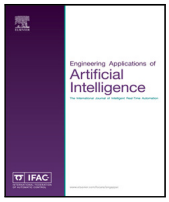
Engineering Applications of Artificial Intelligence 133 (2024) 108380

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research Paper

Fractal interpolation in the context of prediction accuracy optimization

Alexandra Băicoianu [a], Cristina Gabriela Gavrilă [a], Cristina Maria Păcurar [a,*], Victor Dan Păcurar [b]

[a] Faculty of Mathematics and Computer Science, Transilvania University of Braşov, 50 Iuliu Maniu Blvd., Braşov, Romania
[b] Faculty of Silviculture and Forest Engineering, Transilvania University of Braşov, 1 Şirul Beethoven Street, Braşov, Romania

ARTICLE INFO

Keywords:
Machine learning
Fractal interpolation
LSTM
Synthetic data
Meteorological data
Optimization

ABSTRACT

This paper focuses on the hypothesis of optimizing time series predictions using fractal interpolation techniques. In general, the accuracy of machine learning model predictions is closely related to the quality and quantitative aspects of the data used, following the principle of *garbage-in, garbage-out*. In order to quantitatively and qualitatively augment datasets, one of the most prevalent concerns of data scientists is to generate synthetic data, which should follow as closely as possible the actual pattern of the original data.

This study proposes three different data augmentation strategies based on fractal interpolation, namely the *Closest Hurst Strategy*, *Closest Values Strategy* and *Formula Strategy*. To validate the strategies, we used four public datasets from the literature, as well as a private dataset obtained from meteorological records in the city of Braşov, Romania. The prediction results obtained with the LSTM model using the presented interpolation strategies showed a significant accuracy improvement compared to the raw datasets, thus providing a possible answer to practical problems in the field of remote sensing and sensor sensitivity. Moreover, our methodologies answer some optimization-related open questions for the fractal interpolation step using *Optuna* framework.

1. Introduction

Developing successful Artificial Intelligence (AI) and machine learning (ML) models requires access to immense amounts of high-quality data, as it is widely acknowledged that the performance of most ML models depends on the quantity and diversity of data. However, collecting the necessary amount of labelled training data can be cost-prohibitive. Thus, developing various strategies to improve the quantity and quality of data is of utmost importance.

Our research focuses on the use of interpolation to enhance the quality of the predictions of ML models. The interpolation technique that we adopt is fractal interpolation which provides interpolants that are not necessarily differentiable functions at every point. Since differentiability implies smoothness and a continuous behaviour, interpolating data using functions with this property tends to oversimplify or smooth out some of the irregular and rough patterns that are specific to real-world data, especially at smaller scales. This smoothing effect can lead to a loss of crucial information, making the interpolation less suitable for fitting real-world data. Furthermore, fractal interpolation allows interpolants that can capture the inherent roughness and self-similar structures often found in real-world data. These interpolants can better replicate the complexity and irregularity present in natural phenomena.

On one hand, we develop three different strategies for the pre-processing step of data, which all use fractal interpolation. Our first strategy follows a similar approach to the one used by Raubitzek and Neubauer (2021). However, we managed to solve a series of issues, such as answering the question of optimal choice of the vertical scaling factors involved in fractal interpolation. Moreover, we present a detailed scheme of all the steps involved, which makes our research highly reproducible, and through *Optuna* framework we optimize the prediction model presented. The other two strategies, Closest Values Strategy and Formula Strategy are new approaches that have not been considered in literature before as far as we know.

On the other hand, besides testing our strategies with an ML model fed with public datasets, we also provide examples using real meteorologic data to put our research in the existing research context. This opens a new gate for researchers in the field to obtain much-needed data either when sensors break, or when finer data are needed, based on data recorded at larger time intervals. The response time of a temperature monitoring sensor is producer-dependent and defines how fast the sensor can adapt to temperature changes in a defined period of time, thus influencing the frequency of data logging. The sampling rate, which determines the time resolution of data, depends on the sensors' response time (generally correlated with the price of the device). Developing better interpolation techniques could be very useful for enabling time resolution enhancement, and making it possible, for example, to integrate in the predictive models input data (with high time resolution) subsets obtained from the raw data recorded by sensor-loggers installed in the area for climatological purposes (less expensive

* Corresponding author.
E-mail address: cristina.pacurar@unitbv.ro (C.M. Păcurar).

https://doi.org/10.1016/j.engappai.2024.108380
Received 28 May 2022; Received in revised form 9 October 2023; Accepted 1 April 2024
Available online 11 April 2024

devices, with a typical sample rate of 1 h, but with a much better spatial coverage). This matter highly sustains the utility of the present study, which aims to find new and improved techniques for data interpolation, namely for modifying data time resolution.

Generating synthetic data, or data augmentation for time-series data, has been an important research issue for many researchers. Among utilization of data augmentation, we mention augmenting sparse datasets (Forestier et al., 2017), generating controllable datasets (Kang et al., 2019), moving block bootstrap (Bergmeir et al., 2016), a.o. For a comprehensive review on time series augmentation for deep learning see Wen et al. (2021). Among applications of data augmentation, we also mention time series classification, Fawaz et al. (2018), Guennec et al. (2016), Iwana and Uchida (2021) and Kamycki et al. (2020) or improving the accuracy of forecasting (Bandara et al., 2021, Lee and Kim, 2020 and Raubitzek and Neubauer, 2021). Moreover, it is noticeable that research on time-series data augmentation proved interpolation to be a robust method (Oh et al., 2020).

Classical interpolation methods are often used in prediction machine learning techniques (Bélisle et al., 2015; Jia and Ma, 2017, Meijering, 2002, Wu et al., 2020 and Yadav and Ray, 2021). Interpolation techniques have proven to be an essential and effective tool in reconstructing incomplete datasets (Chai et al., 2021).

Raubitzek and Neubauer have recently introduced fractal interpolation in data preprocessing for machine learning (Raubitzek and Neubauer, 2021). However, our approach addressed several challenges, including determining the optimal selection of vertical scaling factors in fractal interpolation. Moreover, we optimize the presented prediction model using the Optuna framework. Two strategies that we use, namely the Closest Values Strategy and the Formula Strategy, represent novel approaches that have not been used before.

Fractal interpolation is a method for generating interpolation points between a given set of data $\Delta = \{(x_i, y_i), i \in \{0, 1, 2, \dots, N\}\}$, where N is a natural number. The main difference between fractal interpolation and other types of interpolation techniques is the outlook of the result of interpolation, which is a continuous function that is not differentiable everywhere. Thus, fractal interpolation is more relevant for fitting real-world data. Fractal interpolation has applications in a vast range of areas, such as computer graphics (Manousopoulos et al., 2008), image compression (Bouboulis et al., 2006 and May, 1996), reconstruction of satellite images (Chen et al., 2011), single-image super-resolution procedure (Zhang et al., 2018), reconstruction of fingerprint shape (Bajahzar and Guedri, 2019), signal processing (Navascués, 2010 and Zhai et al., 2011), reconstruction of epidemic curves (Păcurar and Necula, 2020) and others.

Research combining machine learning and fractal analysis features has been performed before in combination with Support-Vector Machine (see Ni et al., 2011 that focuses on the enhancement of stock trend prediction accuracy by combining a fractal feature selection method with a support vector machine, demonstrating its superiority compared to five other commonly used feature selection methods, and Wang et al., 2019 where the stock price indexes are forecasted, offering improved accuracy compared to three other commonly used models) or Time-Delayed Neural Network (see Yakuwa et al., 2003 where there is shown an improved short-term prediction accuracy compared to a back propagation-type forward neural network).

2. Materials

2.1. Datasets

This section presents the experimental datasets used in the current research study. Their properties sustain the significance of interpolation techniques in the prediction process, but also serve as validation for the methodology that we will propose with respect to the quantitative aspect of the data.

2.1.1. Meteorological data

The data set was provided by the Forest Meteorology-Climatology Laboratory, from the Faculty of Silviculture and Forest Engineering, Transilvania University of Brasov, more precisely it was extracted from the database recorded by the automatic weather station HOBO®RX3000 (research-grade), deployed at the Sânpetru Education and Research Base, located about 10 km north-east of Braşov City Centre (45.71°N, 25.65°W).

The temperature and relative humidity values were measured and recorded every 10 min by an S-THB smart sensor (Hoboware, produced by Onset Computer Corporation). This device is designed to operate in a range from −40 °C to 75 °C, with an accuracy of ±0.21 °C (from 0° to 50 °C, thus in the temperature range of the three autumn months considered in this study) and a resolution of 0.02 °C at 25 °C.

The sensor response time is 5 min (in air moving 1 m/s), consequently, the time resolution of temperature data measurements (10 min) was adequately established. For the data logger programming an important element is the sampling time interval, which depends on the sensor response time (a shorter sampling interval is not acceptable). This issue highly sustains the utility of the present study, which aims to find new, improved techniques for data interpolation, namely for modifying data time resolution. For studying the regional mountain climate, the Forest Meteorology-Climatology Laboratory operates a dense network of temperature and relative humidity data loggers (HOBO Pro v2 Temp/RH logger) deployed in Postavaru Mountains (on different elevations, aspect, wind exposure etc.) with similar accuracy and resolution, but with a response time of 40 min, which forces the sampling interval to be higher, being consequently set at 1 h (suitable for climatological studies but with the time resolution enhancement useful for other applications).

For this study, the temperature data were formatted with two decimals, as considered adequate for developing and testing the interpolation technique. For meteorological applications, the temperature data resulting from direct measurements should be rounded to one decimal (corresponding to readings at the ordinary meteorological thermometer).

The data set is composed of 13 105 temperature entries recorded between 1 September 2021, 0:00:00 and 30 November 2021, 23:50:00. The file is in a *.csv* format with a size of 385 kB.

2.1.2. Additional public datasets

In view of the research by Raubitzek and Neubauer (2021), we consider four additional public datasets: Shampoo sales with 36 data points, (Kaggle, 2022), Airline passengers with 144 data points, (Kaggle, 2022), Annual wheat yields in Austria with 57 data points (FaostatDocs, 2022), and Annual maize yields in Austria with 58 data points (FaostatDocs, 2022).

The consistent differences between our study and the research from Raubitzek and Neubauer (2021) are convincingly outlined by using the same datasets. Moreover, comparing the methods on the same datasets highlights the progress in this direction of research that our study provides.

2.2. Prerequisites

2.2.1. Fractal interpolation

Fractal interpolation was introduced by Barnsley (2012) and it has since been intensively studied and applied.

To interpolate the data set $\Delta = \{(x_i, y_i), i \in \{0, 1, 2, \dots, N\}\}$ consider the equations

$$\begin{aligned} a_i &= \frac{x_i - x_{i-1}}{x_N - x_0} \\ c_i &= \frac{x_N x_{i-1} - x_0 x_i}{x_N - x_0} \\ d_i &= \frac{y_i - y_{i-1}}{x_N - x_0} - s_i \frac{y_N - y_0}{x_N - x_0} \\ e_i &= \frac{x_i y_{i-1} - x_0 y_i}{x_N - x_0} - s_i \frac{x_N y_0 - x_0 y_n}{x_N - x_0}, \end{aligned} \tag{1}$$

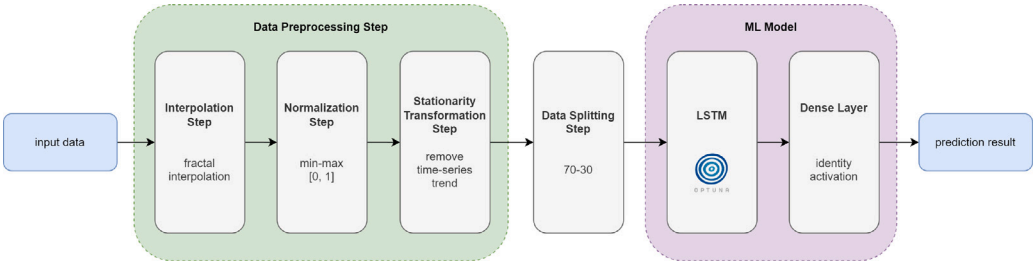


Fig. 1. Methodology outline.

where $s_i \in (-1, 1)$ is called the vertical scaling factor.

Let the family of functions $f_i : [x_0, x_N] \times Y \to [x_0, x_N] \times Y$ defined as

$$f_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & 0 \\ d_i & s_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} c_i \\ e_i \end{pmatrix}, \tag{2}$$

for every $i \in \{1, \dots, N\}$.

Given a metric space (X, d), the pair $((X, d), (f_i)_{i\in\{1,\dots,N\}})$ is called an iterated function system (IFS, for short) if:

(i) (X, d) is a complete space;
(ii) the functions f_i are continuous, for every $i \in \{1, \dots, N\}$.

The concept of IFS is a notion due to Hutchinson (1981). For an IFS, the fractal operator $F_S : \mathcal{P}(X) \to \mathcal{P}(X)$ is defined as $F_S(K) = \cup_{i\in\{1,\dots,N\}} f_i(K)$, for every $K \in \mathcal{P}(X)$, where $\mathcal{P}(X)$ represents the set of all subsets of X.

Taking $X = [x_0, x_N]$ with the Euclidean metric and the functions f_i defined in Eq. (2), we obtain an IFS.

The fixed point of the fractal operator associated with an iterated function system composed of the functions $(f_i)_{i\in\{1,\dots,N\}}$ is an interpolation function for the system of data Δ called the fractal interpolation function.

We construct the fractal interpolation part based on the above formulas and the code provided by Barnsley in Chapter IV of *Fractals everywhere* (Barnsley, 2012).

2.2.2. Optuna framework

Optuna is an open-source hyperparameter optimization framework that provides multiple state-of-the-art algorithms for sampling hyperparameters ranging from grid sampling strategies to genetic algorithms approaches (Optuna-ReadTheDocs, 2023b).

The main steps for using *Optuna* are as follows:

- Define an objective function to be optimized.
- Create an optimization study *optuna_study*, which will determine the best parameters by running several trials. A trial can be defined as a single execution of the objective function.
- Use one or multiple *suggest* API function calls for the parameters that are subject to optimization inside a trial.

The default hyperparameter sampler is TPESampler which implements the Tree-structured Parzen Estimator algorithm (Optuna-ReadTheDocs, 2023a). The algorithm starts by running the objective function on randomly sampled hyperparameter values from the given domain. After a number of observations, the results are divided into two groups depending on whether they fall below or above a certain quantile of the observed values of the objective function, thus separating the best hyperparameter values from the others. In every iteration, the two groups are updated and a Gaussian Mixture Model (GMM) is fitted to each group, resulting in two densities, $l(x)$ for the best hyperparameters values and $g(x)$ for the remaining ones, where x is the value of the hyperparameter. The algorithm will select the value that maximizes the ratio $\frac{l(x)}{g(x)}$.

3. Method and procedures

This section presents the methodology of the present study. We emphasize the steps which are of utmost importance for the final results in Fig. 1. Furthermore, in the following sections, we will explore each block included in the diagram and highlight its role in the whole process.

It is noteworthy that the main contribution of this paper is concentrated on the interpolation step, which is placed in a time series prediction pipeline. The main goal is to evaluate how the data augmentation step, through fractal interpolation, can have an impact on the quality/accuracy of the prediction.

The field of modelling weather and climate is getting increasingly popular, so choosing a suitable learning machine model becomes a challenge. LSTM models are particularly suited for predicting climate change because they can recall and utilize past data to inform future forecasts. Seasonality and patterns in climate data are frequently visible and can last many years. Based on past data, LSTM models may successfully identify these patterns and produce precise forecasts. The capacity of LSTM models to manage missing data is one of their key characteristics. Due to several issues, including sensor malfunctions and data gathering gaps, climate data is frequently unreliable. While LSTM models can make predictions even with insufficient data, traditional statistical methods have difficulty handling missing data.

LSTM models can also be trained to adjust to changing situations. Since climate change is dynamic, conventional statistical models frequently have difficulty adjusting to new patterns and trends. The ability to train LSTM models on an ongoing stream of data, on the other hand, enables them to adjust and produce precise forecasts even as the climate changes.

For all these reasons exposed, in this research, we chose an LSTM model to explore the predictions on the data used.

3.1. Data preprocessing step

Data preprocessing is an essential step in the development of successful ML models. This technique requires some data preparation, including cleaning the data and transforming the data such that their quality is enhanced. Incomplete, inconsistent, or inaccurate data that contain errors or outliers can be eliminated in this preprocessing step. There are numerous preprocessing techniques (for a comprehensive book on data preprocessing see Garcia et al., 2014) that produce quality data that lead to high-quality patterns.

Our preprocessing step includes transformations of data (interpolation, normalization, and stationarity) to obtain the most suitable data for applying ML algorithms.

3.1.1. Interpolation step

Real-world data are often noisy, with incorrect or missing values. Interpolation is a method of creating new data points within the range of known data points, so it is a technique for filling in missing values. The interpolation should be used where there is a trend observed in the input data and the requirement is to fill the missing value along with the same trend.

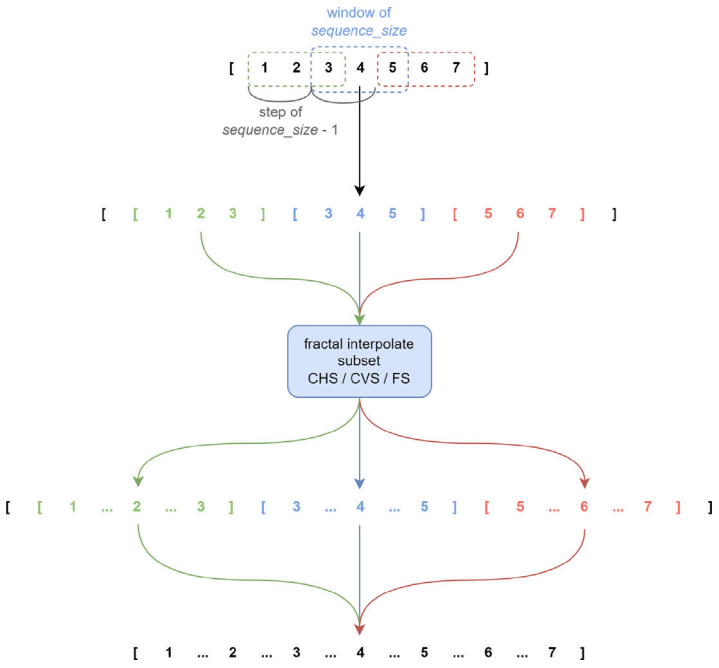


Fig. 2. Interpolation flow diagram for *sequence_size* 3.

We present the complete and detailed steps required for the interpolation step, applied to the datasets considered.

Substep 1: Divide the time series into m subsets of size *sequence_size*

This step consists of splitting the given sequence into m subsets of length *sequence_size* so that the last value in subset i is the first value in the subset $i+1$.

As regards the way we implement this substep, there are some remarks that are worth mentioning:

Remark 1. If the algorithm is used in strict mode, then the dataset is divided into m sequences with equal dimensions; otherwise, the last subset might have a dimension between 3 and *sequence_size* - 1 (at least 3 because 2 points cannot generate new intermediate points in the interpolation process).

Remark 2. At the end of the interpolation, the subsets need to be reunited into a unique list that contains the initial points, chronologically and without repetitions.

Substep 2: The proposed interpolation strategies

In this section, we present three different strategies for the steps specific to interpolation, namely *Closest Hurst Strategy (CHS)*, *Closest Values Strategy (CVS)* and *Formula Strategy (FS)*.

We choose different strategies, firstly to obtain validation for the results from Raubitzek and Neubauer (2021), and most importantly, to enhance the results and obtain improved techniques. We test our methods for both the public datasets Maize (Annual maize yields in Austria), Shampoo Sales, Airline Passengers, Wheat (Annual wheat yields in Austria), see details in Section 2.1, as well as the original data set, Weather described in Section 2.1.1. As regards the latter, for all strategies we use the data corresponding to the first week of entries (1 September 2021–8 September 2021) and we chose the data recorded every hour, to better outline the significance of interpolation.

Fig. 2 describes the general flow of the interpolation step. While the diagram is constructed considering a *sequence_size* of 3, note that a particular *sequence_size* was used for implementing the proposed methodologies. Specific details are given in the next sections for each of the defined strategies.

The interpolation step is presented in Algorithm 1, and it is applicable for all the next proposed strategies. The description of the parameters for the FRACTAL_INTERPOLATION procedure are:

- *subset*: subset created from original data as described in **Substep 1**.
- s_i: a vector of vertical scaling factors which dictate how jagged (and fractal) will be the aspect of the generated data, in the sense that, as its name states, it scales the points vertically.
- *n_interpolation*: the number of distinct interpolation points to be generated between every 2 points of the original data

Algorithm 1 Pseudocode for Fractal Interpolation Computation

procedure FRACTAL_INTERPOLATION(*subset*, s_i, *n_interpolation* = 17)
  Compute the interpolation factors a_i, c_i, d_i și e_i based on Equation (1)
  Generate interpolation points based on Equation (2) until between every 2 points from the *subset* there are *n_interpolation* distinct interpolation points

I. Closest Hurst Strategy (CHS)

The first Strategy is similar to the one employed by Raubitzek and Neubauer (2021). We will refer to it as Closest Hurst Strategy (CHS).

For each subset resulting from Interpolation Substep 1 with *sequence_size* 10, the Algorithm 2 is performed. Note that the parameters have the same signification as previously described.

Results and analysis for Closest Hurst Strategy

We present the results obtained for the considered datasets based on CHS. Firstly, we show the outcome for the public datasets, with the parameter $s_i \in [-1, 1]$ (Figs. 3–6).

However, following tests, we found that the most appropriate vertical scaling factor must be chosen between $s_i \in [0, 0.2]$. In this case, the differences between the points obtained and the initial points are limited, and the initial outlook of the graphic defined by the initial points is conserved. We show in Figs. 7–10 the results obtained in this case.

For our private data set, Weather, we obtain the results presented in Figs. 11 and 12.

Algorithm 2 Pseudocode of Closest Hurst Strategy

```
procedure CLOSEST_HURST_STRATEGY(subset, n_interpolation = 17)
    Compute the initial_hurst, the initial Hurst exponent
    Generate s_i ∈ [−1, 1] a vector with the same value on all positions, representing the
constant vertical scaling factor for the current subset
    for k ← 1, 15 do
        interpolated_subset ← FRACTAL_INTERPOLATION(subset, s_i, n_interpolation)
        Compute the h_new, the Hurst exponent for the interpolated_subset
        if k = 1 then
            h_old ← h_new
            interpolated_result ← interpolated_subset
        else
            if abs(h_new - initial_hurst) < abs(h_old - initial_hurst) then
                h_old ← h_new
                interpolated_result ← interpolated_subset
            else
                Generate a new s_i ∈ [−1, 1]
    return interpolated_result
```

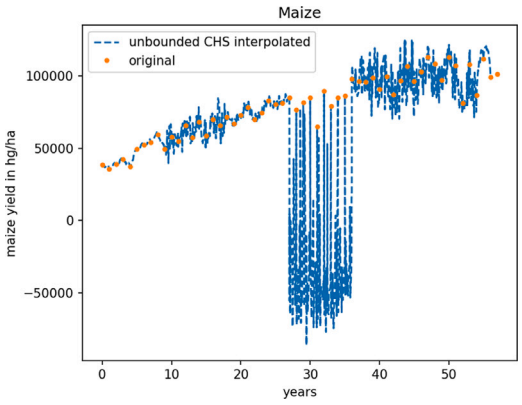


Fig. 3. Maize data set, CHS.

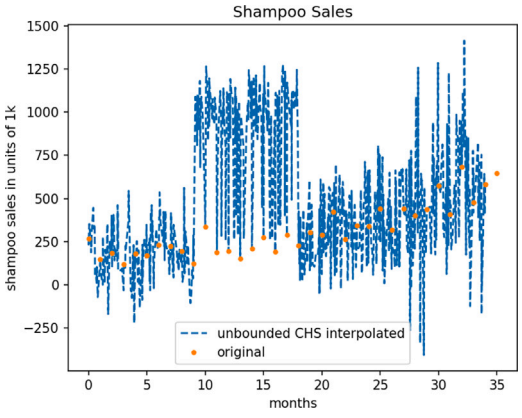


Fig. 4. Shampoo Sales data set, CHS.

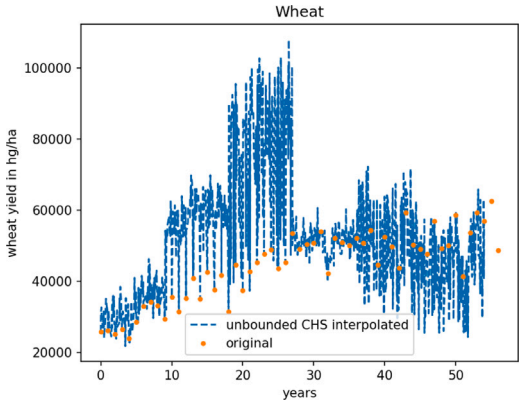


Fig. 5. Wheat data set, CHS.

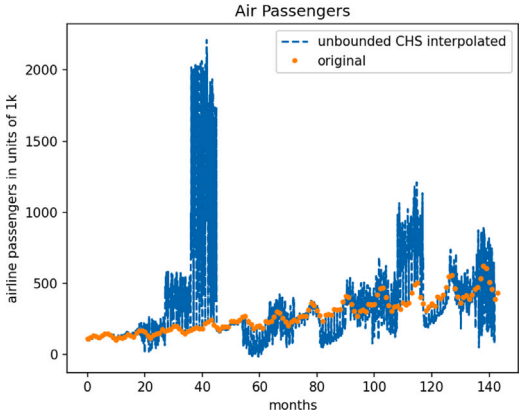


Fig. 6. Air Passengers data set, CHS.

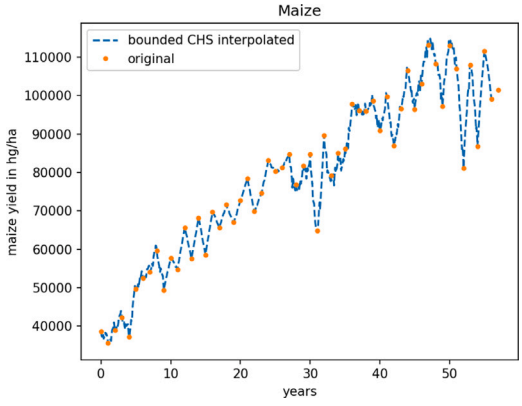


Fig. 7. Maize data set, CHS.

Although the stop condition ensures that the Hurst exponent for the interpolated data is close enough to the initial Hurst value, this does not guarantee the persistence of other properties of the data. This motivates us to define new strategies that ensure the preservation of certain properties of the data through interpolation.

II. Optimized procedure - Closest Values Strategy (CVS)

For this type of strategy, we propose the *Optuna* framework, described in Section 2.2.2.

Thus, in the current CVS strategy, for each data subset obtained using *sequence_size* 10, the steps from Algorithm 3 are performed. Observe that Algorithm 3 defines two procedures, the first one is the objective function that *Optuna* will minimize over the course of 15 trials, having the new parameter *optuna_trial*, and the second one is the main implementation of the strategy. Additionally, procedure LINEAR_INTERPOLATION constructs a linear interpolation of the *subset* considering each interpolation point generated by the FRACTAL_INTERPOLATION procedure, and RMSE represents the Root Mean Square Error.

Results and analysis for Closest Values Strategy

In Figs. 13–17 there are shown the results of the proposed strategy for all five datasets based on CVS. The parameter s_i was optimized by *Optuna* with a possible range set in the interval $[-1, 1]$.

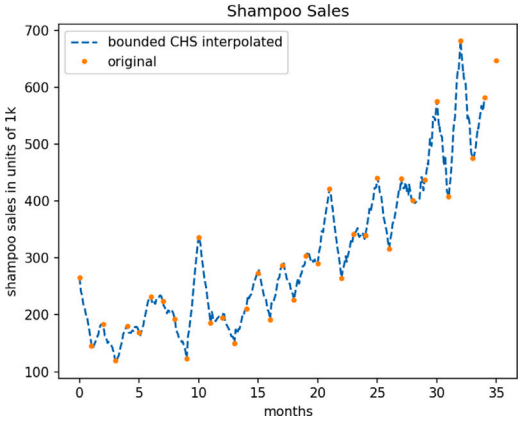


Fig. 8. Shampoo Sales data set, CHS.

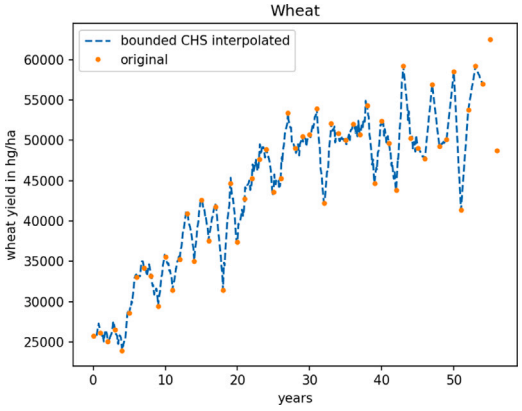


Fig. 9. Wheat data set.

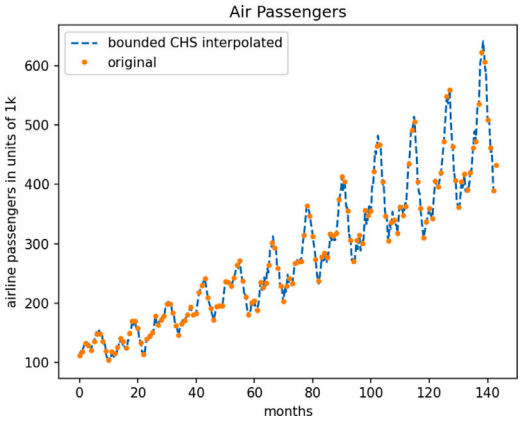


Fig. 10. Air Passengers data set.

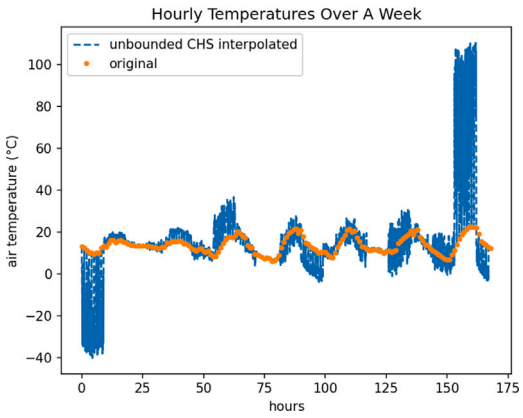


Fig. 11. Weather data set with $s_i \in [-1, 1]$.

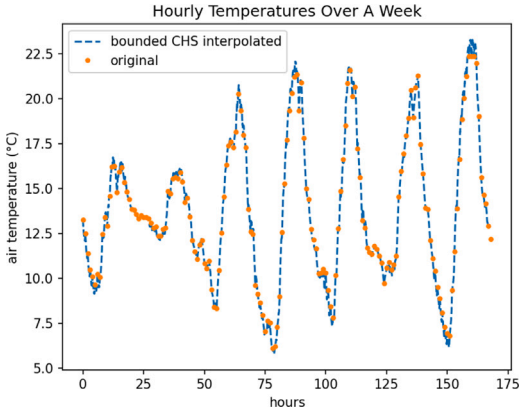


Fig. 12. Weather data set with $s_i \in [0, 0.2]$.

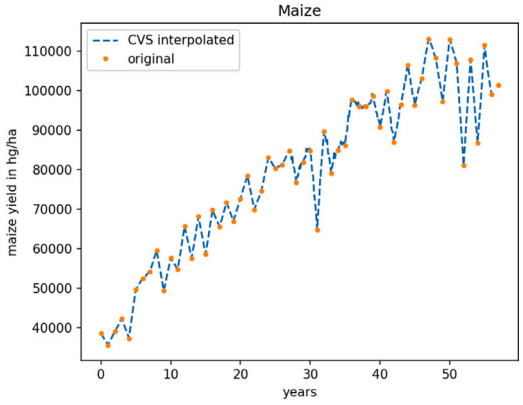


Fig. 13. Maize data set, CVS.

Algorithm 3 Pseudocode of Closest Values Strategy

procedure CLOSEST_VALUES_STRATEGY_OBJECTIVE(*optuna_trial*, *subset*, *n_interpolation* = 17)
  Generate $s_i \in [-1, 1]$, a vector with the same value on all positions, representing the constant vertical scaling factor for the current *subset* in the current *optuna_trial* using *suggest* API
  interpolated_subset ← FRACTAL_INTERPOLATION(*subset*, s_i, *n_interpolation*)
  linear_interpolated_subset ← LINEAR_INTERPOLATION(*subset*)
  return RMSE(*interpolated_subset*, *linear_interpolated_subset*)
▷
procedure CLOSEST_VALUES_STRATEGY(*subset*, *n_interpolation* = 17)
  Create *optuna_study*, a study with direction 'minimize', the objective function CLOSEST_VALUES_STRATEGY_OBJECTIVE() and 15 trials
  s_i ← best trial parameter of *optuna_study*
  return FRACTAL_INTERPOLATION(*subset*, s_i, *n_interpolation*)

It is noticeable that the graphics obtained using CVS resemble the results from CHS with $s_i \in [0, 0.2]$. This is because the RMSE is minimum for the parameter s_i close to the interval $[0, 0.2]$. This can be observed in Fig. 18 where the evolution of the parameter s_i is presented with respect to the objective function described in the CVS context. This result validates our choice of the parameter s_i in the case of CHS.

III. Optimized strategy - Formula Strategy (FS)

For this method, we shall follow a different direction, that is to use a formula to optimize the parameter s_i. Although there is no way to determine a general optimal vertical scaling factor, there are various approaches to optimize this parameter. We follow the ideas from Mazel

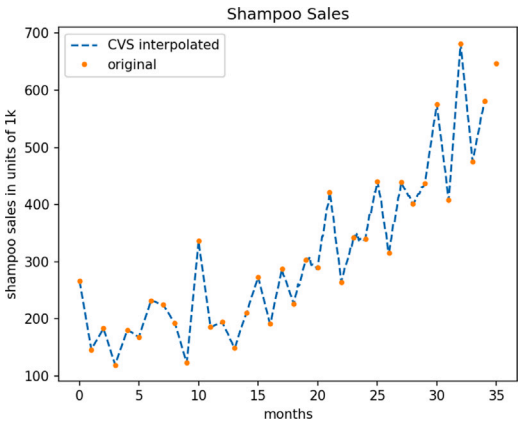


Fig. 14. Shampoo Sales data set, CVS.

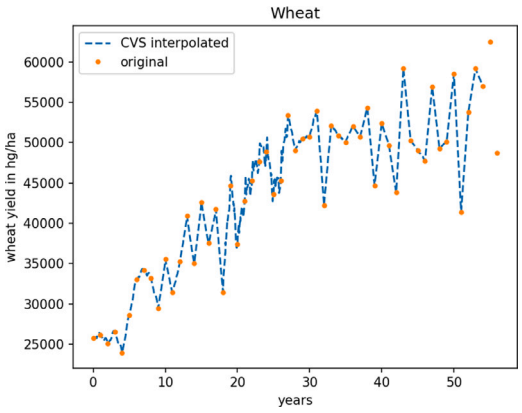


Fig. 15. Wheat data set, CVS.

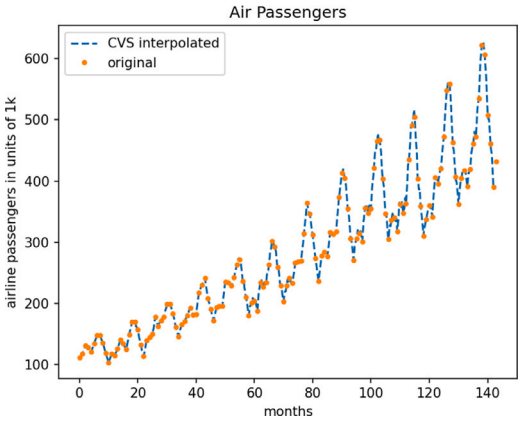


Fig. 16. Air Passengers data set, CVS.

and Hayes (1992), Manousopoulos et al. (2011) and Gowrisankar et al. (2022).

For the data set $\Delta = \{(x_i, y_i), i \in \{0, 1, 2, \dots, N\}\}$, we choose the parameter s_i as follows:

$$s_i = \frac{y_i - y_{i-1}}{\sqrt{(y_N - y_0)^2 + (y_i - y_{i-1})^2}}, \tag{3}$$

for each $i \in \{1, 2, \dots, N\}$.

However, since the denominator in Eq. (3) becomes closer to 0 when the first and the last data in the subset are too close (the line determined by the start point and ending point of the subset is parallel to the Ox axis), then s_i becomes irrelevantly big, inducing an unwanted variation in the data. Thus, we avoid this by optimizing the current strategy. The optimization is achieved by modifying the *sequence_size* parameter such that the difference between the two ends does not tend to zero.

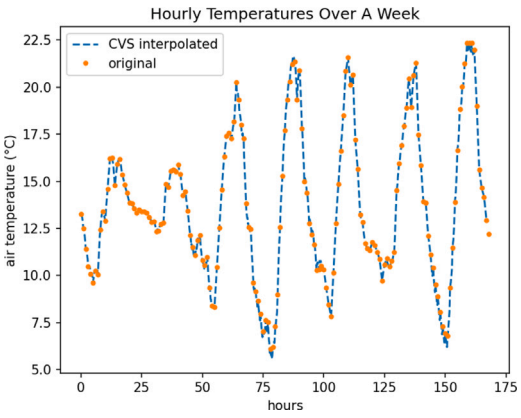


Fig. 17. Weather data set, CVS.

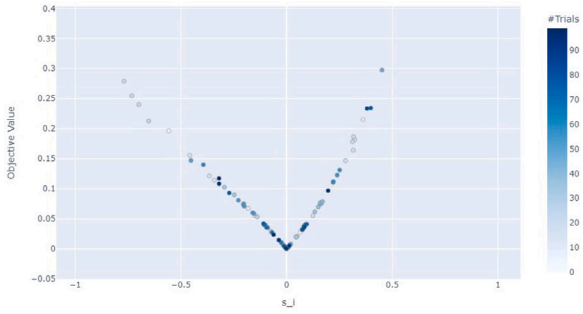


Fig. 18. Evolution of parameter s_i with *Optuna*.

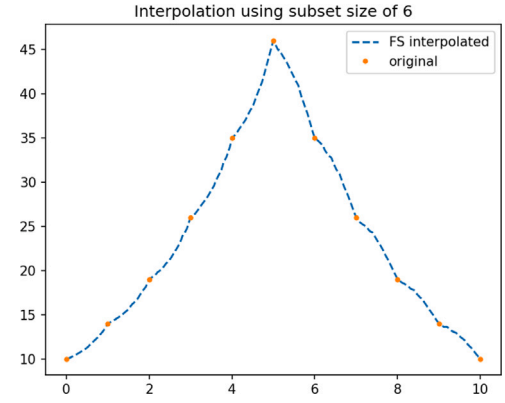


Fig. 19. Interpolation of Γ with *sequence_size* 6.

To exemplify the dependence of the interpolated data on the *sequence_size* parameter chosen, let us consider a trial data set

$$\Gamma = \{(1, 10), (2, 14), (3, 19), (4, 26), (5, 35),$$
$$(6, 46), (7, 35), (8, 26), (9, 19), (10, 14), (11, 10)\}.$$

In Figs. 19 and 20, there are presented the results of interpolation with FS for two different values for *sequence_size*, 6 and 11 respectively.

Therefore, in the context of FS, the initial step is determining the optimal value of the *sequence_size*. For this, we used the procedures defined in Algorithm 4. Note that the optimization is computed for the entire dataset using *Optuna* with 50 trials, as opposed to previous

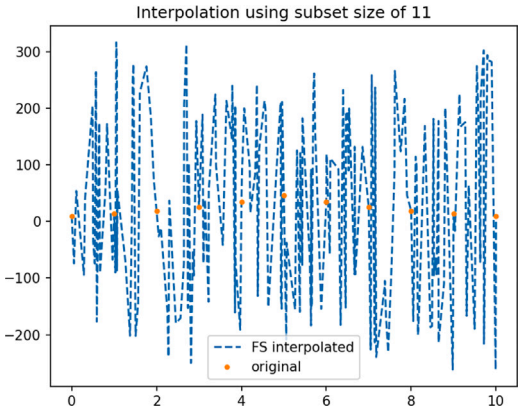


Fig. 20. Interpolation of Γ with *sequence_size* 11.

optimization strategies where the optimization was done at the subset level.

To determine the optimal values for *sequence_size*, we consider the search interval $[4, \text{length(dataset)}-3]$ and allow the fractal interpolation algorithm to work in the non-strict mode, see Remark 1, to minimize data loss.

Algorithm 4 Pseudocode of *sequence_size* optimization

procedure OPTIMIZE_SUBSET_LENGTH_OBJECTIVE(*optuna_trial*, *dataset*, *n_interpolation* = 17)
  Generate subset length *sequence_size* ∈ [4, length(*dataset*) − 3] in the current *optuna_trial* using *suggest* API
  Split *dataset* into *subsets* of length *sequence_size* as described in **Substep 1** from Section 3.1.1
  $total\_RMSE \leftarrow 0$
  for each *subset* **in** *subsets* **do**
    interpolated_subset ← FORMULA_STRATEGY(*subset*, *n_interpolation*)
    linear_interpolated_subset ← LINEAR_INTERPOLATION(*subset*)
    $total\_RMSE \leftarrow total\_RMSE + \text{RMSE}(interpolated\_subset, linear\_interpolated\_subset)$
  return $total\_RMSE$
▷
procedure OPTIMIZE_SUBSET_LENGTH(*dataset*, *n_interpolation* = 17)
  Create *optuna_study*, a study with direction 'minimize', the objective function OPTIMIZE_SUBSET_LENGTH_OBJECTIVE() and 50 trials
  sequence_size ← best trial parameter of *optuna_study*
  return *sequence_size*

With regards to FS, for each subset of data, the procedure from Algorithm 5 is executed.

Algorithm 5 Pseudocode of Formula Strategy

procedure FORMULA_STRATEGY(*subset*, *n_interpolation* = 17)
  Compute s_i based on Equation (3)
  return FRACTAL_INTERPOLATION(*subset*, s_i, *n_interpolation*)

Let us note that for the procedure FORMULA_STRATEGY, s_i is not constant for the entire subset (as is the case for CHS and CVS), but it varies according to each interval defined by two consecutive points in the subset.

One of the main advantages of this strategy is that once the optimal dimension of the subset is found, the repetitive process required to execute the optimization routine for the previous two strategies is no longer required.

The optimal value of *sequence_size* for each data set is presented in Table 1.

Results and analysis for formula strategy

Applying formula (3) using the optimal *sequence_size* values from Table 1, we obtain the interpolation results presented in Figs. 21–25.

Table 1
Optimal *sequence_size* values for FS interpolation.

Data set	*sequence_size*
Maize	29
Shampoo Sales	10
Wheat	54
Air Passengers	141
Weather	6

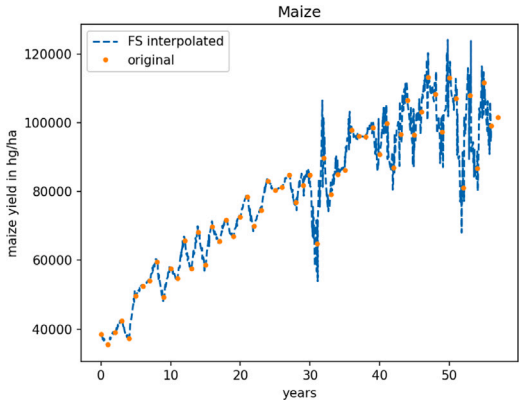


Fig. 21. Maize data set, FS.

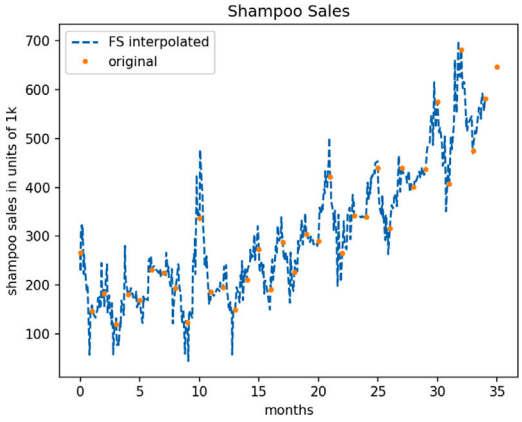


Fig. 22. Shampoo Sales data set, FS.

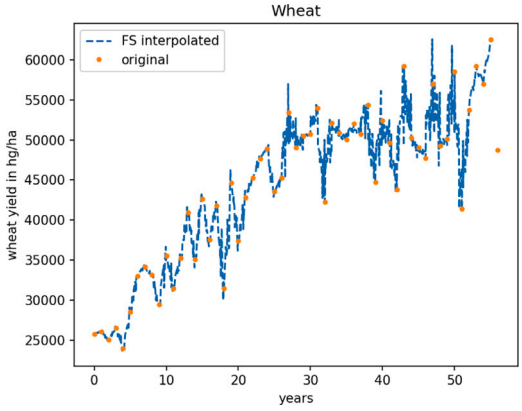


Fig. 23. Wheat data set, FS.

Comparison of methods and results

We proposed three strategies for the interpolation step, each with a different approach. To obtain a better understanding of the differences

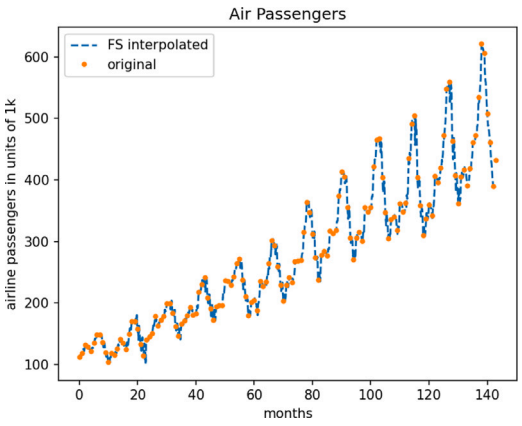


Fig. 24. Air Passengers data set, FS.

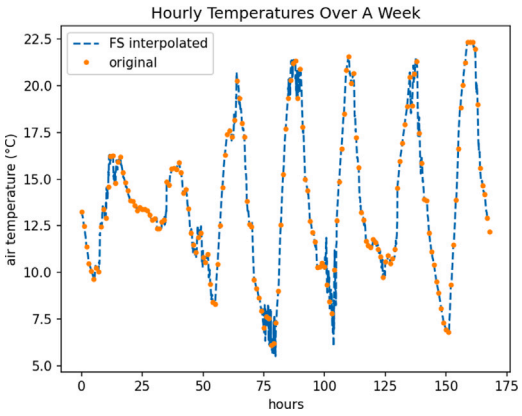


Fig. 25. Weather data set, FS.

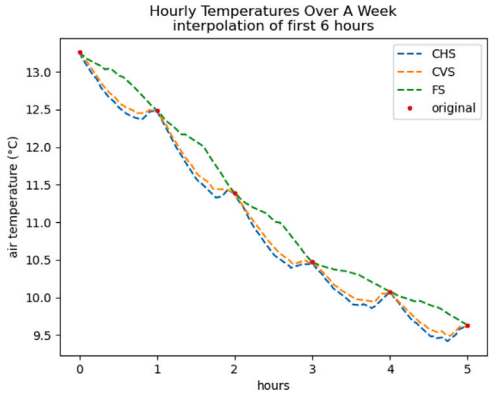


Fig. 26. First 6 points.

between the three methods, we depict in Figs. 26 and 27 the interpolation results for our Weather data set provided by the three strategies on the same graphic.

We can observe that CHS (with $s_i \in [0, 0.2]$) and CVS approaches provide similar results, while the FS approach determines slightly higher variations.

To emphasize the comparison, let us use the original Weather data set. We extract hourly data and use the three strategies for fractal interpolation with a number of five interpolation points (*n_interpolation* = 5) to simulate 10-min data.

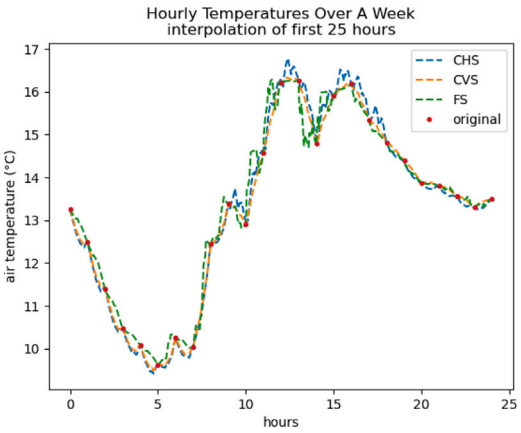


Fig. 27. First 25 points.

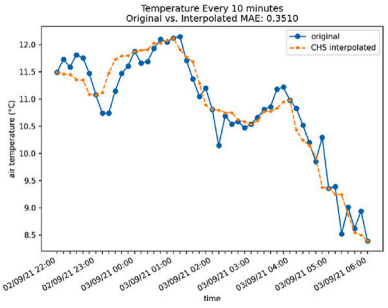


Fig. 28. MAE, CHS.

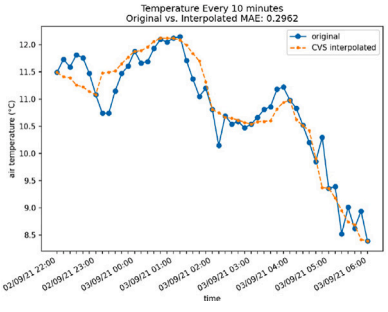


Fig. 29. MAE, CVS.

In Figs. 28–30 there are presented the results for data recorded between 02/09/21, 22:00 and 03/09/21, 06:00 compared to the original data for all three strategies. To obtain a better understanding of the differences between the three strategies, we computed the Mean Absolute Error (MAE) for each data set, which provides us with the mean difference, in degrees, between the real and the interpolated data.

As regards the Weather data set, for CHS we obtain $MAE = 0.3510$, for CVS we have $MAE = 0.2962$ and for FS we get $MAE = 0.4997$. Thus, we can observe that the least MAE is obtained for CVS, thus, this strategy is the optimal one, followed by CHS and FS.

3.1.2. Normalization step

Usually, the normalization step's objective is to convert an attribute's values to a better range. There are more strategies to normalize data. We chose to scale the data using a method similar to the *min–max* method, which performs a linear transform of the data in a given interval.

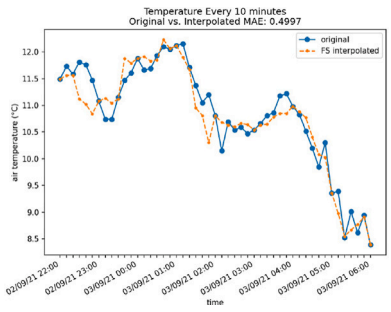


Fig. 30. MAE, FS.

More precisely, to normalize data $x' \in [0, 1]$ we use the formula

$$x' = \frac{x - \min x}{\max x - \min x}.$$

For data $x'' \in [-1, 1]$ the formula becomes

$$x'' = 2\frac{x - \min x}{\max x - \min x} - 1. \tag{4}$$

In general, for data $x''' \in [a, b]$, where $a, b \in \mathbb{R}$ the formula becomes

$$x''' = (b - a)\frac{x - \min x}{\max x - \min x} + a.$$

The purpose of normalization is to transform data in a way that they are either dimensionless and/or have similar distributions. Normalization is an essential step in data preprocessing in any ML algorithm and model fitting. Propagated errors must not have high values, especially in the case of recurrent neural networks (as is LSTM). Moreover, data normalization allows comparison of the results over data with a different configuration, thus reducing biased prioritization of some features over others, which can be caused by providing data with different features that have wide-scale differences to the model.

3.1.3. Stationarity in time series analysis

Stationarity is an important concept in the field of time series analysis with tremendous influence on how the data is perceived and predicted. It indicates whether statistical properties such as mean, variance, and autocorrelation of a time series change over time.

To determine whether a data set must be transformed, we use the Augmented Dicky-Fuller test which uses the coefficient that defines the unit root, p-value. If the p-value obtained is below 0.05, then the current data set is stationary.

For maintaining data stationarity, several transformations can be applied to eliminate trends and seasonality of a data set. In Table 2 there are presented the p-values obtained after the three types of transformations used for eliminating the trend: Log Transformation, Square Root Transformation and Linear Regression Transformation.

Since for the hourly temperatures over a week, the data set is already stationary and no transformation was required as can be seen in Table 2, we manually created a data set composed of the daily maximum temperature recorded.

Daily temperature extremes are more informative than the mean value, indicating the range over this time interval, which is highly important for interpreting its decisive influence on various processes. The minimum temperature generally occurs at dawn, at the end of the negative net radiation interval, when atmospheric status in the boundary layer is rather stable. The situation is completely different as concerns the maximum daily temperature, which arises at midday, after the heating peak, which also induces a considerable enhancement of thermal turbulence, causing increased air temperature variability. Consequently, the daily maximum temperature was chosen for this study, considering that fractal interpolation and a machine learning approach could significantly improve its assessment.

As a result of the different dimensions of the initial and interpolated data set, the p-values obtained are different. For comparison, we also perform linear interpolation. We can notice from Table 2 that regardless of the interpolation strategy employed, the p-values are close to the value obtained for linear interpolation.

We highlight in Table 2 the transforms that are maintained for each strategy.

As seasonality is regarded, a frequent step is the differentiation of data. However, since fractal interpolation produces functions that are continuous, but are not necessarily everywhere differentiable, this step cannot be considered.

Moreover, the LSTM model has a sufficient complexity to process non-stationary data, as is not the case for other statistic models like Autoregressive Integrated Moving Average (ARIMA) which depends closely on data stationarity. For the LSTM model, data normalization is more significant.

3.2. Data splitting step

For all datasets, 70% of them are retained, while the remaining 30% are allocated to the test data set, in chronological order.

For the Weather data set, the division of the data set is presented in Table 3.

The current division presented in Table 3 is relevant for the Weather data set and does not alter the predictions as in the Brasov mountain region, also including the neighbouring depressionary area, the three autumn months (September, October, November) are generally characterized by a more stable weather regime as compared with the symmetrical springtime interval. There are some sudden cold intervals, especially at the end of September or the beginning of October, but after this more dynamic period, the weather pattern shows prolonged intervals of fine weather, favourable for mountain tourism, which is also encountered towards the end of November in many years.

3.3. Model description

The selected prediction model uses an LSTM layer, followed by one dense layer with dimension 1 since the datasets have only one feature (see Fig. 31).

For each data set, the number of hidden layers in LSTM was optimized using *Optuna*. For the Weather data set, the obtained values are presented in Table 4.

LSTM is configured with the default parameters. The metric used for measuring the loss in the training step is Mean Squared Error (MSE) and for the general evaluation of the model we used RMSE.

3.3.1. Specific input structure of LSTM

To be able to feed the data to the LSTM layer, we must transform the data set into a supervised learning format. This is achieved by sliding a window of size *input_data_points* over the whole data set using a step of 1. The obtained subsets will represent the inputs for the network. In the case of Univariate Time Series prediction, the output for subset i $[data_i, data_{\text{i+input\_data\_points-1}}]$ will be $data_{\text{i+input\_data\_points}}$.

The LSTM layer implementation in *Keras - Tensorflow Keras 2.4.1*, expects the input data to be in the format of [*batch_size*, *input_data_points*, *features*]. In our case, we used a batch size of 1.

Special attention is needed such that the window size *input_data_points* considered must be less than 30% of the entire dataset. Otherwise, using the testing set to evaluate the model's performance will not be possible, as for a single data point prediction, *input_data_points* entries are required in the supervised format.

Table 2
p-values obtained for the performed transformations.

Data set	Interpolation strategy	None	Log	Square	Linear regression
Shampoo Sales	None	1.0000	0.9983	0.9991	**0.0000**
	Linear	0.7860	0.8076	0.7829	0.0655
	CHS	0.9655	0.8245	0.9221	**0.1022**
	CVS	0.9755	0.9353	0.9678	**0.0839**
	FS	0.8744	0.0423	0.7658	**0.0054**
Air Passengers	None	0.9919	**0.4224**	0.9181	0.4415
	Linear	0.0934	0.2104	0.1522	0.0001
	CHS	0.1250	0.2869	0.2118	**0.0002**
	CVS	0.1334	0.2248	0.1919	**0.0001**
	FS	0.0919	0.2018	0.1442	**0.0001**
Wheat	None	0.0607	**0.0018**	0.0122	0.9983
	Linear	0.4275	0.3043	0.3704	0.8529
	CHS	0.4840	**0.2979**	0.3924	0.9303
	CVS	0.4997	**0.3335**	0.4207	0.9110
	FS	0.6144	**0.4094**	0.5235	0.8590
Maize	None	0.2370	**0.0207**	0.1073	0.9986
	Linear	0.4905	0.1421	0.3132	0.9008
	CHS	0.4047	**0.0497**	0.1863	0.9751
	CVS	0.4196	**0.0820**	0.2131	0.9798
	FS	0.4110	**0.0965**	0.2446	0.9493
Hourly Temperatures Over a Week	None	**0.0000**	–	–	–
	Linear	0.0000	–	–	–
	CHS	**0.0000**	–	–	–
	CVS	**0.0002**	–	–	–
	FS	**0.0000**	–	–	–
Max Daily Temperature	None	0.1210	0.8297	**0.1160**	0.8669
	Linear	0.0343	0.0772	0.0510	0.4679
	CHS	**0.1374**	0.1942	0.1504	0.7768
	CVS	0.2860	**0.2697**	0.3451	0.8406
	FS	0.0714	**0.0453**	0.0655	0.5549

Table 3
Splitting of the weather data set.

Data set	Train data	Test data
Hourly Temperature	01/09/21, 00:00–05/09/21, 21:00	05/09/21, 21:00–08/09/21, 00:00
Daily Maximum Temperature	01/09/21–02/11/21	02/11/21–30/11/21

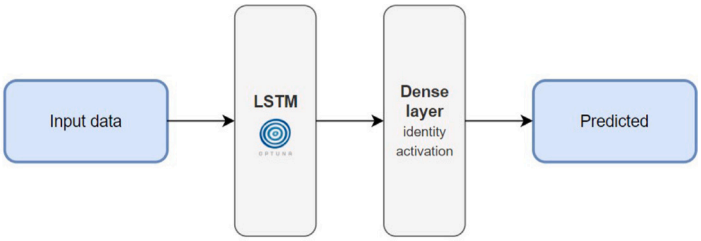


Fig. 31. Neural network structure.

Table 4
Number of hidden layers.

Interpolation strategy	Hidden layers
None	48
CHS	39
CVS	28
FS	10

3.3.2. Model optimization

Optimization plays an important role in a machine learning solution and the step of tuning a chosen model is critical to the model's performance and accuracy. Hyperparameter tuning intends to find the hyperparameters of a given machine learning algorithm that guarantee the best performance as measured on a validation set. Saving time, but also eliminating the chance of overfitting or underfitting influenced the interest in hyperparameter tuning research. Thus, entire branches of machine learning and deep learning theory have been dedicated to the optimization of models.

In our study, we use *Optuna - Optuna 2.10.0* in two key points of our implementation:

1. optimizing the vertical scaling factor s_i for the interpolation step;
2. fine-tuning LSTM network hyperparameters.

For the LSTM network model, we considered the following hyperparameters for optimization:

- *units*: the number of hidden layers used for the LSTM layer, which defines the complexity of the model. There is no formula or rule to determine this number, so it is a natural decision to pick this hyperparameter for the optimization process search space. Our choice is the interval $[2, 64]$;
- *input_data_points* (timesteps): given the data structure expected by the LSTM network, this represents the number of consecutive

Table 5
Model tuning and hyperparameters optimization.

Data set	Linear regression	Hidden layers	Input data points	Epochs	Learning rate	Train RMSE	Test RMSE
Shampoo Sales	None	62	8	80	0.02	0.1717	0.4997
	CHS	48	16	15	0.04	0.0859	0.0951
	CVS	41	28	10	0.01	0.0607	**0.0749**
	FS	17	1	8	0.005	0.1218	0.1264
Air Passengers	None	56	14	65	0.007	0.0947	0.1348
	CHS	33	97	8	0.03	0.0285	0.0655
	CVS	62	51	5	0.02	0.0288	**0.0444**
	FS	29	93	7	0.03	0.0416	0.0619
Wheat	None	13	5	125	0.005	0.1480	0.2410
	CHS	8	93	12	0.03	0.0559	0.0653
	CVS	11	98	10	0.01	0.0431	**0.0552**
	FS	51	99	12	0.01	0.0656	0.0932
Maize	None	52	3	150	0.05	0.1327	0.1766
	CHS	32	82	15	0.01	0.0299	0.0304
	CVS	39	1	10	0.002	0.0187	**0.0273**
	FS	27	20	20	0.007	0.0920	0.0991
Hourly Temperatures Over a Week	None	48	8	60	0.008	0.1111	0.1383
	CHS	39	22	7	0.005	0.0325	0.0414
	CVS	28	34	15	0.001	0.3446	**0.0389**
	FS	10	17	14	0.005	0.0391	0.0462
Daily Max Temperatures	None	43	5	175	0.02	0.1132	0.4944
	CHS	47	84	12	0.03	0.0321	**0.0523**
	CVS	49	11	20	0.005	0.0290	0.0692
	FS	50	27	15	0.005	0.0526	0.1351

input values considered for an output value in the supervised learning format. Since this is highly dependent on the data set size, but also on the frequency of the desired prediction, we considered the interval $[1, min(x, 30 * \text{length(dataset)}/100 - 1)]$, where $x = 15$ for non-interpolated datasets and $x = 100$ for the interpolated ones. This limit was necessary for the optimization process because a large window size produces fewer predictions which misleadingly leads to smaller cumulated errors, however, the purpose of the model is to come up with predictions relying on as few known values as possible.

- *learning_rate*: finding the right value for the learning rate can significantly improve the accuracy of the model. A learning rate too big might lead to poor performance since the algorithm makes leaps too large while searching for the optimal value, whereas a small learning rate slows down the execution time of the training process. For the search space, we chose the interval $[1e-3, 1e-1]$.
- *epochs*: although not specific to Recurrent neural networks (RNNs), the number of epochs to train a model is an important key factor. The more epochs, the better for the model, but too many epochs can often lead to overfitting. Each model corresponding to a non-interpolated data set was trained for 150 epochs, while the interpolated ones were limited to 25. The final epoch number was hand-picked by observing the evolution of the loss at each epoch

As described in Section 2.2.2, the *suggest* APIs from *Optuna* framework are used to define the search space for each selected hyperparameter. Each study ran 50 trials since no significant improvement in the overall score was observed after that.

The objective function defined will create, compile, fit and evaluate the model using the training data set 5 times for each set of considered hyperparameters. This is done to ensure that the optimized hyperparameters produce a more stable model. After the optimization, we obtained the configurations from Table 5 with the corresponding scores. Mean results are again computed for 5 individual runs using the same configuration.

We mention the fact that all optimizations were executed using the free environment offered by Google Colab, having a GPU-accelerated runtime, making this accessible to everybody.

We notice that for the majority of the datasets, the best results are obtained via CVS. However, the other two strategies are also of considerable importance. For all datasets studied, the values obtained for CHS and CVS are rather close, while the results obtained for FS are higher. Even so, FS is a viable strategy for its easy way of usage. Moreover, it is worth noticing that all results obtained with the three strategies are at least 50% better than the results obtained without any interpolation strategy.

4. Results and discussions

We present the graphics of the predictions obtained via the LSTM model for our Weather data set, for the three proposed strategies and the initial data which is not interpolated. The results emphasize the advantages brought to the predictions.

For the manufactured data set which contains the daily maximum temperatures, the results are presented in Figs. 33–35. It can be observed that for the initial data, the LSTM does not perform well enough. However, for the interpolated datasets, the results are visibly better, with the best result being obtained for CHS. All the strategies provide better results and this is motivated by the fact that more training data provides better ML solutions. Thus, the importance of the proposed preprocessing strategies is supported and emphasized.

The results for both the maximum daily temperature (considered for example purposes) and the hourly data set with 17 interpolation points, see Figs. 37–39, prove that fractal interpolation brings a significant upgrade to the model as it improves visibly the results of the predictions when compared to the results obtained for the initial datasets (results that are presented in Figs. 32 and 36). Even though 17 interpolation points provide good theoretical results, we should also practically test the predictions by considering the hourly data set with 5 interpolation points to simulate the 10-min data recorded by the sensor.

Thus, we consider the entries from the Weather data set from 01/09/21, 00:00 to 08/09/21, 00:00. For this data set, we extract the hourly temperatures data and use fractal interpolation with *n_interpolation* = 5 according to the three strategies proposed (CHS, CVS, FS) to simulate 10-min data.

We can observe from Table 6 that while the hourly data without any interpolation provides visibly worse results (as expected) than the predictions with initial 10-min data, the case is the opposite when interpolating the hourly data set to simulate 10-min data with either of the three strategies. Thus, it can be observed that extracting hourly data

Table 6
Comparison between interpolated 10-min data and original data predictions.

Data set	Linear regression	Hidden layers	Input data points	Epochs	Learning rate	Train RMSE	Test RMSE
Hourly Temperatures Over a Week	None	48	8	60	0.008	0.1111	0.1383
	CHS	30	94	15	0.03	0.0415	**0.0462**
	CVS	61	94	10	0.02	0.0443	0.0474
	FS	34	99	17	0.016	0.0488	0.0494
Temperature Every 10 min	None	55	32	25	0.016	0.0523	0.0541

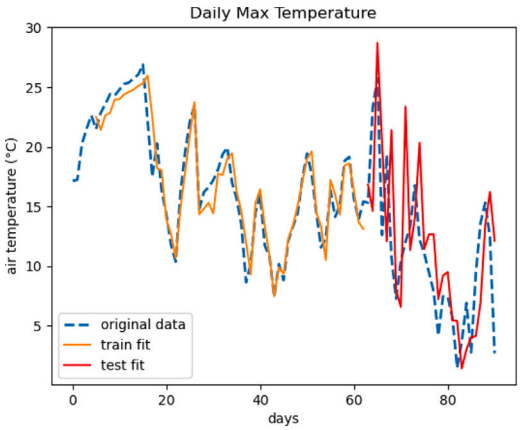


Fig. 32. Prediction for daily maximum data without interpolation.

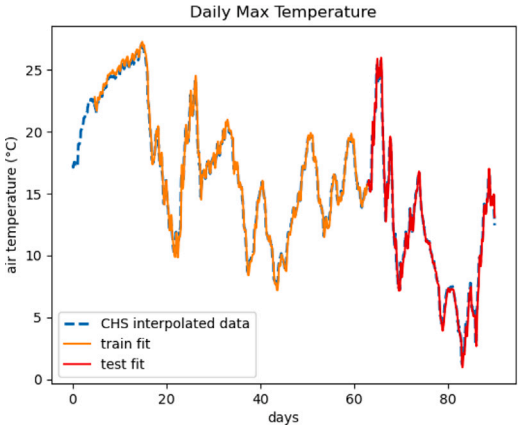


Fig. 33. Prediction for daily maximum data with CHS.

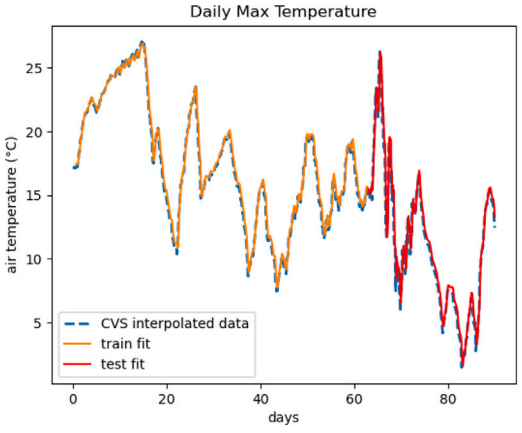


Fig. 34. Prediction for daily maximum data with CVS.

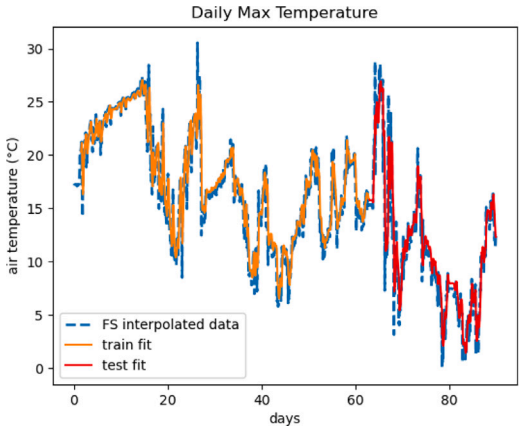


Fig. 35. Prediction for daily maximum data with FS.

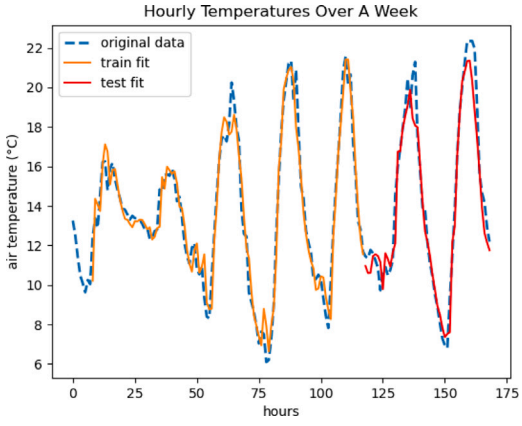


Fig. 36. Prediction for hourly data without interpolation.

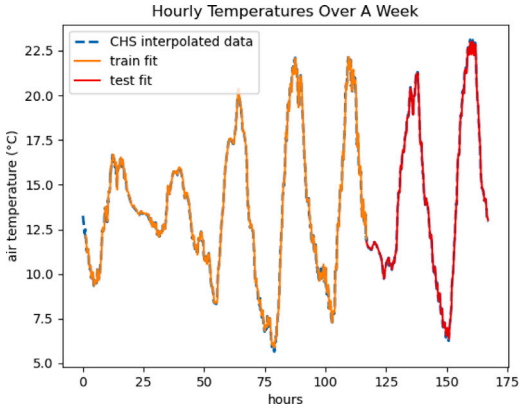


Fig. 37. Prediction for hourly data without CHS.

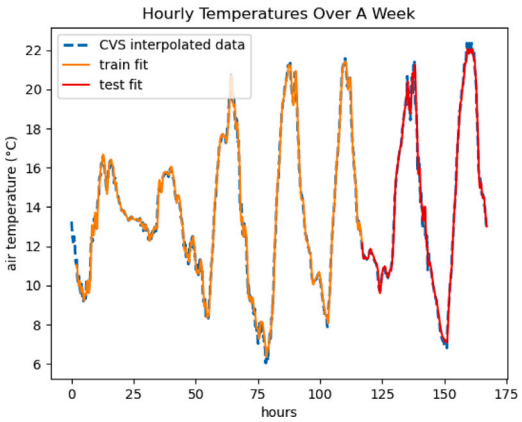


Fig. 38. Prediction for hourly data with CVS.

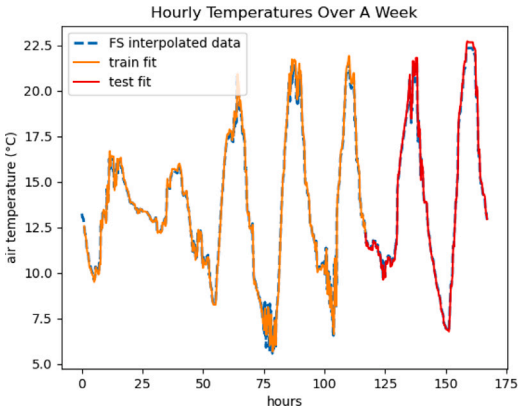


Fig. 39. Prediction for hourly data with FS.

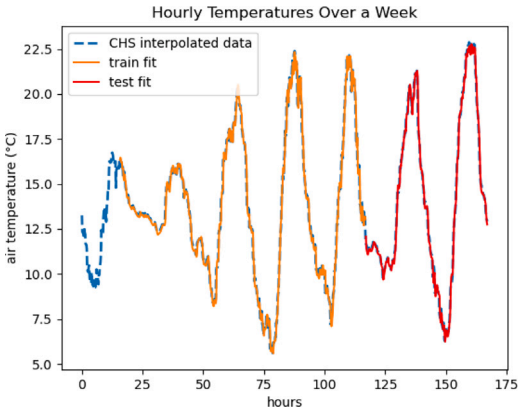


Fig. 40. Prediction for hourly data set interpolated with CHS, n_interpolation = 5.

and constructing artificial fractal interpolation points for 10-min data is a better strategy for prediction. The result is surprising and requires further testing on various weather datasets, but the current results obtained for the considered data set and the consistent differences between the test RMSE results, see Table 6, allow us to be optimistic about the performances of our strategies on real-world data (especially meteorological datasets) (see Figs. 40 and 41).

Artificial Neural Networks (ANN) and machine learning have proven to be efficient tools for predicting meteorological data. These could be significant for predicting data for smaller local areas since some meteorological processes are too small-scale or too complex to be explicitly included in classical numerical weather prediction models.

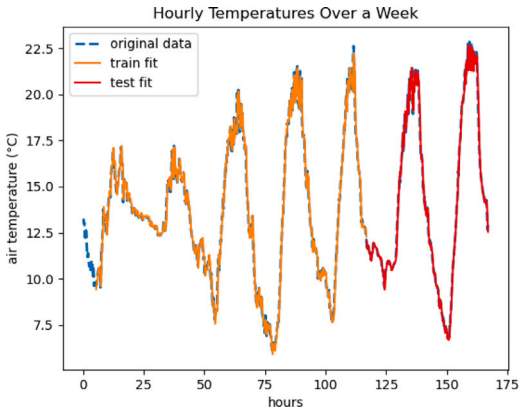


Fig. 41. Prediction using the original 10-min entry data set.

Thus, our study proposes a recent idea of using fractal interpolation tools for preprocessing data before feeding the data to an ML algorithm, in our case, an LSTM algorithm.

Without any doubt, this approach has some limitations, and there are opportunities to improve this study.

In order to provide a comprehensive coverage of the interpolation step, the study proposes three techniques. The persistence of certain aspects of the data is not guaranteed by the stop condition from the CHS method, although it assures that the Hurst exponent for the interpolated data is sufficiently near to the original Hurst value. This inspired us to develop novel interpolation techniques guaranteeing the preservation of specific data features. The final approach, namely FS, similarly focuses on maintaining these characteristics while also improving in terms of time complexity.

Furthermore, future experiments could be completed in order to assess the performance of the proposed strategies in relation to outliers, but also further testing on various weather datasets from different regions needs to be performed. In addition, although the LSTM modelling obtained is sufficient for the selected datasets, a more complex model can be developed to achieve better accuracy results.

5. Conclusions

The results obtained in this study confirm the relevance and extend the applications of fractal interpolation as a time series augmentation technique. The three proposed strategies involve optimizing the vertical scaling factor and the size of the fractal interpolation subset to generate relevant data in the context of the prediction optimization problem. Depending on the source domain and data pattern, we were able to identify the appropriate interpolation strategy in order to improve predictions. As a result, for all considered datasets, the current approach improved accuracy prediction results between 50% and 89% over the base case where the raw data was used.

By using the meteorological dataset of temperatures recorded in Braşov, we were able to show that the three proposed strategies can also be used independently of a prediction model in order to obtain data simulating a higher sampling rate than the maximum capacity of the sensor, with an average error of maximum ± 0.49. Numerical weather prediction models such as those described in Malardel (2019), based on fluid dynamics and thermodynamic equations, could also benefit from this data enrichment. Although they do not outperform solutions based on artificial neural networks, these models can perform well for short time intervals (up to 5 days) when sufficient data are available.

We have highlighted the need to use machine learning modelling in this study. In addition, it is possible to go even further with the results obtained so that a relevant prediction on meteorological data also implies a focus on process optimization in precision agriculture,

early detection of extreme weather phenomena, and local and global environmental understanding.

Our outcomes extend the current results existing in the literature and contribute significantly to research dedicated to data augmentation and data preprocessing, as well as enhancing machine learning prediction models. Moreover, our results provide a significant answer to the question of refining data prediction based on data recorded at larger intervals.

CRediT authorship contribution statement

Alexandra Băicoianu: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Cristina Gabriela Gavrilă:** Conceptualization, Methodology, Software, Validation, Formal analysis, Visualization, Writing – original draft, Writing – review & editing. **Cristina Maria Păcurar:** Conceptualization, Methodology, Validation, Writing - Original Draft,Writing – review & editing, Visualization. **Victor Dan Păcurar:** Conceptualization, Validation, Resources, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Bajahzar, A., Guedri, H., 2019. Reconstruction of fingerprint shape using fractal interpolation. Int. J. Adv. Comput. Sci. Appl. 10 (5), http://dx.doi.org/10.14569/IJACSA.2019.0100514.

Bandara, K., Hewamalage, H., Liu, Y., Kang, Y., Bergmeir, C., 2021. Improving the accuracy of global forecasting models using time series data augmentation. Pattern Recognit. 120, http://dx.doi.org/10.1016/j.patcog.2021.108148.

Barnsley, M., 2012. Fractals Everywhere, third ed. Dover Publications.

Bélisle, E., Huang, Z., Le Digabel, S., Gheribi, A.E., 2015. Evaluation of machine learning interpolation techniques for prediction of physical properties. Comput. Mater. Sci. 98, 170–177. http://dx.doi.org/10.1016/j.commatsci.2014.10.032, URL: https://www.sciencedirect.com/science/article/pii/S092702561400706X.

Bergmeir, C., Hyndman, R.J., Benítez, J.M., 2016. Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. Int. J. Forecast. 32 (2), 303–312. http://dx.doi.org/10.1016/j.ijforecast.2015, URL: https://ideas.repec.org/a/eee/intfor/v32y2016i2p303-312.html.

Bouboulis, P., Dalla, L., Drakopulos, V., 2006. Image compression using recurrent bivariate fractal interpolation surfaces. Int. J. Bifurcation Chaos 16 (07), 2063–2071. http://dx.doi.org/10.1142/S0218127406015908, arXiv:https://doi.org/10.1142/S0218127406015908.

Chai, X., Tang, G., Wang, S., Lin, K., Peng, R., 2021. Deep learning for irregularly and regularly missing 3-D data reconstruction. IEEE Trans. Geosci. Remote Sens. 59, 6244–6265, URL: https://api.semanticscholar.org/CorpusID:261340870.

Chen, C.-J., Cheng, S.-C., Huang, Y.M., 2011. The reconstruction of satellite images based on fractal interpolation. Fractals 19 (03), 347–354. http://dx.doi.org/10.1142/S0218348X11005385, arXiv:https://doi.org/10.1142/S0218348X11005385.

FaostatDocs, 2022. Food and agriculture data. URL: http://www.fao.org/faostat/. (accessed 7 March 2022).

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.A., 2018. Data augmentation using synthetic data for time series classification with deep residual networks. In: AALTD'18 Workshop in ECML/PKDD.

Forestier, G., Petitjean, F., Dau, H.A., Webb, G.I., Keogh, E., 2017. Generating synthetic time series to augment sparse datasets. In: 2017 IEEE International Conference on Data Mining. ICDM, pp. 865–870. http://dx.doi.org/10.1109/ICDM.2017.106.

Garcia, S., Luengo, J., Herrera, F., 2014. Data Preprocessing in Data Mining. In: Intelligent Systems Reference Library, Springer.

Gowrisankar, A., Priyanka, T.M.C., Banerjee, S., 2022. Omicron: a mysterious variant of concern. Eur. Phys. J. Plus 137, 100.

Guennec, A.L., Malinowski, S., Tavenard, R., 2016. Data augmentation for time series classification using convolutional neural networks. URL: https://api.semanticscholar.org/CorpusID:3907864.

Hutchinson, J., 1981. Fractals and self-similarity. Indiana Univ. Math. J. 30, 713–747.

Iwana, B.K., Uchida, S., 2021. An empirical survey of data augmentation for time series classification with neural networks. PLOS ONE 16 (7), 1–32. http://dx.doi.org/10.1371/journal.pone.0254841.

Jia, Y., Ma, J., 2017. What can machine learning do for seismic data processing? An interpolation application. Geophysics 82, URL: https://api.semanticscholar.org/CorpusID:67218941.

Kaggle, 2022. High-quality public datasets. URL: https://www.kaggle.com/. (accessed 7 March 2022).

Kamycki, K., Kapuscinski, T., Oszust, M., 2020. Data augmentation with suboptimal warping for time-series classification. Sensors 20 (1), http://dx.doi.org/10.3390/s20010098, URL: https://www.mdpi.com/1424-8220/20/1/98.

Kang, Y., Hyndman, R.J., Li, F., 2019. GRATIS: GeneRAting TIme Series with diverse and controllable characteristics. Stat. Anal. Data Min.: ASA Data Sci. J. 13, 354–376, URL: https://api.semanticscholar.org/CorpusID:71146933.

Lee, S.W., Kim, H.Y., 2020. Stock market forecasting with super-high dimensional time-series data using ConvLSTM, trend sampling, and specialized data augmentation. Expert Syst. Appl. 161, 113704, URL: https://api.semanticscholar.org/CorpusID:225041816.

Malardel, S., 2019. Weather forecasting models. URL: https://www.encyclopedie-environnement.org/en/air-en/weather-forecasting-models/. (accessed 19 April 2022).

Manousopoulos, P., Drakopoulos, V., Theoharis, T., 2008. Curve fitting by fractal interpolation. In: Gavrilova, M.L., Tan, C.J.K. (Eds.), Transactions on Computational Science I. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 85–103. http://dx.doi.org/10.1007/978-3-540-79299-4_4.

Manousopoulos, P., Drakopoulos, V., Theoharis, T., 2011. Parameter identification of 1D recurrent fractal interpolation functions with applications to imaging and signal processing. J. Math. Imaging Vision 40 (2).

May, M., 1996. Fractal image compression. Am. Sci. 84 (5).

Mazel, D., Hayes, M., 1992. Using iterated function systems to model discrete sequences. IEEE Trans. Signal Process. 40 (7), 1724–1734. http://dx.doi.org/10.1109/78.143444.

Meijering, E., 2002. A chronology of interpolation: from ancient astronomy to modern signal and image processing. Proc. IEEE 90 (3), 319–342. http://dx.doi.org/10.1109/5.993400.

Navascués, M., 2010. Reconstruction of sampled signals with fractal functions. Acta Appl. Math. 110.

Ni, L.-P., Ni, Z.-W., Gao, Y.-Z., 2011. Stock trend prediction based on fractal feature selection and support vector machine. Expert Syst. Appl. 38 (5), 5569–5576. http://dx.doi.org/10.1016/j.eswa.2010.10.079, URL: https://www.sciencedirect.com/science/article/pii/S0957417410012236.

Oh, C., Han, S., Jeong, J., 2020. Time-series data augmentation based on interpolation. In: FNC/MobiSPC. URL: https://api.semanticscholar.org/CorpusID:222112073.

Optuna-ReadTheDocs, 2023a. Optuna samplers - TPESampler docs. URL: https://optuna.readthedocs.io/en/stable/reference/samplers/generated/optuna.samplers.TPESampler.html#optuna.samplers.TPESampler. (accessed 2 October 2023).

Optuna-ReadTheDocs, 2023b. Optuna samplers docs. URL: https://optuna.readthedocs.io/en/stable/reference/samplers/index.html. (accessed 2 October 2023).

Păcurar, C.-M., Necula, B.-R., 2020. An analysis of COVID-19 spread based on fractal interpolation and fractal dimension. Chaos Solitons Fractals 139, 110073. http://dx.doi.org/10.1016/j.chaos.2020.110073, URL: https://www.sciencedirect.com/science/article/pii/S0960077920304707.

Raubitzek, S., Neubauer, T., 2021. A fractal interpolation approach to improve neural network predictions for difficult time series data. Expert Syst. Appl. 169, 114474. http://dx.doi.org/10.1016/j.eswa.2020.114474, URL: https://www.sciencedirect.com/science/article/pii/S0957417420311234.

Wang, H.-Y., Li, H., Shen, J.-Y., 2019. A novel hybrid fractal interpolation-Svm model for forecasting stock price indexes. Fractals 27 (4), http://dx.doi.org/10.1142/S0218348X19500555, 1950055.

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H., 2021. Time series data augmentation for deep learning: A survey. In: 30th International Joint Conference on Artificial Intelligence. IJCAI 2021.

Wu, Y., et al., 2020. Using linear interpolation to reduce the training samples for regression based visible light positioning system. IEEE Photonics J. 12 (2), 1–5.

Yadav, O.P., Ray, S., 2021. A novel method of preprocessing and modeling ECG signals with Lagrange–Chebyshev interpolating polynomials. Int. J. Syst. Assur. Eng. Manag. 12 (3), 377–390. http://dx.doi.org/10.1007/s13198-021-01077-, URL: https://ideas.repec.org/a/spr/ijsaem/v12y2021i3d10.1007_s13198-021-01077-z.html.

Yakuwa, F., Dote, Y., Yoneyama, M., Uzurabashi, S., 2003. Novel time series analysis and prediction of stock trading using fractal theory and time delayed neural network. In: SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483). Vol. 1, pp. 134–141. http://dx.doi.org/10.1109/ICSMC.2003.1243804.

Zhai, M.-Y., Fernández-Martínez, J.L., Rector, J.W., 2011. A new fractal interpolation algorithm and its applications to self-affine signal reconstruction. Fractals 19 (03), 355–365. http://dx.doi.org/10.1142/S0218348X11005427, arXiv:https://doi.org/10.1142/S0218348X11005427.

Zhang, Y., Fan, Q., Bao, F., Liu, Y., Zhang, C., 2018. Single-image super-resolution based on rational fractal interpolation. IEEE Trans. Image Process. 27 (8), 3782–3797. http://dx.doi.org/10.1109/TIP.2018.2826139.