ISSN 2007-9737

HELI: An Ensemble Forecasting Approach for Temperature Prediction in the Context of Climate Change

Erick Estrada-Patiño[1], Guadalupe Castilla-Valdez[1], Juan Frausto-Solis[1,*], Juan Javier Gonzalez-Barbosa[1], Juan Paulo Sánchez-Hernández[2]

[1] Tecnológico Nacional de México,
Instituto Tecnológico de Ciudad Madero,
Mexico

[2] Universidad Politécnica del Estado de Morelos,
Information of Technology Division,
Mexico

estrada1792@gmail.com, gpe_cas@yahoo.com.mx,
jjgonzalezbarbosa@hotmail.com, juan.paulosh@upemor.edu.mx

Abstract. Climate change is a critical challenge, demanding the development of effective methods for temperature forecasting. Statistical and machine learning models emerge as promising alternatives. However, there is no widely accepted superior method; ensemble approaches integrate strategies that take advantage of each forecasting method. Ensemble methodologies combine methods, weighing their participation to integrate each of them. Forecasting researchers have shown that evolutionary algorithms are highly effective in achieving an ensemble that is at least as effective as the best single method. This paper presents HELI, a forecasting methodology designed to forecast the climate temperature variable; its architecture is modular, aiming to provide a flexible forecasting application in the climate change area. We present experimentation and a hypothesis test for a region in Mexico City and show HELI's competitiveness compared to leading strategies. Besides, we present experiments with other climate change variables that show HELI flexibility in the context of climate change.

Keywords. Ensemble methods, LSTM, CNN, evolutive ponderation.

1 Introduction

The current climate change phenomena are crucial problems that require exhaustive study [1]. Understanding and correcting these dynamics is essential to anticipate and mitigate their possible long-term consequences [2].

Besides, the United Nations Framework Convention on Climate Change, specifically the 2015 Paris Agreement, proposed that global average temperature increase should not exceed 1.5 °C, with a critical threshold of 2°C by 2100 [3]. This convention defined temperature as a fundamental variable for estimating the evolution and impacts of climate change [4, 5, 6, 7].

Therefore, it is imperative to research strategies and tools for predicting temperatures beyond seasonal periods to cover the medium and long term [8, 9]. Although traditional climate models use current and past observations, they are not accurate methods due to anthropogenic, natural, and other factors associated with climate change. On the other hand, forecasting models, whether classical or based on machine learning algorithms, are a valuable alternative.

These approaches model nonlinear relationships among several climate change variables based on historical data. Their effectiveness in many areas, including weather forecasting in the context of climate change, has been demonstrated and equated with conventional meteorological strategies [10, 11, 12]. Many authors have explored climate phenomena using statistical strategies such as SARIMA and Exponential Smoothing, which capture complex behaviors with good performance [8, 10].In addition, neural networks such as LSTM and CNN,

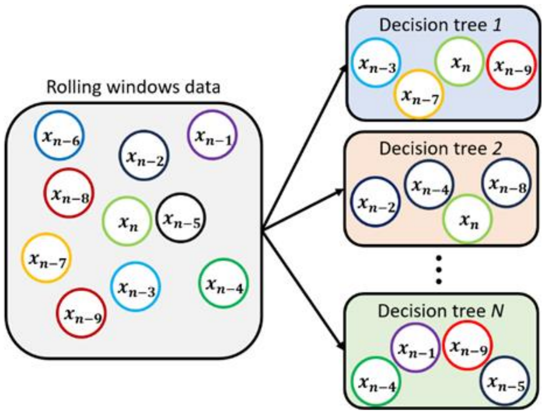

Fig. 1. Division of information into a random forest

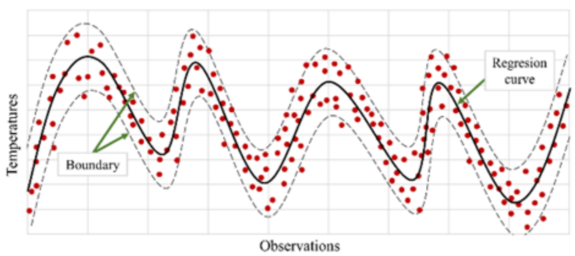

Fig. 2. Confidence intervals in support vector regression

Random Forest Regression (RFR), and Support Vector Regression have yielded competitive results [11, 12, 9]. The Forecasting and Machine Learning communities have observed that no method is superior to any other for all instances of a given problem in their areas [13, 14].

Given this difficulty of choice, hybrid and ensemble models have been proposed to enhance the forecast of a variable in any domain. This work proposes forecasting temperature; thus, we designed HELI to combine single forecasting methods with a relevant variable for climate change. Ensemble methods, in their general form, unify several strategies, taking advantage of the best of each one to deliver a high-quality integrated forecast [15, 16, 17].

These strategies have demonstrated success, highlighting the superior performance of SMYL in the M4 competition [18]. These are comparable to individual machine learning or classical strategies [18, 19]. In this paper, we present HELI, a methodology for temperature forecasting in the context of climate change. HELI incorporates robust preprocessing that facilitates the tuning and learning of multiple forecasting strategies, both classical and machine learning.

These strategies are combined and optimized for improved participation to produce a regression function that is at least as effective as the best single-trained method. The results evidence the competitiveness of HELI against leading state-of-the-art assembly forecasting approaches, with a low computational cost and high scalability and flexibility in adjusting the methods and their configuration in the assembly.

The structure of this paper is as follows. Section two presents a comprehensive review of the approaches and methodologies used and their advantages and disadvantages; we provide a solid context for the proposed methodology. Section three describes the HELI methodology in detail, including its main components, the configuration used in this work, and considerations for its replication.

The fourth section presents the results of a case study demonstrating the performance of HELI in temperature forecasting in Mexico; we decided to test HELI and present its forecasting results with other climate change variables, letting us evaluate its flexibility in scenarios different from the original. We discuss the implications of the results for the decision-making and future improvements. Finally, in section five we present the conclusions of this work.

2 Related Methods

In this paper, we explore two forecasting approaches applicable to time series: classical approaches and approaches supported by machine learning algorithms.

Classical forecasting approaches are strategies that base their predictive model on statistical methods, allowing the identification of possible trends and seasonal patterns in the data. These models are powerful because they generally involve tuning a few parameters and effectively adjusting to previously described behaviors. In addition, they have a low computational cost [20].

In contrast, forecasting strategies based on machine learning facilitate an exhaustive analysis of historical data, especially when these exhibit complex or highly nonlinear patterns [21]. These

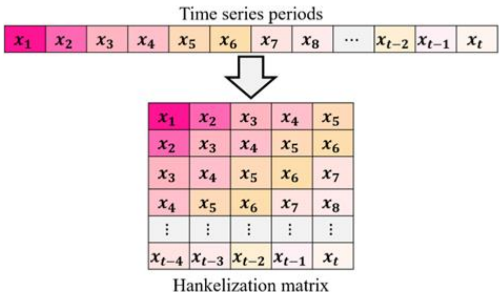

Fig. 3. Hankelization process

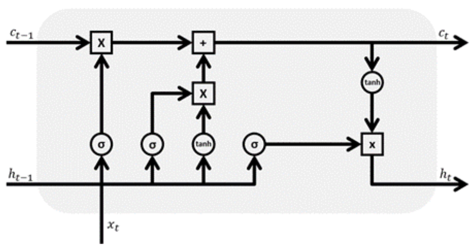

Fig. 4. LSTM cell

approaches have a high adaptive capacity and a remarkable tolerance to irregularities in the data; however, they usually entail a high computational cost, and the parameters to be adjusted are significantly more numerous compared to classical strategies [22].

Additionally, we explore strategies that evaluate the efficiency of algorithms beyond the improvement in error metrics on a data set [23]. Ranking methods, such as BIC or AIC, allow other metrics to be considered for scoring the effectiveness of an algorithm, with the understanding that a lower amount of error can mean an adequate strategy or an overfitted strategy for a given behavior [24].

These metrics make it possible to consider algorithms that present balanced results, translating into good performance in the long term [25]. Likewise, the weighting adjustment methods allow the solutions created by the ranking methods and some other random ones to evolve to find the best-weighted combination of the forecasting methods to issue a quality forecast.

2.1 Classical Forecasting Models

Classical forecasting methods are characterized by robust and solid strategies with statistical fundamentals, refined over an extensive period, enabling the development of robust models yielding competitive results [26].

These techniques are characterized by adjusting their hyperparameters based on historical data [27]. These methods maintain competitiveness when integrated in a hybrid mode with contemporary strategies, such as machine learning [18].

2.1.1 SARIMA

AutoRegressive Moving Average (ARMA) models have proven to be highly recognized and practical tools in the field of forecasting, dedicated to the analysis and prediction of time series [28]. Although ARMA focuses mainly on analyzing the autocorrelation and moving average present in the time series, it does not consider other significant factors that may manifest themselves in more complex series [29].

In this context, Seasonal Auto Regressive Integrated Moving Average (SARIMA) emerges as an evolution of ARIMA or an extension of ARMA. SARIMA addresses the limitations of ARMA by incorporating specific trend and seasonality components, thus allowing the generation of forecasts that adapt more efficiently to the behavior of the training data [30]:

$$\mathrm{SARIMA}(p, d, q)(P, D, Q)s. \tag{1}$$

In general terms, SARIMA uses the orders specified in equation 1, where p, d, q represent the orders for autoregressions, differences, and moving averages, respectively, applied to seasonally adjusted data. Likewise, P, D, Q indicate the corresponding orders for autoregressions, differences, and moving averages, but this time applied to seasonal data.

The parameter s represents the size of the seasonal window, a crucial component for capturing temporal patterns. SARIMA fitting is carried out using model selection techniques, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC), as described in detail in later sections of this paper. These strategies provide a solid basis for model optimization, ensuring higher prediction accuracy and adaptability to the inherent complexities of the time series [31].

ISSN 2007-9737

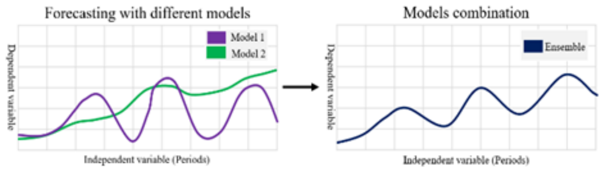

Fig. 5. Ensemble process

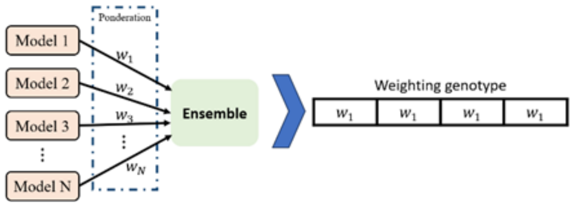

Fig. 6. Ponderation of forecasting models

2.1.2 Exponential Smoothing

Exponential Smoothing (ES) is an old and popular technique based on weighting the last data with the previous forecast; however, it is effective only for one period ahead. Holt & Winter is a valuable ES method, mainly combined with machine learning and ensemble methods [32].

This technique represents an advanced methodology commonly known as Holt&Winters-Triple (H&WT) in time series forecasting [33]. Derived from the simple exponential smoothing proposed by Brown & Holt, H&WT constitutes a sophisticated extension of double exponential smoothing [34].

In contrast to its predecessors, H&WT not only estimates a weighting that controls the influence of the current period in forecasting the next but also incorporates weights for the current trend and seasonality [35].

These weights are reflected in equations 2, 3, and 4, where the smoothing variables α, ,β, γ, which require tuning for optimal adaptation to the behavior of the data, are highlighted.

The synthesis of these components is expressed in equation 5, which represents the expected observation for the next period:

$$A_t = \alpha(Y_t - S_{t-L}) + (1-\alpha)(A_{t-1} + T_{t-1}), \quad (2)$$

$$T_t = \beta(A_t - A_{t-1}) + (1-\beta)T_{t-1}, \quad (3)$$

$$S_t = \gamma(Y_t - A_t) + (1-\gamma)S_{t-L}, \quad (4)$$

$$Y'_{t+p} = A_t + pT_t + S_{t-L+p}. \quad (5)$$

Despite the longevity of H&WT, it remains a highly effective strategy, evidenced by its accuracy in estimating seasonal components [18]. The consistent results it presents reinforce its continued relevance as a reliable tool in the time series forecasting landscape.

2.2 Machine Learning Forecasting Models

Machine learning methods represent artificial intelligence strategies dedicated to creating generalizations and learning patterns in data, which may be imperceptible to the human eye [22].

This achievement is materialized through the exhaustive analysis of data through statistical techniques that enable the identification of thresholds for the issuance of forecasts or decision-making. In this context, we address machine learning strategies specifically designed for time series forecasting in a general way [22].

These strategies focus on the ability to anticipate and model the evolution of data over time, thus improving forecasting accuracy in this context. This approach is essential to address problems related to the temporal dynamics of data and to maximize the usefulness of machine learning applications in the field of time series [32].

2.2.1 Random Forest

Random forest (RF) is one of the most effective decision tree strategies. RF builds an adaptive structure, defining and adjusting itself on a limited data set, this allows the trees to train with different sections of data and decreases the incidence of overfitting. As visualized in Figure 1, this structure exhibits reliability, allowing the adjustability of hyperparameters such as loss function, tree depth, and number of leaves.

However, the RF effectiveness may be bounded by high nonlinearity in the data or when the data are presented in large quantities [36, 37]. In response to these limitations, an effective technique is the integration of several decision trees operating on typically small data sets.

These techniques, known as bagging or boosting, provide superior results compared to individual decision trees [36, 38]. Random Forest Regression is positioned as a bagging technique that generates an abundance of decision trees [39]. These trees are trained with reduced

ISSN 2007-9737

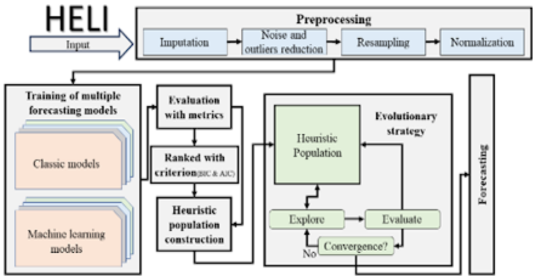

Fig. 7. HELI (Heuristic ensemble for learning information) methodology

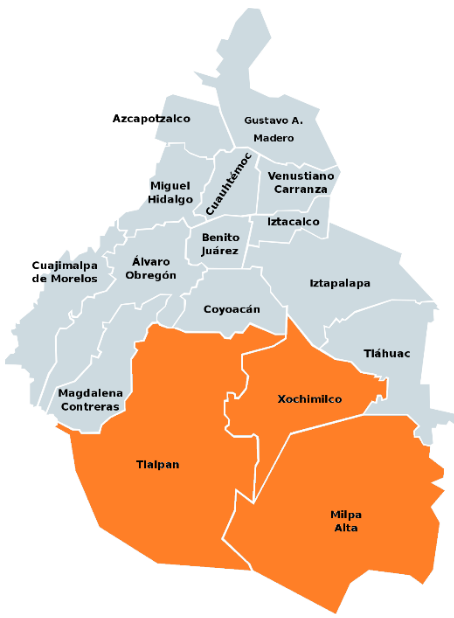

Fig. 8. Data collection area

fragments of information, each emitting its forecast. Then, an ensemble criterion is implemented, following the philosophy of masses. In the regression area, an arithmetic mean is commonly described by $y = \frac{1}{T}\sum_{t=1}^{T} y_t$, where y_t corresponds to the results of each decision tree. This ensemble approach offers notable advantages, such as its high parallelization capability, robustness, and adaptability to a wide variety of problems [40].

2.2.2 Support Vector Regression

SVR constructs a regression function aimed at minimizing error concerning the regression of the training data. The chosen kernel directly influences the efficacy of this function's fit. Additionally, two confidence intervals are established and depicted alongside the proposed regression curve.

These intervals are determined by the hyperparameters C and epsilon, which play a pivotal role in defining the flexibility and width of the tolerance margin [41]. Support Vector Machines (SVM) have demonstrated remarkable efficiency at a modest computational cost [42].

In this context, we delve into its variant suitable for time series: Support Vector Regression (SVR). The efficacy of SVR lies in its capacity to adapt to scenarios characterized by nonlinear behavior, a common occurrence in time series marked by significant noise levels or intricate seasonal patterns [41].

The SVR process is illustrated in Figure 2, wherein the SVM technique endeavors to optimize the fitting of the regression line to the data, accounting for the inherent complexity of the dataset, be it due to its nonlinear nature, presence of noise, or the existence of sophisticated seasonal patterns.

2.2.3 Convolutional Neural Networks

The convolutional neural network (CNN) methodology is a promising approach to the forecasting area, standing out for its ability to perform deep learning based on convolutional and pooling processes [43, 44].

Central to the model, these processes automatically identify relevant features in data sets [44]. This auto-generalization capability enables the model to discover significant patterns and features, which can be used to determine forecasts [8]. Traditionally, image recognition area involving matrix information successfully uses CNN models.

However, in the case of univariate time series, the representation takes a vector form. Figure 3 illustrates the Hankelization process, a technique that, by defining an amplitude value, generates a matrix representation of a time series [45]. The convolutional procedure in CNNs is deployed sequentially through one or more convolution and pooling layers.

The convolutional layers, fundamental in the architecture, apply convolution operations to the input data, enabling local pattern learning. In the specific context of time series forecasting,

ISSN 2007-9737

Algorithm 1. General structure of methodology

```
Algorithm 1 HELI
  Input Time_series
  Output Ensemble_ponderation, Error_metrics_results
procedure HELI
    preprocessing_ts ← preprocessing(time_series)
    train, val, test ← timeSeriesSplit(preprocessing_ts)
    Define models ← all_models_types
    for each model in models do
        model.fit(train, val)
        mse_models[model.index] ← calculateMSE(model)
    end for
    sol_pop ← heuristicPopulation(models, mse_models)
    best_sol ← evolutiveStrategy(sol_pop, models, train, val)
    solution ← evaluateEnsemble(best_sol, test)
    print(solution)
end procedure
```

Algorithm 2. Generation of heuristic population

```
Algorithm 2 Heuristic Population Generation
  Input models, mse_models
  Output Population
procedure heuristicPopulation(models, mse_evaluations)
    population ← Empty
    aic_rnk_w ← weightByPer(aicRnk(models, mse_eval))
    population.append(aic_rank_w)
    bic_rnk_w ← weightByPer(bicRnk(models, mse_eval))
    population.append(bic_rank_w)
    rmse_w ← weightByPer(rmseRnk(models, mse_eval))
    population.append(rmse_w)
    uniform_w ← weightByPer(uniPart(models))
    population.append(uniform_w)
    for each model in models do
        weighting_solution ← zeros()
        weighting_solution[models[model.index]] ← 100
        population.append(weighting_solution)
    end for
    for n in rest_population do
        solution ← randomFactibleSolution()
        population.append(solution)
    end for
    return population
end procedure
```

convolutional layers are employed to discern temporal patterns in the data.

For example, a convolutional layer may specialize in identifying trend patterns, cycles, or seasonality. Pooling layers are incorporated to reduce the dimensionality of the data resulting from the convolutional layers. This process helps to optimize learning efficiency and mitigate the risk of overfitting. In the specific context of time series forecasting, convolutional layers are used to discern temporal patterns in the data.

Thus, a convolutional layer may specialize in identifying trend patterns, cycles, or seasonality. Pooling layers play a crucial role in reducing the amount of data to be processed, which is particularly beneficial for time series with many elements. After the convolutional phase, we apply internal vectorization to connect a multilayer perceptron (MLP) network.

In the forecast context, this network outputs a numerical value representing the forecast for a specific input. This connection process between the convolutional stage and the MLP culminates in generating the desired forecast.

2.2.4 Long Short-term Memory

This approach is based on recurrent neural networks, standing out for its ability to address various problems inherent to this structure [46, 47]. The introduction of cells represents a significant advance, substantially improving the forecasts' quality.

These cells, equipped with gates, play a crucial role in selecting which information to retain or discard in the short, medium, and even long term; this makes it possible to generate quality results in problems related to time series forecasting, enabling the generalization of short-term changes such as those of a seasonal or cyclical nature [48].

In contrast to traditional neural networks, which lack a structure that facilitates the relationship between sequences of previous and subsequent inputs, and recurrent networks that attenuate the influence of the past in large temporal data sets due to their nature, long short-term memory (LSTM) networks represent a functional alternative [8].

This model replaces single neurons with cells, which can be organized in layers and support stacking. This configuration contributes significantly to the learning of long-term dependencies. LSTM structure has four fundamental gates: the output gate, the input gate, the forgetting gate, and the memory gate.

These gates operate through activation functions, exercising precise control over the data flow, as illustrated in Figure 4. This level of control provided by the gates increases the model's predictive capability and confers essential flexibility to adapt to complex patterns and changes in the time series under study.

Algorithm 3. Evolutive process

```
Algorithm 3 Evolutionary Strategy for Ensemble Optimization
  Input Heuristic_population, Trained_models, Train, Val
  Output Best_solution
procedure evolutiveStrategy(H_pop, Trained_models, Train, Val)
    Initialize evolutionary parameters
    best_solution ← Evaluate(H_pop)
    while not converged do
        offspring_pop ← GenOffspring(H_pop)
        offspring_pop ← mutation(offspring_pop)
        for each offspring in offspring_pop do
            ensemble_sol ← CombMdls(offspring, Trained_models)
            ensemble_mse ← Evaluate(ensemble_sol, Train, Val)
            if ensemble_mse < best_solution_mse then
                best_solution ← ensemble_solution
                best_solution_mse ← ensemble_mse
            end if
        end for
        H_pop ← Update(H_pop, offspring_pop)
    end while
    return best_solution
end procedure
```

2.3 Ensemble Methods

Forecasting strategies exhibit diverse behaviors, varying depending on the series processed and even on the specific segment of the series used to fit the model [15, 16]. The absence of a universally superior model stands out since the optimal performance of a model is not guaranteed in all cases or all segments of a series.

This well-recognized problem can be addressed by assembling different forecasting methods, thus generating a single result that incorporates the best individual features of each approach, either through hybridization or ensemble results [39].

Ensemble strategies are based on combining the results of two or more forecasting methods, assigning each of them a percentage of participation [15]. Figure 5 illustrates the essential idea of the assembly process by combining two or more strategies.

However, the simple combination or even the use of algorithm ranking metrics may, in most cases, not be optimal, as it does not match the specific behavior of the algorithms on the regression curves [49]. For this reason, weighting optimization methods emerge as an essential practice, ensuring results are at least as good as the best single method.

These methods perform an exhaustive exploration of the solution space, adjusting to the data and the algorithms participating in the ensemble. In this way, a precise and efficient adaptation to the inherent complexities of the time series is guaranteed, strengthening the robustness and accuracy of the forecasting process.

2.3.1 Evolutionary Weight Optimization

The weighting adjustment in an ensemble method constitutes a combinatorial optimization challenge, which may be infeasible to solve exhaustively. Therefore, the heuristic strategy obtains results in a reasonable time, and in this sense community scientists can use it to obtain fast results.

Evolutionary algorithms find their inspiration in biological evolution, where different individuals intermingle or mutate over generations, acquiring or improving skills necessary for their survival and discarding those not necessary or counterproductive [50].

In this strategy, each weighting or solution is treated as an individual, coding it to obtain its genotype and evaluating its performance to determine its fitness value, as shown in Figure 6. The evolutionary weighting strategy is derived from Holland's approach, which uses a population consisting of a set of solutions that will undergo an iterative evolutionary mechanism similar to biological generations [51].

This process selects parents by evaluating their fitness, and then a combination strategy of the parents produces offspring solutions. The replacement process applies a criterion based on the evaluation of the fitness to integrate into the next generation.

Between the process of generating new solutions and their evaluation for integration into the population, there is the possibility of modifying these new solutions by perturbing some genes of the solution; this strategy is known as mutation and allows greater exploration of the solution space, also decreasing the stagnation of the population and facilitating the rapid attainment of the convergence criterion.

The evolutionary strategy will evolve over a fixed number of generations or may end earlier or be extended, depending on the definition of different convergence criteria. These criteria evaluate the performance of the best solutions for

2.4 Rank Methods

In machine learning, prediction models are nurtured by strategies that seek to minimize a specific error metric. This metric commonly contrasts the predicted value with the actual value in a given period, thus accumulating errors throughout the training process. Several error metrics are available, each focusing on different aspects of the dispersion for the original value.

However, these metrics only guarantee the convergence of the forecast models to an optimal fit for the training and validation set. An inherent danger in a soon convergence is that it can lead to model overfitting, especially in strategies involving many parameters and hyperparameters [35].

On the other hand, ranking methods take a different approach when assessing the quality of a model. These methods integrate additional features along with the error metric to assign a score that more holistically reflects the effectiveness of the model.

This approach allows a balance to be stuck between model accuracy, computational complexity, run times, and other relevant factors. In essence, ranking methods offer a more comprehensive perspective for evaluating and selecting models, overcoming the limitations inherent in pure convergence to the training set fit.

2.4.1 Akaike Information Criterion

The Akaike information criterion (AIC) is a fundamental statistical tool for evaluating forecasting methods [52]. Its purpose is to strike a balance between model accuracy, as measured by a likelihood function, and model complexity, which is mainly determined by the number of hyperparameters fitted to perform the forecast [53].

The value of the AIC decreases as the models achieves better results. This behavior is reflected in equation 6, where k represents the number of hyperparameters of the model, and L denotes the likelihood function. In this context, the likelihood the population in each generation and determine whether the strategy has converged prematurely or requires a more extensive exploration process function is constructed from the Mean Squared Error (MSE) of the model evaluated under the assumption of the normality of the errors:

$$\text{AIC} = 2k - 2\ln(L). \tag{6}$$

2.4.2 Bayesian Information Criterion

The Bayesian information criterion (BIC) is an essential statistical tool for evaluating and comparing various forecasting methods in the forecasting field [54]. This approach is based on the search for an optimal balance between the forecasting accuracy of the model, evaluated through the likelihood function, and the inherent complexity of the model. A distinctive aspect of the BIC is its ability to penalize those models that benefit from short sample sizes, contributing to a more robust and generalizable evaluation [25]:

$$\text{BIC} = n \cdot \ln(L) + k \cdot \ln(n). \tag{7}$$

In equation 7, n represents the size of the data set, L the likelihood function, and k the number of hyperparameters in the model. Unlike AIC, BIC multiplies the number of parameters by the logarithm of the sample size, a weighting that accentuates the penalty as the complexity of the model increases.

3 Proposed Methodology

In this section, we present the design and configurations of HELI (Heuristic Ensemble for Learning Information), a highly flexible and robust methodology for time series forecasting in the context of climate change. The methodology is divided into three fundamental sections: preprocessing, training individual forecast models, and adjusting the participation weighting of each method to form an ensemble.

These phases are shown in detail in Figure 7. Each of these sections houses methods whose configurations are detailed below. These methods are modularly configurable. Thus, removing, extending, or replacing elements with other strategies that perform similar functions is possible.

The methodological structure is detailed in Algorithm 1, which represents the general framework of the methodology. Its input is a time series (with the variable time_series), and its output consists of the optimal weighting obtained together with the evaluation of this weighting using various error metrics.

ISSN 2007-9737

Table 1. Hyperparameter for forecasting models

Hyperparameter	Value
Random Forest	
Criterion	Friedman mse
Estimators	23000
Maximum features	Sqrt
Bootstrap	True
Maximum samples	0.8
Minimum samples leaf	10
Minimum samples split	8
SVR	
Kernel	Poly
Degree, Epsilon, C	5,0.01,5
Gamma	None
Kernel Ridge	
Kernel	Poly
Alpha, Gamma	1, 0.1
Degree, Coef0	4, 0.1
CNN	
Convolutional Layers	**2 Conv2D**
Kernel	**(15,15) & (5,5)**
Filters	**64**
Strides	**1**
Pooling	**Average Pooling**
Dropout	**0.25**
Activation	ReLU
Pooling	AverangePooling
Optimizer	Adam
Learning rate	0.01
Epochs	5000
Batch size	10
LSTM	
LTSM Layers	**1**
LSTM Cells	**35**
MLP Layers	**4**
MLP Neurons	**200**
MLP Dropout	**0.3**
MLP Activation	Tanh
Output Activation	Linear
Optimizer	Adam
Learning rate, Epochs	0.005,500
Batch size	10
Early stopping	40

Line 2 describes the data cleaning and processing process (where the function preprocessing cleans the data and is assigned with the variable preprocessing_ts); then, the time series is divided into the three blocks shown in line 3. From lines 4 to 8, individual forecast models are defined, trained, and evaluated using the mean squared error (MSE) metric. This evaluation results in the creation of a heuristic population mentioned in line 9 and detailed in Algorithm 2.

Next, an evolutionary strategy, presented in line 10 and explained in Algorithm 3, explores and exploits the population to find a solution at least as good as the best individual model. This solution reflects the contribution of the individual models to the ensemble, and its evaluation to obtain the resulting error values is indicated in line 11.

The generation of the heuristic population, described in Algorithm 2, takes as input the forecast models together with their corresponding MSE evaluation and returns an initial population of solutions. This population starts empty in line 1, and solutions are generated and subsequently added to it. Delimiting lines 3 to 6, the models are ranked using the BIC and AIC criteria, which rank and score the suitability of the models.

In this context, the models will receive a proportional representation in percentage of the score obtained by the ranking methods, and the generated solutions will be incorporated into the population. For lines 7 and 8, RMSE is used, which is not a ranking method but is used to evaluate the error.

The results of this evaluation make it possible to rank the algorithms from least to most error, generating a percentage inversely proportional to their error and adding the solution to the population. Lines 9 and 10 describe the integration of a uniform solution in which all forecasting models have an equivalent participation in the ensemble.

Lines 11 to 15 detail the integration of solutions where each forecast model receives 100% participation in the ensemble, leaving the others with a participation of 0. This strategy ensures that the results obtained are at least as good as the best individual model.

A random process generates the remaining solutions (lines 16 to 19). These solutions must be feasible in terms of the proportionality of their participation in the ensemble. Algorithm 3 describes an evolutionary strategy to optimize the

Table 2. Description of the best configuration found for the evolutionary strategy

Configuration	Value
Number of Generations	Uncertain
Population Size	500 individuals
Genotype Size	7 genes
Selection Technique	Weighted Roulette
Selected Parents	200 parents
Crossover Type	Scattered
Mutation Probability	Adaptative (0.2 – 0.9)
Mutation Type	Double Random
Allow Duplicates	No
Elitism	50 individuals
Stopping Criterion	Rate of change over 20 generations
Fitness Objective	Minimize MSE

weights in the ensemble participation. This algorithm receives the heuristic population, the previously adjusted models, and the validation and training set, returning the best solution found.

In line 2, an evaluation of the population is performed to temporarily store the best solution. From line 4 to line 16, an evolutionary process is run to an evolutionary process is executed that ends when the convergence criterion is met. This process involves the generation of offspring and their mutation, explained in lines 5 and 6. The generated offspring are evaluated using the same metrics and criteria as the initial population, as described in line 9.

When a better solution than the previously stored one is found, it is updated, as shown in lines 10 to 13. Some elements of the population are updated with the generated offspring, as described in line 15. Finally, the algorithm returns the best solution found during the evolutionary process.

The inherent flexibility of HELI translates into the ability to work with various strategies or the incorporation of new ones; however, it is essential to note that such adaptability entails further adjustment and optimization. Nevertheless, the results may vary significantly depending on these factors and the nature of the test data used.

3.1 Preprocessing

Given the inherent complexity of the nature of the data, as will be discussed in detail in later sections, a preprocessing of the information is required. This process becomes crucial to ensure the data's integrity and completeness, scale them appropriately, and mitigate the presence of noise and outliers.

Although the methods described find applicability in various time series, it is crucial to recognize that the effectiveness of each strategy may vary according to the specific characteristics of the data. Therefore, the selection of methods must be customized for each data set to be evaluated, adjusting optimally to the particularities of each time series.

An initial review revealed the existence of missing data in the series. However, these were presented in isolation and not consecutively. Opting for data trimming could distort the width of the seasonal spaces, biasing the learning process in specific sections of the series. Therefore, we decided to impute the missing data through quadratic interpolation, thus ensuring the completeness of the series without compromising its temporal structure.

In later stages, we describe two strategies for noise reduction and outlier containment are implemented. Outlier reduction in data sets that may include trend and/or seasonal values represents a challenge since robust techniques applied to other areas such as IQR that base their behavior on median dispersion are not sensitive to seasonal behaviors, so important information may be considered as noise [13].

The 3-sigma technique made it possible to limit the series by eliminating outlier observations. In consideration of the seasonal nature of the series, subject to fluctuations associated with climate change, the 3-sigma strategy uses a moving window whose size represents a seasonal period, allowing relevant information within the seasonal periods to be retained and noise to be suppressed.

The singular Value Decomposition (SVD) process was then applied to the data series. This technique interprets the low-influence components as noise and then allows its smoothing by eliminating them. Subsequently, a singular value decomposition (SVD) process was carried out on

ISSN 2007-9737

Table 3. Statistical information of the data used in the experimentation

Data filled and re-sampled					
	Count	**Min**	**Max**	**Media**	**Std Dev.**
Min. Temp	3527	-2.6°	14.83°	8.74°	2.6546
Max. Temp	3527	7.52°	32.43°	22.16°	2.6836

Table 4. Average results obtained from the individual tuned forecast models for validation set

Evaluation Models with sMape		
Models	**Min Temp**	**Max Temp**
SARIMA	25.42%	37.40%
Holt & Winters	24.8%	24.45%
CNN	9.63%	7.24%
LSTM	11.02%	6.93%
Random Forest	11.74%	6.87%
SVR	10.23%	6.66%
Kernel Ridge	19.96%	8.41%

the data series. This technique enables smoothing by eliminating low-influence components that are interpreted as noise.

To address the seasonal nature of the problem, defining the amplitude of the Hankelization matrix as a function of a seasonal period is necessary to preserve 90% of the explained variance components. Climate change studies usually focus on medium and long-term changes, since short-term observations can be affected by other natural or anthropogenic phenomena, leading to high variability [3].

Therefore, weekly or monthly observations provide a better generalization of climate behavior. We resampled the original data collected from daily observations as weekly observations. This process applies an arithmetic mean every seven observations to obtain a weekly data series that facilitates the analysis in a more effective climate forecasting context.

3.2 Training Models

The previous section gave an overview of the forecasting methods used in the methodology addressed. The choice of the hyperparameters of the forecasting methods was made through fast tuning using the grid search strategy.

This strategy combines the hyperparameters from a previously defined list, however, to guarantee an optimal performance it is suggested the use of other tuning techniques such as evolutionary algorithms or Bayesian strategies, however, this may significantly impact the model fitting times.

In the same way, the same strategy was used for the tuning of the neural network architecture whose parameters are listed in Table 1.

Table 1, shows the hyperparameters used in each strategy, which were tuned and optimized through genetic algorithms or mesh searches, depending on the intrinsic complexity of each specific algorithm. This process may be performed for each new set of time series, thus ensuring an optimal adaptation to the particular characteristics of each set of information.

Additionally, since the forecasting methods execute and format the data independently, parallel execution was chosen whenever possible, thus optimizing computation time on the processor or GPU. However, this type of execution demands a more considerable amount of working memory. It is important to note that the type of execution, whether parallel, sequential, or mixed, does not affect the results obtained. After training the methods, the Mean Squared Error (MSE) value is obtained for the validation set.

Since this set is invisible in the training process, it provides a more realistic approximation for the models when the entire set of cases is unknown. Next, the construction of the heuristic population begins. We incorporate a weighting process, where the initial set of weights can be determined with the following strategies: a) randomly generated using a uniform distribution, b) an error metric such as RMSE, and c) using ranking metrics, such as AIC or BIC.

Now we describe the ranking generated by BIC and AIC, along with an evaluation of the different models using these metrics. This ranking punctuates the suitability of the models, although they do not explicitly give the percentage of participation of each one.

Taking AIC as an example, any individual weight is given by equation 8, where AIC_i

Table 5. Average results for different ensembles for validation set

Ponderations obtained with sMape		
Ponderations	**Min Temp**	**Max Temp**
Uniform	19.45%	14.12%
AIC	11.69%	8.45%
BIC	11.52%	6.23%
HELI	9.12%	5.87%

represents the ranking value of each model, N the number of models, and w_i the resulting weight. The weighting process performed in BIC uses a similar equation. The weights process is not necessarily the best, but they give us an initial set of weights taken to explore their neighborhood and evolve toward an optimal set of weights:

$$w_i = \frac{1}{AIC_i \cdot \sum_{j=1}^{N} \frac{1}{AIC_j}}. \quad (8)$$

The initial weights determined for AIC or BIC are taken as solutions in the Ponderation Optimization process described in the next section.

3.3 Weighting Optimization

The previous section pointed out that the initial weights obtained by RMSE, AIC, BIC, or a uniform distribution are not necessarily the most adequate. The ranking or selection methods only qualify the most capable strategies but do not determine the degree of participation that each one should have when integrated into the ensemble.

Therefore, it is necessary to adjust their degree of participation to obtain the best possible combination of algorithms. The population in this problem consists of 500 solutions, which are generated by a heuristic strategy, while the rest of the solutions are randomly generated. The order generated by the ranking methods and their obtained values are converted to participation percentages to obtain the heuristic solutions, on the other hand, a solution with uniform weighting is generated, i.e. in which all the algorithms have the same participation in the ensemble.

To ensure that the strategy allows obtaining a solution at least as good as the best individual forecasting method, solutions are also generated where 100% of the ensemble belongs to each of the methods. The parameters of the evolutionary algorithm are detailed in Table 2 likewise, a sub-condition is implemented that is satisfied when the rate of change of the best solutions during ten generations tends to zero.

In this case, a population perturbation is performed on a random amount, logically discarding the elite solutions; this allows us to avoid premature stagnation and explore unexplored sections of the solution space. When the convergence criterion is met, the algorithm returns the best solution found, which will constitute the best weighting of the forecast models and form the ensemble's weights.

The convergence criterion indicates that the algorithm reaches its convergence when the variability of the best solution over 20 generations tends to zero. The value was chosen based on the results of a preliminary test. In this test we observed that improvements of the best solutions occurred at most every 10 or 12 iterations before convergence. That is, when the algorithm stops obtaining improvements.

4 Experimentations and Results

This section describes the error metrics used to tune the strategies and generate the results report internally. The specific instances used in the experimental process are presented, along with a complete exposition of all the results obtained.

4.1 Experimental Conditions

The proposed methodology and strategies used in the comparison were developed and executed using the Jupyter Notebook interactive environment in Python. Experiments included parallel executions to increase computational efficiency, resulting in significantly faster training time. Specifically, the deep learning neural networks were executed using GPU computing capabilities to minimize learning time and increase their performance.

The equipment used to carry out these tasks has a Ryzen 5700x processor with 32GB of RAM and a Nvidia 4060 graphics processing unit. In all cases, the executions were carried out dynamically

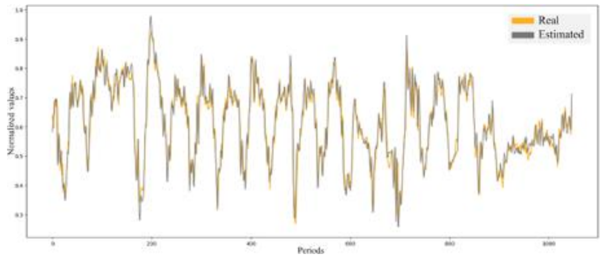

Fig. 10. Trend component from 2018 to 2028

with access to hardware resources, with no time constraints imposed. An analysis based on average results derived from at least 30 runs of each strategy is presented to ensure consistency in the predictions and avoid results biased by randomness.

This rigorous replication is applied to the proposed strategies and those from the state of the art, providing a more robust and reliable view of the effectiveness of each approach. Additionally, to ensure the optimality of the results, all models were tuned using specific algorithms.

This tuning phase was carried out comprehensively, adjusting the key parameters of each model to ensure optimal performance in terms of accuracy and generalization. This meticulous approach reinforces the validity and reliability of the results presented in the framework of this comparative analysis.

4.2 Error Metrics

Two different types of error metrics are used in the proposed methodology: The Mean Squared Error (MSE) and the Symmetric Mean Absolute Percentage Error (sMAPE). The MSE is used to refine the accuracy of the model and adjust its hyperparameters, while the sMAPE is used exclusively to report model performance.

The MSE is used as an effective tool during the fitting phase, as it amplifies significant errors, thus contributing to the learning process to minimize this error metric. Its formula is expressed in equation 9, where y_i represents the actual observation and $\hat{y}_i$ represents the prediction generated by the model [35]:

$$MSE = \frac{1}{N} \cdot \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{9}$$

On the other hand, we try to present the results understandably. Thus, we show the sMAPE error in percentage terms, allowing a simple comparison with other methodologies. Its formula is expressed in equation 10, where y_i represents the actual observation and $\hat{y}_i$ represents the prediction generated by the model [35]:

$$sMAPE = \frac{1}{100} \cdot \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{\frac{|y_i| + |\hat{y}_i|}{2}}. \tag{10}$$

The last two metrics provide a comprehensive assessment of the model performance, allowing a detailed understanding of internally adjusted accuracy and externally reported performance.

4.3 Data Descriptions

The time series used in this study were derived from meticulously collected observations from meteorological stations, which are part of the Mexican National Meteorological Service (SMN) database [55]. The sample includes data from February 1, 1950, to August 31, 2017, with an hourly observation period.

This reliable source, available for free download at the SMN's official website, is a fundamental resource for analyzing climatic phenomena. The time series, specifically for Maximum and Minimum Temperature, were structured in Celsius degrees, as detailed in Table 3, and reflect carefully recorded observations in the southern region of Mexico City. The data are public, and can be found in the Kaggle repository website [56].

The choice of this geographical area was not arbitrary; on the contrary, it is based on carefully selected criteria. This locality encompasses the areas of Milpa Alta, Xochimilco, and Tlahuac, regions with specific climatic relevance that are observable in Figure 8. The vulnerability of the Valley of Mexico to climate change is evidenced by the significant population density of this region, the most populated in the country.

The marked anthropogenic influence derived from industrialization and the consequent generation of pollutants makes this area a crucial enclave for understanding the effects of climate change in an urban context [57]. Additionally, the choice of the southern zone of Mexico City responds to its particular orographic condition. This

Table 6. Average results were obtained from HELI and compared to the best state-of-the-art strategy for test set

Evaluation Methodologies with sMape		
Methods	Min Temp	Max Temp
SMYL	8.78%	5.13%
HELI	9.54%	6.02%

Table 7. Wilcoxon test shows Heli and SMYL are equivalent

Statistical results	
∞	0.05
Z-statistic	2.126
p-value	0.0554

geographical feature mitigates the heat island effect, a common phenomenon in urbanized areas.

The exclusion of this effect allows a more accurate analysis of climate variations and the impacts of climate change in the region. The methodology included to subdividing the data into three distinct blocks: the training set, validation, and test, allocating 60% for training and 20% for each remaining set, respectively.

The training block played a key role in the process, as it was used to train each forecast model and adjust the ensemble weights. This methodological approach contributes to a comprehensive and robust evaluation of forecast models in forecasting.

4.4 Results Obtained

The first experimental phase focuses on the adjustment and optimization of the individual forecasting methods. Table 4 presents the results obtained for each of them in both instances, expressed in terms of the scaled mean absolute mean percentage error (sMAPE) for the validation set, where an ensemble with which the models are not trained is evaluated.

The effectiveness of forecasting methods depends exclusively on the forecasting model, its hyperparameter configuration and the data. In this case CNN and SVR present the best results in general, however for other instances it is not guaranteed that they have similar effectiveness, the “No free lunch theorem” suggests that there are no universally superior methods for all problems raised in machine learning [14].

Machine learning models outperform classical forecasting models, attributable to the complexity of the data and the presence of natural temperature changes treated as noise. CNN, LSTM, and Random Forest generally exhibit superior performance.

However, SVR and Kernel Ridge show good results with fewer adjustable parameters, which could mitigate the risk of overfitting. Besides, the weights of metrics are adjusted by a heuristic population that evolves to find an optimal combination, thus issuing quality forecasts. Table 4 shows that classical methods have a more limited performance than machine learning methods due to the better generalization of the latter for time series complexity.

Neural and Random Forest strategies stand out, although they require more hyperparameter tuning and training time. In Table 5 includes the performance of models with ensemble solutions. The uniform ensemble worsens most of the individual models since it does not consider the differences in the generalization of the curves.

Although the results improve with metrics such as BIC and AIC, they are not guaranteed to outperform the best individual method. The optimized forecast outperforms the individual methods and the evaluated ensemble strategies thanks to an initial population that includes other ensemble weights.

In addition, the proposed strategy is compared with state-of-the-art ensemble approaches in Table 6, such as SMYL. The latter was built from the winning paper of the M4 prognostics competition, with similar modifications to the preprocessing of the proposed strategy and hyperparameter tuning by genetic algorithm. Although SMYL shows a slight superiority in the tests shown in Table 7, a Wilcoxon test reveals no significant differences, indicating equivalence between both strategies.

In terms of time, the average time for HELI is barely 20 minutes per execution, while SMYL takes 1 minute if it already knows the instance and has a semi-empty pool, and 25 minutes if it does not.

This value can be misleading since the Smyl strategy is designed to have a pool of solutions, so

ISSN 2007-9737

Fig. 9. HELI results compared to real data

Table 8. Estimated values of the proposed model compared to the actual values from 2018 onwards

Forecasting results				
	Minimum Temp		**Maximum Temp**	
Year	**Real**	**Estimated**	**Real**	**Estimated**
2018	10.2°	10.5°	23.5°	22.7°
2019	10°	10.6°	23.3°	22.9°
2020	10.1°	10.4°	23.2°	22.7°
2021	9.8°	10.3°	23°	22.8°
2022	9.9°	10.4°	23.1°	22.8°
2023	10°	10.2°	23.2°	23°

Table 9. Results for second experiment

sMAPE forecasting evaluation		
Variables	**Monterrey**	**Guadalajara**
Max Temp	10.8%	9.5%
Min Temp	12.7%	11.5%
CO_2	6.4%	5.6%
NO_2	4.3%	6.4%
PM_{10}	7.2%	5.1%

the more knowledge it has, the longer it will take to execute [19].

4.5 Interpretation of the Forecasts Obtained

According to the regression curve obtained, the actual results can be contrasted with those obtained through the proposed forecasting method. This comparison is illustrated in Figure 9, where the inherent complexity of the forecasting task is evident, as well as the ability of the ensemble approach to integrate strategies, effectively capturing trend, seasonality, and even random noise patterns.

4.6 Results Discussion

This study focuses on presenting the results using the sMAPE metric; however, it is crucial to highlight that other metrics allow for discerning the effectiveness of the proposed strategy compared with the leading method in state-of-the-art.

In particular, the importance of considering training time in this evaluation is emphasized. Smyl was executed under the same experimental conditions as HELI; however, the architecture proposed by SMYL does not implement parallel structures, so its execution is sequential.

The proposed strategy, HELI, and SMYL exhibit similar training times in the analyzed instances. However, it is essential to note that SMYL accumulates knowledge as it is tested with various time series.

In this context, the raw learning time taken by Smyl is considerably larger to achieve its results. In contrast, the HELI method presents a constant adjustment time, depending only on the individual complexity of the series and its size.

From this perspective, HELI provides equivalent results in a lower adjustment time; thus, it offers a practical and time-efficient alternative. The forecasting tasks were not limited to the initial test set.

4.7 Prediction Over New Horizons

We performed an additional test to evaluate the ability of HELI to predict a longer period, by extending the forecast beyond the limits of the actual time series data until 2028. For this experiment we expanded the forecast period in 521 weeks in the future, which is equivalent to 10 years ahead. In other words, we projected the prediction of HELI since 2018 to 2028. Although the data set is limited to 2018, at the time of writing this paper is possible to contrast the results up to 2023. In Table 8, we can observe a clearly convergence between the estimated annual averages and the actual annual values. These results reveal the effectiveness of HELI.

In addition, Figure 10 shows the estimated evolution of the trend component, which presents a slight increase in the gap between maximum and minimum temperatures as we move into the future. This finding suggests that long-term weather patterns in southern Mexico City are undergoing changes, which could indicate an influence of phenomena associated with climate change.

4.8 Evaluation for Other Climate Change Instances

To evaluate the effectiveness of the method, we analyzed other instances of climate change-related variables, this time for the Mexican cities of Guadalajara and Monterrey. We used public data collected from airport stations.

Table 9, shows the results obtained for the variables of maximum temperature, minimum temperature, the gases CO_2, and NO_2 and the PM_{10} particles. We performed this experiment with 4012 observations, and we divided the data in the same way as in the first experiment. In the second experiment, we adjusted the hyperparameters of the forecasting algorithms.

The sMAPE results show that the accuracy is maintained, with error values ranging since 4.3 to 12.7% for Monterrey and 5.1 to 11.5% for Guadalajara. Notice that in this second experiment, Heli obtained the best forecast for the two gases, followed by PM10 particles. Besides, the forecast for the two temperature variables is reasonably acceptable.

5 Conclusions and Discussions

Climate change represents one of the most decisive challenges for the present and future of humanity. Its profound environmental, economic, and social implications demand the urgent development of effective methods for temperature forecasting. In this context, this paper presented HELI, a novel methodology for temperature forecasting that integrates robust preprocessing, training of several individual methods, and optimization of their weights through evolutionary algorithms. Experiments performed on extensive temperature time series in the southern area of Mexico City show promising results.

HELI achieves a sMAPE error of 9.54% for minimum and 6.02% for maximum temperature, outperforming both state-of-the-art individual methods and sophisticated ensemble strategies previously in the state of the art. Rigorous statistical tests demonstrate HELI's equivalence to SMYL, the leading global method. HELI's competitiveness is even more valuable when considering its flexibility and scalability.

Unlike other approaches, HELI does not accumulate knowledge between time series but performs a specific adjustment for each one. This translates into bounded and constant training times, dependent only on the individual complexity of each series. On the other hand, HELI's modularity allows easy incorporation of new methods and configurations. The solid results obtained position HELI as a highly effective alternative for temperature prediction.

This work lays the groundwork for further exploration of integrations with new individual methods, improved preprocessing, and more comprehensive hyperparameter optimizations. It remains a pending task to extend the evaluation of HELI to other regions and types of time series related to climate phenomena. In conclusion, HELI represents a significant advance in the crucial and urgent field of climate forecasting.

Both its methodological design and experimental contributions are concrete contributions to develop increasingly effective solutions to the complex and urgent challenges posed by climate change.

HELI's potential for flexibility, enabling the assessment of multiple individual variables; scalability, allowing the incorporation of other forecasting variables, and competitiveness lay a solid foundation for its potential adoption in climate studies. We hope this work will stimulate new lines of research in this priority area for the planet's sustainability.

Acknowledgments

The authors would like to acknowledge with appreciation and gratitude to CONAHCYT, TecNM, and Laboratorio Nacional de Tecnologías de la Información(LaNTI) in the Instituto Tecnológico de Ciudad Madero for the access to the cluster. Erick Estrada Patiño would like to thank CONAHCYT for the Ph.D. Scholarship.

References

1. **Raffel, T. R., Romansic, J. M., Halstead, N. T., McMahon, T. A., Venesky, M. D., Rohr, J. R. (2012).** Disease and thermal acclimation in a more variable and unpredictable climate. Nature Climate Change, Vol. 3, No. 2, pp. 146–151. DOI: 10.1038/nclimate1659.
2. **Moss, R. H., Edmonds, J. A., Hibbard, K. A., Manning, M. R., Rose, S. K., van-Vuuren, D. P., Carter, T. R., Emori, S., Kainuma, M., Kram, T., Meehl, G. A., Mitchell, J. F. B., Nakicenovic, N., Riahi, K., Smith, S. J., Stouffer, R. J., Thomson, A. M., Weyant, J. P., Wilbanks, T. J. (2010).** The next generation of scenarios for climate change research and assessment. Nature, Vol. 463, No. 7282, pp. 747–756. DOI: 10.1038/nature08823.
3. **Masini, R. P., Medeiros, M. C., Mendes, E. F. (2021).** Machine learning advances for time series forecasting. Journal of Economic Surveys, Vol. 37, No. 1, pp. 76–111. DOI: 10.1111/joes.12429.
4. **Al-Ghussain, L. (2018).** Global warming: Review on driving forces and mitigation. Environmental Progress and Sustainable Energy, Vol. 38, No. 1, pp. 13–21. DOI: 10.1002/ep.13041.
5. **Hardy, J. T. (2003).** Climate change: Causes, effects, and solutions. John Wiley and Sons.
6. **Houghton, J. T., Jenkins, G. J., Ephraums, J. J. (1990).** Climate change: The IPCC scientific assessment. American Scientist (United States), Vol. 80, No. 6.
7. **Peters-Lidard, C. D., Rose, K. C., Kiang, J. E., Strobel, M. L., Anderson, M. L., Byrd, A. R., Kolian, M. J., Brekke, L. D., Arndt, D. S. (2021).** Indicators of climate change impacts on the water cycle and water management. Climatic Change, Vol. 165, No. 1–2. DOI: 10.1007/s10584-021-03057-5.
8. **Hou, J., Wang, Y., Zhou, J., Tian, Q. (2022).** Prediction of hourly air temperature based on CNN–LSTM. Geomatics, Natural Hazards and Risk, Vol. 13, No. 1, pp. 1962–1986. DOI: 10.1080/19475705.2022.2102942.
9. **Ray, S., Das, S. S., Mishra, P., Al-Khatib, A. M. G. (2021).** Time series SARIMA modelling and forecasting of monthly rainfall and temperature in the south Asian countries. Earth Systems and Environment, Vol. 5, No. 3, pp. 531–546. DOI: 10.1007/s41748-021-00205-w.
10. **Pang, B., Yue, J., Zhao, G., Xu, Z. (2017).** Statistical downscaling of temperature with the random forest model. Advances in Meteorology, Vol. 2017, pp. 1–11. DOI: 10.1155/2017/7265178.
11. **Papacharalampous, G., Tyralis, H., Koutsoyiannis, D. (2018).** Predictability of monthly temperature and precipitation using automatic time series forecasting methods. Acta Geophysica, Vol. 66, No. 4, pp. 807–831. DOI: 10.1007/s11600-018-0120-7.
12. **Radhika, Y., Shashi, M. (2009).** Atmospheric temperature prediction using support vector machines. International Journal of Computer Theory and Engineering, Vol. 1, No. 1, pp. 55–58. DOI: 10.7763/ijcte.2009.v1.9.
13. **Wolpert, D., Macready, W. (1997).** No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, pp. 67–82. DOI: 10.1109/4235.585893.
14. **Dietterich, T. G. (2000).** Ensemble methods in machine learning. International Workshop on Multiple Classifier Systems, pp. 1–15. DOI: 10.1007/3-540-45014-9_1.
15. **Frausto-Solis, J., Rodriguez-Moya, L., González-Barbosa, J., Castilla-Valdez, G., Ponce-Flores, M. (2022).** FCTA: A forecasting combined methodology with a threshold accepting approach. Mathematical Problems in Engineering, Vol. 2022, No. 1, pp. 1–19. DOI: 10.1155/2022/6206037.
16. **Yan, G., Yu, C., Bai, Y. (2021).** A new hybrid ensemble deep learning model for train axle temperature short term forecasting. Machines, Vol. 9, No. 12, pp. 312. DOI: 10.3390/machines9120312.
17. **Makridakis, S., Spiliotis, E., Assimakopoulos, V. (2018).** The M4 competition: results, findings, conclusion and way forward. International Journal of

Forecasting, Vol. 34, No. 4, pp. 802–808. DOI: 10.1016/j.ijforecast.2018.06.001.

18. **Smyl, S. (2020).** A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. International Journal of Forecasting, Vol. 36, No. 1, pp. 75–85. DOI: 10.1016/j.ijforecast.2019.03.017.
19. **Gelper, S., Fried, R., Croux, C. (2009).** Robust forecasting with exponential and holt–winters smoothing. Journal of Forecasting, Vol. 29, No. 3, pp. 285–300. DOI: 10.1002/for.1125.
20. **Alpaydin, E. (2020).** Introduction to machine learning, fourth edition. MIT Press.
21. **Ahmed, N. K., Atiya, A. F., Gayar, N. E., El-Shishiny, H.** (**2010).** An empirical comparison of machine learning models for time series forecasting. Econometric Reviews, Vol. 29, No. 5–6, pp. 594–621. DOI: 10.1080/07474938.2010.481556.
22. **Engelhardt, I., De-Aguinaga, J., Mikat, H., Schüth, C., Liedl, R. (2013).** Complexity vs. simplicity: groundwater model ranking using information criteria. Groundwater, Vol. 52, No. 4, pp. 573–583. DOI: 10.1111/gwat.12080.
23. **Chakrabarti, A., Ghosh, J. K. (2011).** AIC, BIC and recent advances in model selection. Philosophy of Statistics, Vol. 7, pp. 583–605. DOI: 10.1016/b978-0-444-51862-0.50018-6.
24. **Kuha, J. (2004).** AIC and BIC: comparisons of assumptions and performance. Sociological Methods and Research, Vol. 33, No. 2, pp. 188–229. DOI: 10.1177/0049124103262065.
25. **Faloutsos, C., Gasthaus, J., Januschowski, T., Wang, Y. (2018).** Forecasting big time series: old and new. Proceedings of the VLDB Endowment, Vol. 11, No. 12, pp. 2102–2105. DOI: 10.14778/3229863.3229878.
26. **Peter, G., Matskevichus, M. (2019).** Hyperparameters tuning for machine learning models for time series forecasting. 6th International Conference on Social Networks Analysis, Management and Security, Vol. 6, pp. 328–332. DOI: 10.1109/snams.2019.8931860.
27. **Makridakis, S., Hibon, M. (1997).** ARMA models and the box-Jenkins methodology. Journal of Forecasting, Vol. 16, No. 3, pp. 147–63. DOI: 10.1002/(sici)1099-131x(199705)16;3<147::aid-for652>3.0.co;2-x.
28. **Bustos, O. H., Yohai, V. J. (1986).** Robust estimates for ARMA models. Journal of the American Statistical Association, Vol. 81, No. 393, pp. 155–168. DOI: 10.1080/01621459.1986.10478253.
29. **Fang, T., Lahdelma, R. (2016).** Evaluation of a multiple linear regression model and SARIMA model in forecasting heat demand for district heating system. Applied Energy, Vol. 179, pp. 544–552. DOI: 10.1016/j.apenergy.2016.06.133.
30. **Hyndman, R. J., Athanasopoulos, G. (2018).** Forecasting: Principles and practice. OTexts.
31. **Chatfield, C. (1978).** The holt-winters forecasting procedure. Applied Statistics, Vol. 27, No. 3, pp. 264–279. DOI: 10.2307/2347162.
32. **Indriani, R., Sugandha, A., Tripena, A., Larasati, N., Rokhman, A. F., Bon, A. T. (2020).** Forecasting of air temperature in Cilacap Regency with triple exponential smoothing (holt-winter) method. 5th NA International Conference on Industrial Engineering and Operations Management.
33. **Ali, J., Khan, R., Ahmad, N., Maqsood, I. (2012).** Random forests and decision trees. International Journal of Computer Science Issues, Vol. 9, No. 5, p. 272
34. **Breiman, L. (2001).** Random forests. Machine learning, Springer, Vol. 45, No. 1, pp. 5–32. DOI: 10.1023/a:1010933404324.
35. **Esmaily, H., Tayefi, M., Doosti, H., Ghayour-Mobarhan, M., Nezami, H., Amirabadizadeh, A. (2018).** A comparison between decision tree and random forest in determining the risk factors associated with type 2 diabetes. Journal of Research in Health Sciences, Vol. 18, No. 2, pp. 412.
36. **Altman, N., Krzywinski, M. (2017).** Ensemble methods: Bagging and random forests. Nature Methods, Vol. 14. No. 10, pp. 933–935.
37. **Ziegler, A., König, I. R. (2013).** Mining data with random forests: Current options for real-world applications. WIREs Data Mining and

Knowledge Discovery, Vol. 4, No. 1, pp. 55–63. DOI: 10.1002/widm.1114.

38. **Sapankevych, N., Sankar, R. (2009).** Time series prediction using support vector machines: a survey. IEEE Computational Intelligence Magazine, Vol. 4, No. 2, pp. 24–38. DOI: 10.1109/mci.2009.932254.
39. **Cortes, C., Vapnik, V. (1995).** Support-vector networks. Machine Learning, Vol. 20, No. 3, pp. 273–297. DOI: 10.1007/bf00994018.
40. **Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., Chen, T. (2018).** Recent advances in convolutional neural networks. Pattern Recognition, Vol. 77, pp. 354–377. DOI: 10.1016/j.patcog.2017.10.013.
41. **O'Shea, K., Nash, R. (2015).** An Introduction to Convolutional Neural Networks. DOI: 10.48 550/arXiv.1511.08458.
42. **Galajit, K., Karnjana, J., Unoki, M., Aimmanee, P. (2019).** Semi-fragile speech watermarking based on singular-spectrum analysis with CNN-based parameter estimation for tampering detection. APSIPA Transactions on Signal and Information Processing, Vol. 8, No. 1. DOI: 10.1017/atsip. 2019.4.
43. **Greff, K., Srivastava, R. K., Koutnik, J., Steunebrink, B. R., Schmidhuber, J. (2017).** LSTM: A search space odyssey. IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 10, pp. 2222–2232. DOI: 10.1109/tnnls.2016.2582924.
44. **Hochreiter, S., Schmidhuber, J. (1997).** Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
45. **Gers, F. A., Schmidhuber, J., Cummins, F. (2000).** Learning to forget: continual prediction with LSTM. Neural Computation, Vol. 12, No. 10, pp. 2451–2471. DOI: 10.1162/089976600 300015015.
46. **Taillardat, M. (2017).** Non-parametric Methods of post-processing for Rnsemble Forecasting. PhD Thesis, Université Paris Saclay. theses.hal.science/tel-01723573/.
47. **Estrada-Patiño, E., Castilla-Valdez, G., Frausto-Solís, J., Terán-Villanueva, J. D. (2017).** Enfoque de aprendizaje híbrido evolutivo para redes neuronales en la clasificación de casos médicos. Programación Matemática y Software, Vol. 9, No. 3. DOI: 10. 30973/progmat/2017.9.3/8.
48. **Holland, J. H. (1992).** Genetic algorithms. Scientific American, Vol. 267, No. 1, pp. 66– 73.
49. **Bozdogan, H. (1987).** Model selection and Akaike’s information criterion (AIC): The general theory and its analytical extensions. Psychometrika, Vol. 52, No. 3, pp. 345–370. DOI: 10.1007/bf02294361.
50. **Yamaoka, K., Nakagawa, T., Uno, T. (1978).** Application of Akaike’s information criterion (AIC) in the evaluation of linear pharmacokinetic equations. Journal of Pharmacokinetics and Biopharmaceutics, Vol. 6, No. 2, pp. 165–175. DOI: 10.1007/bf01 117450.
51. **Neath, A. A., Cavanaugh, J. E. (2011).** The bayesian information criterion: background, derivation, and applications. WIREs Computational Statistics, Vol. 4, No. 2, pp. 199–203. DOI: 10.1002/wics.199.
52. **Gobierno de México (2022).** Servicio Meteorológico Nacional. smn.conagua. gob.mx/es/
53. **Gobierno de México (2024).** Servicio Meteorológico Nacional: Máximas y mínimas de temperatura–Ciudad de México, México. www.kaggle.com/datasets/erickestradapatio/ maximum-and-minimum-temperatures-cdmx-mxico.
54. **Instituto Nacional de Estadística y Geografía (2022).** Mapas. Climatológicos. www.inegi.org.mx/temas/climatologia/.

Article received on 28/02/2024; accepted on 15/05/2024.
**Corresponding author is Juan Frausto Solis.*