single SVR and LSTM improved the MSE value in one order of magnitude. Among the single and literature models, the NARX\(^1\) achieved the best results for the evaluated times series.

Tables 5 and 6 show the percentage difference (Eq. 9) in terms of the MSE metric between the literature models and the perturbative and NoLiC approaches, respectively. The tables show that the hybrid systems improved the performance of both single models for the S1 series. The NoLiC using the LSTM model attained an improvement greater than 65% for all evaluation metrics (Table 5). The versions of the hybrid systems attained a superior performance, at least 20% when compared with the literature. Figure 4a,b show the forecasts of the S1 series test set of the hybrid approaches using SVR and LSTM, respectively. It can be seen that both hybrid systems were able to improve the forecasting of the single models. Both hybrid approaches achieved forecasts closer to the real when compared with the initial model.

For S2, Table 5 shows that the perturbative approach reached an improvement higher than 30% in all comparisons. This approach with SVR obtained a percentage gain regarding SVR of 98.93%, 90.56%, and 91.09% for MSE, MAPE, and MAE, respectively. Table 6 shows the NoLiC using LSTM model attained a gain concerning single LSTM of 98.42%, 88.40% and 88.87% for MSE, MAPE, and MAE, respectively. Figure 4c,d show the forecasts of the S2 series test set of the hybrid approaches using SVR and LSTM, respectively. Both figures show that the hybrid systems improved the forecast of the single models. In both comparisons, it is possible to verify that the forecast of the hybrid systems using SVR or LSTM is closer to the test set of S2 than the respective single model. Supplementary Information presents additional analyzes.

### Table 4. Comparison in terms of MSE, MAPE, and MAE of the combination approaches with single statistical and Machine Learning models of the literature applied to the SST daily forecasting. For each data set, the best value of the metrics is highlighted in bold.

| Dataset | Approach | Model    | MSE     | MAPE   | MAE     |
|---------|----------|----------|---------|--------|---------|
| S1      | Perturbative | SVR      | 6.89E−04 | 3.61   | 1.85E−02 |
|         |          | LSTM     | 6.89E−04 | 3.61   | 1.85E−02 |
|         | NoLiC    | SVR      | 6.71E−04 | 3.60   | 1.84E−02 |
|         |          | LSTM     | 3.97E−04 | 2.65   | 1.38E−02 |
|         | Literature | ETS\(^6\) | 5.36E−03 | 12.30  | 5.85E−02 |
|         |          | Single SVR\(^{1,13}\) | 3.78E−03 | 9.40   | 4.56E−02 |
|         |          | Single LSTM\(^{1,19,39}\) | 5.06E−03 | 10.83  | 5.21E−02 |
|         |          | ConvLSTM\(^{1,34}\) | 1.39E−03 | 5.53   | 2.72E−02 |
|         |          | NARX\(^1\) | 8.77E−04 | 5.28   | 2.51E−02 |
| S2      | Perturbative | SVR      | 1.08E−04 | 1.97   | 7.82E−03 |
|         |          | LSTM     | 1.08E−04 | 1.98   | 7.86E−03 |
|         | NoLiC    | SVR      | 1.01E−03 | 4.52   | 2.08E−02 |
|         |          | LSTM     | 1.31E−04 | 2.19   | 8.82E−03 |
|         | Literature | ETS\(^6\) | 3.87E−03 | 13.93  | 5.43E−02 |
|         |          | Single SVR\(^{1,13}\) | 1.01E−02 | 20.83  | 8.78E−02 |
|         |          | Single LSTM\(^{1,19,39}\) | 8.30E−03 | 18.86  | 7.93E−02 |
|         |          | ConvLSTM\(^{1,34}\) | 8.59E−04 | 5.92   | 2.33E−02 |
|         |          | NARX\(^1\) | 2.03E−04 | 3.04   | 1.16E−02 |
| S3      | Perturbative | SVR      | 9.38E−04 | 3.80   | 2.36E−02 |
|         |          | LSTM     | 7.91E−04 | 3.41   | 2.11E−02 |
|         | NoLiC    | SVR      | 9.02E−04 | 3.75   | 2.34E−02 |
|         |          | LSTM     | 7.74E−04 | 3.43   | 2.15E−02 |
|         | Literature | ETS\(^6\) | 5.78E−03 | 9.00   | 5.56E−02 |
|         |          | Single SVR\(^{1,13}\) | 2.58E−03 | 6.20   | 3.75E−02 |
|         |          | Single LSTM\(^{1,19,39}\) | 1.15E−03 | 3.96   | 2.44E−02 |
|         |          | ConvLSTM\(^{1,34}\) | 1.27E−03 | 4.30   | 2.06E−02 |
|         |          | NARX\(^1\) | 9.18E−04 | 3.89   | 2.35E−02 |

Discussion. To verify if there are (or not) significant statistical differences between the hybrid systems and literature approaches, we employed the Diebold–Mariano statistical test\(^66\). We use MSE since it is the target metric employed to guide the search of the parameters of the models. Table 7 shows that both versions of the perturbative approach attain MSE values statistically different from the single and literature models, i.e., the p...
value is smaller than the significance level adopted (0.05) in all comparisons. The NoLiC employing LSTM also reached results statistically better than other models. Only the NoLiC version using SVR attained an MSE worse than NARX1 and ConvLSTM11,50 models.

Table 5. Percentage difference between the perturbative approach and literature models for MSE, MAPE, and MAE.

| Dataset | Model            | Perturbative approach | LSTM  |
|---------|------------------|-----------------------|-------|
|         |                  | SVR                   | MAE   |
|         |                  | MSE     | MAPE   | MAE     | MSE     | MAPE   | MAE     |
| S1      | ETS60            | 87.14   | 70.63  | 68.27  | 87.14   | 70.63  | 68.27  |
|         | Single SVR134    | 81.78   | 61.59  | 59.29  | 81.78   | 61.59  | 59.29  |
|         | Single LSTM1826  | 86.37   | 66.65  | 64.41  | 86.37   | 66.65  | 64.41  |
|         | ConvLSTM1535     | 50.25   | 34.73  | 31.85  | 50.25   | 34.73  | 31.85  |
|         | NARXX            | 21.44   | 31.63  | 26.15  | 21.44   | 31.63  | 26.15  |
| S2      | ETS60            | 97.20   | 85.88  | 85.59  | 97.20   | 85.79  | 85.51  |
|         | Single SVR134    | 98.93   | 90.56  | 91.09  | 98.92   | 90.50  | 91.05  |
|         | Single LSTM1826  | 98.70   | 89.57  | 90.13  | 98.69   | 89.51  | 90.08  |
|         | ConvLSTM1535     | 87.39   | 66.75  | 66.40  | 87.38   | 66.55  | 66.21  |
|         | NARXX            | 46.76   | 35.27  | 32.71  | 46.71   | 34.88  | 32.34  |
| S3      | ETS60            | 83.76   | 57.73  | 57.62  | 86.31   | 62.14  | 61.98  |
|         | Single SVR134    | 63.64   | 38.66  | 37.20  | 69.35   | 45.06  | 43.65  |
|         | Single LSTM1826  | 18.07   | 3.94   | 3.36   | 30.93   | 13.97  | 13.30  |
|         | ConvLSTM1535     | 25.90   | 9.25   | 9.54   | 20.72   | 18.57  |        |
|         | NARXX            | −2.15   | 2.16   | −0.30  | 13.89   | 10.01  |        |

Table 6. Percentage difference between the NoLiC and literature models for MSE, MAPE, and MAE.

| Dataset | Model            | NoLiC   | LSTM   |
|---------|------------------|---------|--------|
|         |                  | SVR     | MAE    |
|         |                  | MSE     | MAPE   | MAE     | MSE     | MAPE   | MAE     |
| S1      | ETS60            | 87.48   | 70.76  | 68.58  | 92.59   | 78.46  | 76.36  |
|         | Single SVR134    | 82.26   | 61.75  | 59.69  | 89.51   | 71.83  | 69.67  |
|         | Single LSTM1826  | 86.73   | 66.79  | 64.75  | 92.15   | 75.54  | 73.48  |
|         | ConvLSTM1535     | 51.57   | 35.00  | 32.51  | 71.35   | 52.13  | 49.23  |
|         | NARXX            | 23.51   | 31.91  | 26.87  | 54.75   | 49.86  | 44.98  |
| S2      | ETS60            | 73.98   | 67.52  | 61.60  | 96.62   | 84.29  | 83.74  |
|         | Single SVR134    | 90.01   | 78.28  | 76.28  | 98.70   | 89.50  | 89.95  |
|         | Single LSTM1826  | 87.87   | 76.01  | 73.72  | 98.42   | 88.40  | 88.87  |
|         | ConvLSTM1535     | −21.24  | 23.53  | 10.49  | 84.76   | 63.02  | 62.09  |
|         | NARXX            | −359.08 | −48.86 | −79.26 | 35.64   | 28.00  | 24.09  |
| S3      | ETS60            | 84.38   | 58.30  | 57.94  | 86.60   | 61.86  | 61.29  |
|         | Single SVR134    | 65.03   | 39.50  | 37.68  | 70.00   | 44.66  | 42.64  |
|         | Single LSTM1826  | 21.20   | 5.25   | 4.10   | 32.41   | 13.34  | 11.74  |
|         | ConvLSTM1535     | 28.73   | 12.69  | 9.94   | 38.87   | 20.14  | 17.11  |
|         | NARXX            | 1.75    | 3.50   | 0.47   | 15.73   | 11.73  | 8.39   |

Table 8 shows the execution time (in seconds) of the testing phase calculated over 30 executions. The evaluated approaches presented an execution time smaller than 1 s in all data sets. It is important to highlight that the hybrid systems based on the LSTM model are more costly regarding computational effort than the ones based on SVR. For instance, the SVR’s perturbative approaches were less computationally costly than single LSTM for S1 and S2 series.

The complexity analysis of the hybrid system can be divided into $p$ steps, each one corresponding to the training of a model ($M_0, M_1, \ldots, M_p$). The evaluated hybrid systems are trained sequentially, and their training time can be described as $MT_0 + MT_1 + \cdots + MT_p$, where $MT$ is the training time of the model in a specific phase. In this way, the NoLiC approach is approximately three times more expensive than the single models, because the NoLiC uses three models ($M_0, M_1,$ and $M_C$), and the perturbative approach is approximately $p$ times more expensive than the single models because it uses $M_0, M_1, \ldots, M_p$ models. This work applies two compositions of hybrid systems, using SVR or LSTM. The SVR training process has a complexity of $O(ln^{1/2})$, where $l$ is the size of...
of the data set, and \( m \) represents the number of input features. The training process of the LSTM has a complexity of \( O(W) \), where \( W \) is the total number of parameters\(^{68}\).

**Conclusion**

The Sea Surface Temperature (SST) is an important environmental variable due to its strong relationship to climate, weather, and nature events, such as El Niño. So, the SST accurate forecast can support decisions in several science fields.
In this work, we evaluated two types of hybrid systems intending to improve the performance of single ML models in the task of SST forecast. The hybrid systems are evaluated in the 1-day-ahead forecasting scenario. The purpose was to correct biased and deteriorated forecasts of the ML models by modeling the error series. The Perturbative and NoLiC hybrid approaches employ linear and nonlinear combinations, respectively. For each approach, two versions were generated, one using SVR and another using LSTM as base models. All the models were evaluated in three data sets of different locations in the tropical Atlantic using traditional metrics (MSE, MAPE, and MAE) of the literature. Compared with the ML single models, the hybrid system approaches obtained a significant performance improvement (more than 20%).

Regarding the hybrid systems, it was possible to verify the influence of the combination function in their performance. The linear combination used by the Perturbative approach obtained the best performance in two out of three study cases regarding MSE. Although NoLiC employs a combination function more versatile than the simple sum, it could not overcome the linear combination in most cases. In particular, when the perturbative

| Dataset | Model | Perturbative | NoLiC |
|---------|-------|-------------|-------|
|         |       | SVR | LSTM | SVR | LSTM |
| S1      | ETS   | +   | +   | +   | +   |
|         | Single SVR | +   | +   | +   |
|         | Single LSTM | +   | +   | +   |
|         | NARX   | +   | +   | +   |
|         | ConvLSTM | +   | +   | +   |
| S2      | ETS   | +   | +   | +   |
|         | Single SVR | +   | +   | +   |
|         | Single LSTM | +   | +   | +   |
|         | NARX   | +   | +   | +   |
|         | ConvLSTM | +   | +   | +   |

Table 7. Results of the comparison of the hybrid systems using SVR and LSTM with single and literature models using Diebold–Mariano hypothesis test.

| Datasets | Model | Approach | Execution time |
|----------|-------|----------|----------------|
|          |       |          | Mean (Std)     |
| S1       | SVR   | Single   | 0.014 (0.002) |
|          |       | P0 + P1 + P2 + P3 | 0.053 (0.005) |
|          | NoLiC |          | 0.064 (0.120) |
|          |       | Single   | 0.117 (0.037) |
|          |       | P0 + P1 + P2 | 0.371 (0.088) |
|          | NoLiC |          | 0.261 (0.057) |
| S2       | SVR   | Single   | 0.011 (0.005) |
|          |       | P0 + P1 + P2 + P3 | 0.062 (0.011) |
|          | NoLiC |          | 0.303 (0.027) |
|          |       | Single   | 0.105 (0.009) |
|          |       | P0 + P1 + P2 | 0.371 (0.036) |
|          | NoLiC |          | 0.271 (0.016) |
| S3       | SVR   | Single   | 0.020 (0.005) |
|          |       | P0 + P1 + P2 + P3 | 0.100 (0.012) |
|          | NoLiC |          | 0.282 (0.024) |
|          |       | Single   | 0.098 (0.025) |
|          |       | P0 + P1 + P2 | 0.421 (0.107) |
|          | NoLiC |          | 0.338 (0.043) |

Table 8. Testing time in seconds of the single models and combination approaches for 1 day ahead forecasting. For each approach is presented the mean testing time and the respective standard deviation.
approach uses the LSTM as the base model, it reached the highest performance in three out of five cases. The LSTM's best performance compared to the other ML models can be attributed to its ability to capture long-term temporal dependencies due to its recurrent abilities, such as memory cells. In this way, the LSTM can consider previous training examples on its forecasting process, creating a better understanding of the past data and a more robust combination process.

It is crucial to remark that both single and hybrid models struggled to forecast extreme points. This issue is a challenging task in the time series literature. Another is the absence of sufficient extreme cases in the training set, which can bias the training process towards more regular cases and the applied target metric (MSE). The hybrid system's computational effort is the sum of the costs of modeling its counterparts and depends on each model's parameters set. The computational cost can be minimized in the test phase, parallelizing the time series and residual forecasting.

For future works, we intend to improve the accuracy of the hybrid systems to better forecast extreme values by automatically searching for the most suitable combination function. Besides, different base models, such as convolutional neural networks, echo state networks, and decision trees for regression, can be investigated.

Received: 28 August 2021; Accepted: 17 December 2021
Published online: 11 January 2022

References

1. Mahongo, S. & Deo, M. Using artificial neural networks to forecast monthly and seasonal sea surface temperature anomalies in the Western Indian Ocean. *Int. J. Ocean Clim. Syst.* **4**, 133–150 (2013).

2. Reynolds, R. W. & Smith, T. M. Improved global sea surface temperature analyses using optimum interpolation. *J. Clim.* **7**, 929–948 (1994).

3. Lins, I. D., Araujo, M., das Chagas Moura, M., Silva, M. A. & Droguett, E. L. Prediction of sea surface temperature in the tropical Atlantic by support vector machines. *Comput. Stat. Data Anal.* **61**, 187–198 (2013).

4. Cho, J. et al. A study on the relationship between Atlantic Sea surface temperature and Amazonian greenness. *Ecol. Inform.* **5**, 367–378 (2010).

5. Garcia-Gorriz, E. & Garcia-Sanchez, J. Prediction of sea surface temperatures in the western Mediterranean Sea by neural networks.

6. Salles, R. Evaluating temporal aggregation for predicting the sea surface temperature of the Atlantic Ocean. *Ecol. Inform.* **36**, 94–105 (2016).

7. Tripathi, K., Das, I. & Sahai, A. Predictability of sea surface temperature anomalies in the Indian Ocean using artificial neural networks. *Indian J. Mar. Sci.* **35**, 210–220 (2006).

8. Garcia-Gorriz, E. & Garcia-Sanchez, J. Prediction of sea surface temperatures in the western Mediterranean Sea by neural networks using satellite observations. *Environ. Res. Lett.* **34**, 1–6 (2007).

9. Zhang, G., Patuwo, B. E. & Hu, M. Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **13**, 35–62 (1998).

10. Xiao, C. Predicting sea surface temperature: A comprehensive review and assessment. *Environ. Modell. Softw.* **120**, 104502 (2019).

11. Chatfield, C. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **50**, 159–175 (2003).

12. Haghbin, M., Sharafati, A., Motta, D., Al-Ansari, N. & Noghani, M. Applications of soft computing models for predicting sea–surface temperature. *Int. J. Ocean Clim. Syst.* **36**, 3855–3864 (2011).

13. Gheyas, I. A. & Smith, L. S. A novel neural network ensemble architecture for time series forecasting. *Neurocomputing* **74**, 3855–3864 (2011).

14. Chatfield, C. *Time Series Analysis: Forecasting and Control* (Wiley, 2015).

15. Michalski, R. S., Carbonell, J. G. & Mitchell, T. M. *Machine Learning: An Artificial Intelligence Approach* (Springer, 2013).

16. Sulaiman, J. & Wahab, S. H. Heavy rainfall forecasting model using artificial neural network for flood prone area. *IT Converg. Secur.* **449**, 68–76 (2018).

17. Basak, D., Pal, S. & Patranabis, D. C. Support vector regression. *Neural Inf. Process. Lett. Rev.* **11**, 203–224 (2007).

18. Zhang, G. P. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing* **50**, 159–175 (2003).

19. Khashi, M. & Bijari, M. A novel hybridization of artificial neural networks and ARIMA models for time series forecasting. *Appl. Soft Comput.* **11**, 2664–2675 (2011).

20. de Oliveira, J. F. & Ludermir, T. B. A hybrid evolutionary decomposition system for time series forecasting. *Neurocomputing* **180**, 27–34 (2016).

21. Silva, E. G., Domingos, S. d. O., Cavalcanti, G. D. & de Mattos Neto, P. S. Improving the accuracy of intelligent forecasting models using the perturbation theory. In 2018 International Joint Conference on Neural Networks (IJCNN) 1–7 (IEEE, 2018).

22. Santos, D. S. d. O. Jr., de Oliveira, J. F. & de Mattos Neto, P. S. An intelligent hybridization of ARIMA with machine learning models for time series forecasting. *Knowl. Based Syst.* **175**, 72–86 (2019).

23. de Oliveira, J. F., Silva, E. G. & de Mattos Neto, P. S. A hybrid system based on dynamic selection for time series forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **1–13**. https://ieeexplore.ieee.org/document/9340584 (2021).

24. de Mattos Neto, P. S., Ferreira, T. A., Lima, A. R., Vasconcelos, G. C. & Cavalcanti, G. D. A perturbative approach for enhancing the performance of time series forecasting. *Neural Netw.* **88**, 114–124 (2017).

25. McDermott, P. L. & Wilde, C. K. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *Entropy* **21**, 184 (2019).

26. Qian, S. *Entropy: A Comprehensive Treatment* (Springer, 2019).

27. Sun, Y. et al. Time-series graph network for sea surface temperature prediction. *Big Data Res.* **25**, 100237 (2021).

28. Ginzburg, I. & Horn, D. Combined neural networks for time series analysis. In NIPS’93: Proceedings of the 6th International Conference on Neural Information Processing Systems 224–231. https://dl.acm.org/doi/10.5555/2987189.2987218 (1993).

29. Pai, P.-F. & Lin, C.-S. A hybrid ARIMA and support vector machines model in stock price forecasting. *Omega* **33**, 497–505 (2005).

30. Ferrino, P. R. A., de Mattos Neto, P. S. & Ferreira, T. A. Error modeling approach to improve time series forecasters. *Neurocomputing* **153**, 242–254 (2015).

31. de Mattos Neto, P. S., Cavalcanti, G. D., Madeiro, F. & Ferreira, T. A. An approach to improve the performance of PM forecasters. *PLoS ONE* **10**, e0138507 (2015).

32. Gheyas, I. A. & Smith, L. S. A novel neural network ensemble architecture for time series forecasting. *Neurocomputing* **74**, 3855–3864 (2011).
The authors declare no competing interests.

Author contributions

P.S.G.M.N. and G.D.C.C. conceived the experiment(s), E.G.S and D.S.O.S.J. conducted the experiment(s), P.S.G.M.N., E.G.S and D.S.O.S.J. analysed the results. All authors wrote and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-04238-z.
