

Proceeding Paper

Reconstructed Phase Spaces and LSTM Neural Network Ensemble Predictions[†]

Sebastian Raubitzek * and Thomas Neubauer

Information and Software Engineering Group, Institute of Information Systems Engineering, Faculty of Informatics, TU Wien, Favoritenstrasse 9-11/194, 1040 Vienna, Austria; thomas.neubauer@tuwien.ac.at

* Correspondence: sebastian.raubitzek@tuwien.ac.at

† Presented at the 8th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 27–30 June 2022.

Abstract: We present a novel approach that combines the concept of reconstructed phase spaces with neural network time-series predictions. The presented methodology aims to reduce the parametrization problem of neural networks and improve autoregressive neural network time-series predictions. First, the idea is to interpolate a dataset based on its reconstructed phase space properties and then filter an ensemble prediction based on its phase space properties. The corresponding ensemble predictions are made using randomly parameterized LSTM (Long Short-Term Memory) neural networks. These neural networks then produce a multitude of auto-regressive predictions, which are then filtered to achieve a smooth reconstructed phase space trajectory. Thus, we can circumvent the problem of parameterizing the neural network for each dataset individually. Here, the interpolation and the ensemble prediction aim to produce a smooth trajectory in a reconstructed phase space. The best results are compared to a single hidden layer LSTM neural network and benchmark results from the literature. The results show that the baseline predictions are outperformed for all three discussed datasets, and one of the benchmark results from the literature is bested by the presented approach.



Citation: Raubitzek, S.; Neubauer, T. Reconstructed Phase Spaces and LSTM Neural Network Ensemble Predictions. *Eng. Proc.* **2022**, *18*, 40. <https://doi.org/10.3390/engproc2022018040>

Academic Editors: Ignacio Rojas, Hector Pomares, Olga Valenzuela, Fernando Rojas and Luis Javier Herrera

Published: 25 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: phase space reconstruction; LSTM; neural networks; ensemble prediction; stochastic interpolation

1. Introduction

The rise of artificial intelligence, i.e., machine learning and deep learning, motivates many researchers to perform predictions and analyses based on historical data using these methods, rather than employing mechanistic expert models. The reason for making predictions in the first place is to answer important questions, e.g., future population estimates, predicting epileptic seizures, and estimating future stock market prices. The outcomes of these predictions are encouraging, e.g., solar radiation can be predicted using machine learning methods [1].

One reason for machine learning's poor performance is an overall lack of data. A means of overcoming the lack of data for time-series analysis approaches is to employ an interpolation technique to increase the amount of data. Here, one can choose from many different techniques, such as polynomial, fractal [2], or stochastic interpolation methods [3]. In this article, we use an improved version of the Brownian multi-point bridges developed by [3], which is discussed and validated in detail in ref. [4]. For simplicity, we refer to this method as PhaSpaSto interpolation, which is an abbreviation for **phase space trajectory smoothing stochastic interpolation**.

Next, when it comes to the autoregressive prediction of time-series data, we want to consider the properties of the reconstructed phase space trajectories of a given set of time-series data, i.e., we want the reconstructed phase space trajectory of our prediction to be as smooth as possible.

Thus, we want to find out to what extent the idea of the reconstructed phase spaces of time-series data can be used to improve neural network time-series predictions. For this reason, we present the following scheme, depicted in Figure 1. We first interpolate a given time series using the discussed PhaSpaSto interpolation (Section 4); next, we employ the randomly parameterized neural networks developed in ref. [5] (Section 6), thus generating a multitude of different autoregressive predictions for each set of time-series data. Finally, we filter these predictions based on the smoothness of their reconstructed phase space trajectories, i.e., we want to keep only the smoothest phase space trajectories (Section 6.2).

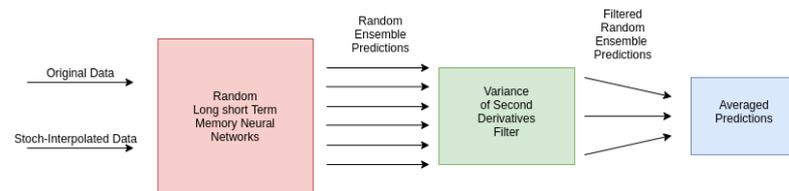


Figure 1. Schematic depiction of the filtering process. The whole pipeline is applied, first, to the original non-interpolated data, and second, to the stoch.-interpolated data set.

This article is structured as follows. The current section, i.e., Section 1, provides a brief introduction and explains the developed scheme. Section 2 lists related work and briefly describes the connections to this article. Section 3 discusses the idea of reconstructed phase spaces and introduces the used terminology and notations. Next, Section 4 introduces the employed stochastic interpolation method, whereas Section 6 describes the employed neural network approach for autoregressive time-series prediction and the prediction filter. All datasets are described and plotted with their interpolation and corresponding phase space reconstructions in Section 5. Section 7 gives all prediction results. Section 8 concludes this article and gives ideas for future research.

2. Related Work

The presented research is mainly motivated by the findings of [2,4–6]. This section will briefly describe the mentioned publications, list them chronologically, i.e., by their publication date, and put them into context.

- Ref. [7]: This publication presents a method to determine if images are blurry. For this purpose, the second derivatives of grey-scale images are taken, and the corresponding variance over all pixels is analyzed. This concept is used in the presented article. We adapted the idea of variances of second derivatives, which is discussed in Section 4.2.
- Ref. [3]: This publication presents a novel stochastic interpolation technique where the idea of a Brownian bridge, i.e., a constrained fractional Brownian motion (fBm), is extended to more than two points, i.e., to *multi-point fractional Brownian bridges*. This method is the basis for the employed interpolation techniques and provides the population of random interpolations for the genetic algorithm.
- Ref. [2]: In this publication, a fractal interpolation to interpolate univariate time-series data is presented. This research suggests that different interpolation methods for univariate time-series data may yield predictions of different quality. Thus, as presented here, employing an attractor-based interpolation is an obvious next step compared to a fluctuation-based interpolation.
- Ref. [5]: This publication is a continuation of [2]. The fractal interpolation and LSTM neural network approach is continued as ensembles of predictions. Randomly parameterized LSTM neural networks are generated from non-, linear-, and fractal-interpolated data. Afterward, these predictions are filtered based on their signal complexities. Contrary to this publication, we test LSTM neural network predictions of stochastically interpolated data.
- Ref. [4]: This publication validates a stochastic interpolation based on the smoothness of reconstructed phase space trajectories and Brownian Bridges, i.e., the PhaSpaSto

interpolation. The basic idea is to filter/improve a multitude of stochastic interpolations of the same time-series data using a genetic algorithm and the variance of second derivatives along a reconstructed phase space trajectory to generate smooth phase space embeddings.

The interpolation technique developed in this paper is briefly described in Section 4 and used to improve the presented predictions.

3. Phase Space Reconstruction

First, we need to introduce the concept of reconstructed phase spaces [8,9].

We estimate a phase space embedding for all data under study. To find a suitable phase space embedding, one has to determine two parameters, the embedding dimension d_E and the time delay τ .

To estimate the time delay τ , i.e., the delay between two consecutive time steps, we use the method based on the average information between two signals [10].

To estimate the embedding dimension d_E , we use the algorithm of false nearest neighbors [11].

The phase space embedding for a given signal $[x_1, x_2, \dots, x_n]$, thus, is

$$\vec{y}(i) = [x_i, x_{i+\tau}, \dots, x_{i+(d_E-1)\tau}] \quad (1)$$

and a corresponding three-dimensional phase space embedding, thus, is

$$\vec{y}(i) = [x_i, x_{i+\tau}, x_{i+2\tau}] \quad (2)$$

4. PhaSpaSto Interpolation

The used interpolation technique consists of two parts: first, the multi-point fractional Brownian Bridges from [3], and second, a corresponding genetic algorithm choosing the best parts of the so-created fractional Brownian bridges.

4.1. Multi-Point Fractional Brownian Bridges

The employed genetic algorithm is fueled by a population of stochastically interpolated time-series data—in our case, multi-point fractional Brownian bridges. To generate these stochastically interpolated time-series data, multi-point fractional Brownian bridges [3] were used. Thus, we briefly summarize this approach.

We consider a Gaussian random process $X(t)$, whose covariance is defined as $C(t, t') = \langle X(t)X(t') \rangle$. In the following, we focus on fractional Brownian motion where the covariance is given according to $\langle X(t)X(t') \rangle = \frac{1}{2}(t^{2H} + t'^{2H} - |t - t'|^{2H})$, where H is the Hurst exponent. To elucidate our interpolation scheme, we first define a so-called fractional Brownian bridge [12,13], which is a construction of fBm starting from 0 at $t = 0$ and ending at X_1 at $t = t_1$, i.e.,

$$X^B(t) = X(t) - (X(t_1) - X_1) \frac{\langle X(t)X(t_1) \rangle}{\langle X(t_1)^2 \rangle} \quad (3)$$

This construction ensures that $X^B(t_1) = X_1$, which is also depicted in Figure 1. This single bridge can now be generalized to an arbitrary number of (non-equidistant) prescribed points X_i at t_i by virtue of a multi-point fractional Brownian bridge [3]

$$X^B(t) = X(t) - (X(t_i) - X_i) \sigma_{ij}^{-1} \langle X(t)X(t_j) \rangle, \quad (4)$$

where $\sigma_{ij} = \langle X(t_i)X(t_j) \rangle$ denotes the covariance matrix. Furthermore, we imply summation over identical indices. The latter linear operation on the Gaussian random process $X(t)$ ensures that the bridge takes on exactly the values X_k at t_k , which can be seen from $X^B(t_k) = X(t_k) - (X(t_i) - X_i) \sigma_{ij}^{-1} \sigma_{kj} = X(t_k) - (X(t_i) - X_i) \delta_{ik} = X_k$, where δ_{ik} denotes

the Kronecker delta. Hence, this method allows for the reconstruction of a sparse signal, where small-scale correlations are determined by the choice of the Hurst exponent H .

4.2. Genetic Algorithm

We build a simple genetic algorithm to find the best possible interpolation given the data's phase space reconstruction using Taken's theorem. We want our reconstructed phase space curve to be as smooth as possible and thus define the trajectory's fitness as follows.

The basic idea is to use a concept from image processing, i.e., the blurriness of a picture, and apply it to phase space trajectories. We want our trajectory to be as blurry, i.e., as smooth, as possible. In image processing, the blurriness is determined via second-order derivatives of grey-scale images at each pixel [7]. We employ this concept, but instead of using it at each pixel, we calculate the variance of second-order derivatives along our phase space trajectories. Similar to the idea from image processing, where the low variance of second-order derivatives implies more blurriness, curves with a low variance of second-order derivatives exhibit comparatively smooth trajectories. The reason here is intuitively apparent: whereas curves with a high variance of second-order derivatives have a range of straight and pointy sections, curves with a low variance of second-order derivatives have a similar curvature along the trajectory and thus are smoother. Hence, in order to guarantee smoothness along the trajectory, we want this variance to be as low as possible, which thus is our loss L . Concluding, our fitness is maximal when our loss L is minimal.

Again, we start with the phase space vector and the corresponding embedding dimension d_E and time delay τ (see Section 3) of each signal as

$$\vec{y}(i) = [x_i, x_{i+\tau}, \dots, x_{i+(d_E-1)\cdot\tau}] \quad (5)$$

Thus, we have one component for each dimension of the phase space. Consequently, we can write the individual components as

$$y_j(i) = x_{i+(j-1)\cdot\tau} \quad (6)$$

where $j = 1, 2, \dots, d_E$. We then take the second-order finite difference central derivative of a discrete function [14]

$$u_j''(i) = x_{i+(j-1)\cdot\tau+1} - 2x_{i+(j-1)\cdot\tau} + x_{i+(j-1)\cdot\tau-1} \quad (7)$$

at each point, and for each component. Next, we add up all the components as

$$u''(i) = \sqrt{\sum_{j=1}^{d_E} u_j''(i)^2} \quad (8)$$

Then, finally, we use the variance of the absolute values of second derivatives along the phase space curve as our loss L of a phase space trajectory:

$$L = \text{Var}_i[u''(i)] \quad (9)$$

The employed genetic algorithm consists of the following building blocks.

A candidate solution is an interpolated time series using a random Hurst exponent $H \in (0; 1)$. The corresponding population of candidates is, e.g., 1000 of these stochastically interpolated time-series data with random Hurst exponents. A population of interpolated time-series data is generated using the multi-point Brownian bridges such that, for each member of the population, a random Hurst exponent with $H \in (0; 1)$ is chosen, which then defines the interpolation of the member of the population. After generating the population, all members are sorted with respect to their fitness, i.e., the lower the loss L , the better an interpolation is. The mating is implemented such that only the best 50%, with respect to fitness, can mate to produce new offspring. The mating is done such that, for every gene,

i.e., each interpolation between two data points, there is a 50:50 chance to inherit it from either one of the parents. The mutation was implemented such that, in each generation, there is a 20% chance that a randomly chosen interpolated time series is replaced with a new interpolated time series within a corresponding randomly chosen new Hurst exponent. Moreover, we implemented a criterion for aborting the program, which was fulfilled if the population fitness mean did not change for ten generations. This described procedure was then performed for 1000 generations. However, the 1000 generations were never reached, as the criterion for abortion was always triggered, and the program was ended, thus yielding the best interpolation with respect to the fitness of the phase space trajectories before reaching the 1000th generation.

5. Datasets

We chose three datasets to test and demonstrate our approach. Two of the featured datasets are from the Time Series Data Library [15] and thus are known test datasets and provide us with benchmark results, which are discussed in Section 7.4.

The third dataset is the annual maize yields in Austria, which can be obtained from <http://www.fao.org/faostat/>, accessed on 21 July 2022. This third dataset is considered the most challenging of the three datasets for two reasons. First, it is an agricultural dataset, i.e., it is affected by the weather, genetic improvements of the plants, varying fertilization strategies, etc., meaning that we will most likely not discover any reasonable seasonalities and or trends, despite the apparent increase in maize yields due to various improvements in agriculture. Second, this dataset is collected annually for all of Austria, i.e., a lot of the information contained in the dataset is lost due to the annual and regional averaging. Thus, we conclude that it will be challenging or impossible to predict annual maize yields several years ahead effectively.

5.1. Car Sales in Quebec Dataset

This is a dataset from the Time Series Data Library [15]. It depicts monthly car sales in Quebec from January 1960 to December 1968, with an overall 108 data points.

The corresponding phase space embedding, with a time delay $\tau = 1$, was detrended by subtracting a linear fit from the data and normalized such that the range of all data is between $[0, 1]$. The interpolated time series and the corresponding reconstructed phase space are depicted in Figure 2.

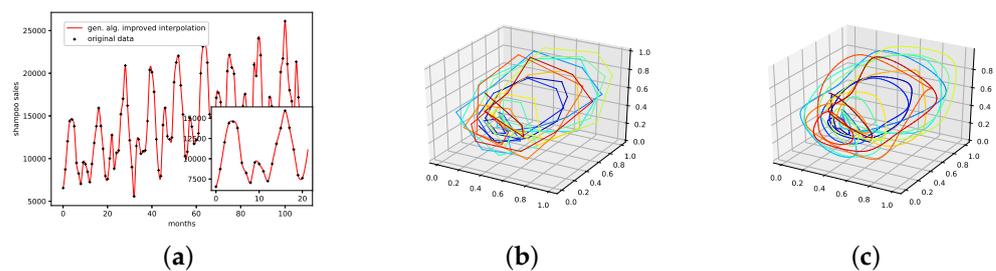


Figure 2. Time-series and attractor plots for the annual car sales in Quebec dataset. (a) Stoch. interpolated, 13 interpolation points, time-series plot; (b) non-interpolated, reconstructed attractor plot, detrended, normalized; (c) stoch.-interpolated, 13 interpolation points, reconstructed attractor plot, detrended, normalized. The rainbow colors in the phase space plots correspond to different steps in time. The spectrum starts with blue (early) and ends with red (later).

5.2. Monthly International Airline Passengers Dataset

This is a dataset from the Time Series Data Library [15]. It depicts monthly international airline passengers from January 1949 to December 1960, with an overall 144 data points, given in units of 1000.

The corresponding phase space embedding, with a time delay $\tau = 1$, was detrended by subtracting a linear fit from the data and normalized such that the range of all data is

between $[0, 1]$. The interpolated time series and the corresponding reconstructed phase space are depicted in Figure 3.

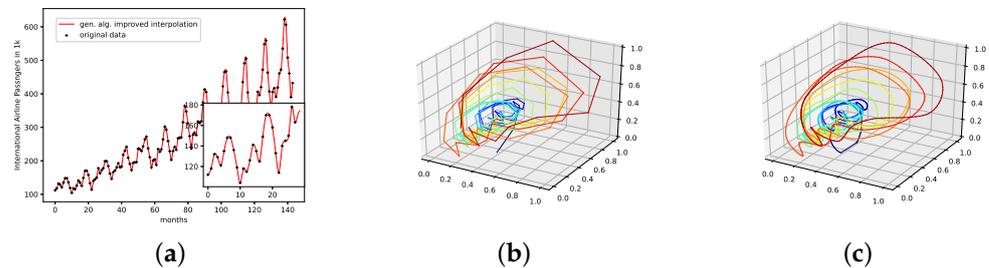


Figure 3. Time-series and attractor plots for the monthly international airline passengers dataset. (a) Stoch.-interpolated, 13 interpolation points, time-series plot; (b) non-interpolated, reconstructed attractor plot, detrended, normalized; (c) stoch.-interpolated, 13 interpolation points, reconstructed attractor plot, detrended, normalized. The rainbow colors in the phase space plots correspond to different steps in time. The spectrum starts with blue (early) and ends with red (later).

5.3. Annual Maize Yields in Austria

This is a dataset of the annual yields of maize in Austria ranging from 1961 to 2017, with an overall 57 data points. This dataset can be downloaded at <http://www.fao.org/faostat/>, accessed on 21 July 2022. The corresponding phase space embedding, with a time delay $\tau = 1$ and an embedding dimension of $d_E = 3$, was detrended by subtracting a linear fit from the data and normalized such that the range of all data is between $[0, 1]$. The interpolated time series and the corresponding reconstructed phase space are depicted in Figure 4.

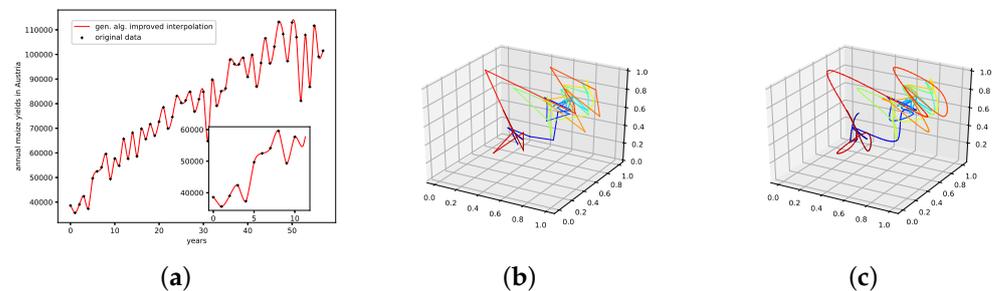


Figure 4. Time-series and attractor plots for the annual maize yields in Austria data set. (a) Stoch.-interpolated, 13 interpolation points, time-series plot; (b) non -interpolated, reconstructed attractor plot, detrended, normalized; (c) stoch.-interpolated, 13 interpolation points, reconstructed attractor plot, detrended, normalized. The rainbow colors in the phase space plots correspond to different steps in time. The spectrum starts with blue (early) and ends with red (later).

5.4. Data Preprocessing

Two steps of preprocessing were performed before forecasting the featured datasets. First, each dataset was made stationary by subtracting a linear fit. Second, each dataset was scaled to $[0.1, 0.9]$.

Finally, each dataset was split into a train and test dataset with an 80%/20% ratio.

6. LSTM Neural Network Time-Series Prediction

LSTMs are a category of recurrent neural networks (RNNs). RNNs are capable of using feedback or *recurrent* connections to cope with time-series data.

LSTMs [16] feature a component called a *memory block* to enhance their capability to model long-term dependencies. This memory block is a recurrently connected subnet containing two functional modules, i.e., the memory cell and the corresponding gates. The task of the memory cell is to remember the temporal state of the neural network. On the

other hand, the gates are responsible for controlling the information flow and consist of multiplicative units.

6.1. Randomly Parameterized Neural Networks

In this article, we are using an approach developed in [5]. The idea is to generate many randomly parameterized neural networks to build ensemble predictions based on the phase space properties of the autoregressively produced predictions. An autoregressive prediction is a one-step-at-a-time prediction, whereas old outputs are used as inputs for the next step.

These randomly parameterized neural networks feature one to five hidden LSTM layers, a *hard sigmoid* activation function in the hidden and input layers, and a rectified linear unit as the output activation function. No dropout criterion or regularization was used.

We used the following ranges for the parameters for our randomly parameterized neural network implementation.

- Number of input nodes: $1 \rightarrow \text{size of the training data} - 1$
- Number of neurons for each hidden layer: $1 \rightarrow 50$
- Batchesizes: $2 \rightarrow 128$
- Epochs: $1 \rightarrow 50$

We used LSTM architectures for this research, one can use any type of neural network cell for this approach.

6.2. Prediction Filter

The so-generated autoregressive predictions are then filtered using the criterion for smooth phase space trajectories from Section 4.2, i.e., we want the variance of second derivatives along a reconstructed phase space trajectory to be as low as possible. We thus randomly chose 1 to 10 predictions from the whole set of predictions. Next, these predictions are averaged to form an ensemble prediction. This ensemble prediction is merged with the training data. Then, the variance of second derivatives along the phase space trajectory is analyzed. This process is repeated 1 million times. The set of averaged predictions with the lowest variance of second derivatives is kept. On all plots, this procedure is referred to as `loss_rand`.

7. Experiments and Results

In this Section, we provide the experimental setup and the corresponding results.

First, each dataset was interpolated using PhaSpaSto interpolation (Section 4) with varying interpolation points. For the monthly international airline passengers and the car sales in Quebec datasets, interpolations with the following numbers of interpolation points were performed: $N_I = \{1, 3, 5, 7, 9, 11, 13\}$. For the annual maize yields in Austria, this range was changed to $N_I = \{9, 11, 13, 15\}$ to save computational resources. Further, we produced 500 randomly parameterized neural network predictions for the monthly international airline passengers and car sales in Quebec datasets. In contrast, for the annual maize yields in Austria dataset, 1000 of these predictions were produced. These multitudes of predictions were created for the non-interpolated and the interpolated datasets. As initially mentioned, the whole scheme is depicted in Figure 1.

All of these predictions were analyzed using the root mean squared error (RMSE). Here, we used the RMSE on both a normalized dataset, i.e., the dataset and the prediction are scaled to be $\in [0, 1]$ (denoted as **RMSE [0, 1]**, Equation (10)), and the regular dataset and prediction (denoted as **RMSE**, Equation (10)).

All errors for all datasets are collected in Table 1. The corresponding plots for the car sales in Quebec dataset are collected in Figure 5, the results for the monthly international airline passengers are plotted in Figure 6, and finally, the results for the annual maize

yields in Austria are depicted in Figure 7. Further, we discuss each dataset separately in the following.

$$RMSE = \left(\frac{1}{n} \sum_{i=1}^n [\hat{x}_i - x_i]^2 \right)^{\frac{1}{2}}, \quad (10)$$

Here, \hat{x}_i are the predicted values, x_i is the ground truth, and n is the number of samples.

Table 1. Results for all datasets and experiments, i.e., interpolated, non-interpolated, filtered, and unfiltered.

| Data | Approach | RMSE [0, 1] | RMSE |
|--|--|-------------|--------------|
| Car Sales in Quebec | not interpolated, unfiltered | 0.31148 | 6395.04838 |
| Car Sales in Quebec | not interpolated, filtered | 0.11635 | 1927.52494 |
| Car Sales in Quebec | stoch. interpolated, $N_I = 13$, unfiltered | 0.24762 | 5104.42560 |
| Car Sales in Quebec | stoch. interpolated, $N_I = 1$, filtered | 0.07958 | 1617.40461 |
| Monthly International Airline Passengers | not interpolated, unfiltered | 0.19676 | 101.92294 |
| Monthly International Airline Passengers | not interpolated, filtered | 0.06823 | 35.34095 |
| Monthly International Airline Passengers | stoch. interpolated, $N_I = 13$, unfiltered | 0.17180 | 86.28560 |
| Monthly International Airline Passengers | stoch. interpolated, $N_I = 9$, filtered | 0.05286 | 21.20474 |
| Annual Maize Yields Austria | not interpolated, unfiltered | 0.23536 | 18,253.10327 |
| Annual Maize Yields Austria | not interpolated, filtered | 0.16424 | 12,737.42487 |
| Annual Maize Yields Austria | stoch. interpolated, $N_I = 15$, unfiltered | 0.20499 | 15,442.32505 |
| Annual Maize Yields Austria | stoch. interpolated, $N_I = 15$, filtered | 0.14563 | 11,227.39159 |

7.1. Car Sales in Quebec Dataset

All errors for the car sales in Quebec dataset are collected in Table 1, and the corresponding plots are collected in Figure 5.

When comparing the errors for the results with and without interpolation, we see that the interpolated results are reduced, i.e., the unfiltered, interpolated results have a lower error than the unfiltered, not interpolated ones. The same is true for the filtered results.

Next, the filtered results consistently drastically outperformed the unfiltered ones. Exactly this behavior is depicted in Figure 5. The overall best result is the interpolated and filtered prediction approach, which can be seen in Figure 5d.

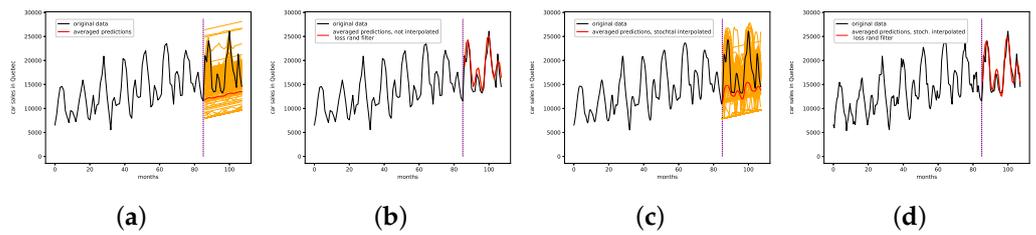


Figure 5. Autoregressive prediction results for the car sales in Quebec dataset. (a) Non-interpolated, non-filtered; (b) non-interpolated, filtered; (c) stoch. interpolated, 13 interpolation points, non-filtered; (d) stoch. interpolated, 1 interpolation point, filtered.

7.2. Monthly International Airline Passengers

All errors for the monthly international airline passengers dataset are collected in Table 1, and the corresponding plots are collected in Figure 6.

When comparing the errors for the results with and without interpolation, we see that the errors for the interpolated results are reduced, i.e., the unfiltered interpolated results have a lower error than the unfiltered, not interpolated ones. The same is true for the filtered results.

Next, the filtered results always drastically outperformed the unfiltered ones. Exactly this behavior is depicted in Figure 6. The overall best result is the interpolated and filtered prediction approach, which can be seen in Figure 6d.

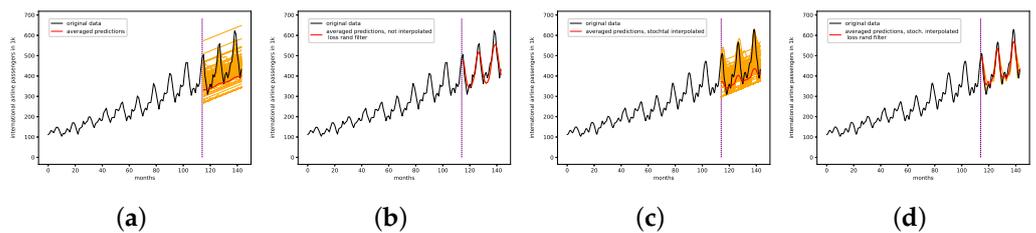


Figure 6. Autoregressive prediction results for the car sales in Quebec dataset. (a) Non-interpolated, non-filtered; (b) non-interpolated, filtered; (c) stoch. interpolated, 13 interpolation points, non-filtered; (d) stoch. interpolated, 9 interpolation points, filtered.

7.3. Annual Maize Yields in Austria Dataset

All errors for the monthly international airline passengers dataset are collected in Table 1, and the corresponding plots are collected in Figure 7.

When comparing the errors for the results with and without interpolation, we see that the interpolated results are reduced, i.e., the unfiltered, interpolated results have a lower error than the unfiltered, not interpolated ones. The same is true for the filtered results.

Next, the filtered results consistently drastically outperformed the unfiltered ones. Exactly this behavior is depicted in Figure 7. The overall best result is the interpolated and filtered prediction approach, which can be seen in Figure 7d.

This dataset is considered to be the most difficult of the three featured datasets, and although our predictions are still off, as can be seen in Figure 7d, the performed procedures, i.e., PhaSpaSto interpolation and the prediction filter, do improve the accuracy of the forecast. Further, the result depicted in Figure 7d suggests that the employed neural networks learned some inherent behavior, especially when taking into account the initial variations after the train/test cut.

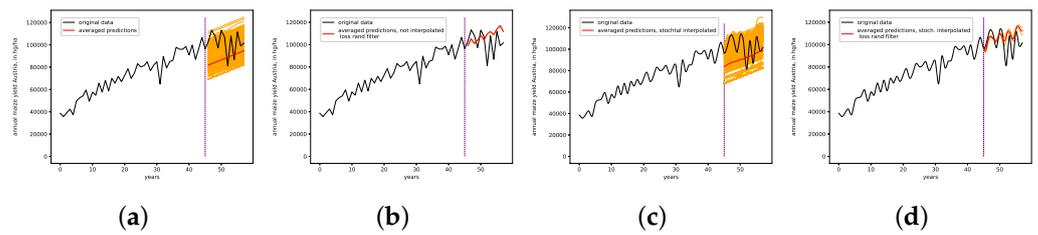


Figure 7. Autoregressive prediction results for the car sales in Quebec dataset. (a) Non-interpolated, non-filtered; (b) non-interpolated, filtered; (c) stochastic interpolated, 15 interpolation points, non-filtered; (d) stochastic interpolated, 15 interpolation points, filtered.

7.4. Benchmark and Baseline Predictions

We finally provide some baseline and benchmark results for the conducted experiments. We used an LSTM neural network with one hidden layer as a baseline prediction. Each neural network was trained with a batch size of 2 and varying epochs. Further, verbose was set to 2. For the activation of the neural network, hard_sigmoid was chosen, and the activation function of the output layer was relu. For the initialization, gloriot_uniform was used for the LSTM layer, orthogonal was used as the recurrent initializer, and gloriot_uniform for the Dense layer. For the LSTM layer, the bias was set to use_bias=True, with a corresponding bias_initializer="zeros". Further, no constraints, regularizers, or dropout criteria were used for the recurrent and the Dense layers. As an optimizer, rmsprop was used and the loss was calculated using mean_squared_error. The output node returned only one result, i.e., the next time step. The varying architectures are collected in Table 2 and the corresponding predictions are depicted in Figure 8.

Table 2. Errors for the baseline predictions for each dataset.

| Data | Architecture | RMSE [0, 1] | RMSE |
|--|---|-------------|--------------|
| Car Sales in Quebec | 20 input nodes 30 hidden layer neurons 45 training epochs | 0.08593 | 1764.38996 |
| Monthly International Airline Passengers | 20 input nodes 30 hidden layer neurons 40 training epochs | 0.05899 | 30.56100 |
| Annual Maize Yields Austria | 20 input nodes 30 hidden layer neurons 18 training epochs | 0.16617 | 12,886.99962 |

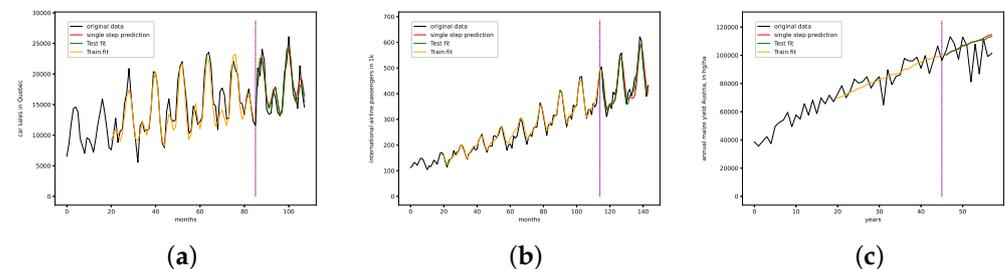


Figure 8. LSTM baseline predictions. The red line denotes the autoregressive single step-by-step prediction, which is featured in Table 2. (a) Car sales in Quebec dataset; (b) monthly international airline passengers dataset; (c) annual maize yields in Austria dataset.

The featured baseline predictions, though reasonable, are consistently outperformed by the interpolated filtered results from our main experiments—see Table 1—in terms of RMSE. Here, we want to highlight that the baseline neural network did not capture the characteristics of the annual maize yields dataset. This may be due to poor parameterization

or the fact that a single hidden layer is not sufficient to capture the dynamics of this dataset. Still, we tuned the employed neural networks by hand, and the results for the other dataset show that neural networks of these sizes are sufficient for univariate time-series data of this length. The best ensemble result, in comparison, provides a drastically improved result compared to the baseline prediction.

When it comes to benchmark results from the literature, we found the best result for the monthly international airline passengers dataset in ref. [17], with an RMSE of 13.0 for a hybrid MLP-ARIMA approach, which is superior to our baseline of 30.6 and our best ensemble result of 21.2. Thus, we conclude that our ensemble approach with the presented specifications cannot outperform state-of-the-art methods for this dataset.

For the monthly car sales in Quebec dataset, we found a comparable result in ref. [18], with an RMSE $[0, 1]$ of 0.08143 for the additive Holt–Winters method. Our baseline LSTM result for this dataset has an RMSE $[0, 1]$ of 0.08593 and our best ensemble results is at 0.07958. We conclude that our ensemble prediction is able to outperform the best results from ref. [18] for this dataset.

As far as the authors know, there is no benchmark result for the annual maize yields in Austria dataset. Thus, we stick to the previously presented baseline prediction, which is outperformed by the presented ensemble prediction.

7.5. Summary

We briefly summarize our findings and point out the main results below.

1. The presented stochastic interpolation method—for simplicity, referred to as *PhaSpaSto* interpolation—can be used to improve retrogressive neural network time-series predictions. This is supported by the findings of Table 1. Here, we can see that both the filtered and unfiltered interpolated results outperformed those without interpolation. The same is true for all filtered results, i.e., the interpolated results always outperformed the unfiltered ones. These results are depicted in Figures 5–7.
2. Filtering the multitude of predictions based on the second derivatives along their reconstructed phase space portraits drastically improved the results for all datasets. The corresponding results are, again, collected in Table 1 and Figures 5–7.
3. The presented interpolated and filtered approach outperformed the baseline and benchmark predictions for the monthly car sales in Quebec dataset, discussed in Section 7.4.

Though the interpolated and filtered ensemble approach did outperform a given baseline prediction for the monthly international airline passengers dataset, the featured benchmark prediction from the literature still outperformed our approach on this dataset.

We provide a baseline prediction for the annual maize yields in Austria dataset, which was outperformed using our interpolated and filtered ensemble approach, discussed in Section 7.4. We cannot provide a benchmark result from the literature for this dataset.

4. The employed neural network ensembles were not individually parameterized for each dataset. Instead, we filtered the predictions according to the phase space properties of each dataset. Thus, we could circumvent the problem of parameterizing neural networks.

8. Conclusions

This article presents an experiment to test the applicability of a novel interpolation method—for simplicity, abbreviated as *PhaSpaSto* interpolation—combined with randomly parameterized neural network autoregressive predictions. These predictions are then filtered using the variance of second derivatives along a reconstructed phase space trajectory to only keep forecasts that ensure a smooth phase space trajectory.

First, a given time series is interpolated using the featured interpolation method. It is forecast by generating multiple differently parameterized neural networks, each providing

an autoregressive prediction of the data under study. Finally, this multitude of predictions is filtered such that the result guarantees a smooth reconstructed phase space trajectory.

The results show that this novel approach outperforms the provided baseline predictions. Further, we were able to best a given benchmark result from the literature for one of the three discussed datasets.

The concept of reconstructed phase spaces can be applied to interpolate time series to guarantee a smooth phase space trajectory, which, in turn, improves the accuracy of our neural network predictions. Further, filtering ensemble predictions based on their phase space properties, i.e., the smoothness of their phase space trajectories, improves the presented ensemble predictions. Moreover, we can circumvent the problem of parameterizing neural networks by generating many predictions and filtering them based on their phase space properties.

Ideas for future research in this field are, e.g., to test the presented filter on other state-of-the-art ensemble approaches or to test the robustness of neural network predictions using the presented PhaSpaSto and related interpolation techniques. We further want to highlight that the proposed methodology improves the accuracy on the featured annual maize yields dataset, which the authors expected to be a challenging dataset. Thus, another idea for future research might be to specifically target challenging time-series prediction problems, such as forecasting agricultural or financial time-series data.

Author Contributions: Conceptualization, S.R.; Data curation, S.R.; Funding acquisition, T.N.; Investigation, S.R.; Methodology, S.R.; Project administration, T.N.; Resources, T.N.; Software, S.R.; Supervision, T.N.; Validation, S.R.; Visualization, S.R.; Writing—original draft, S.R.; Writing—review and editing, S.R. All authors have read and agreed to the published version of the manuscript.

Funding: The authors acknowledge the funding of the project “DiLaAg—Digitalization and Innovation Laboratory in Agricultural Sciences”, by the private foundation “Forum Morgen”, the Federal State of Lower Austria, and by the FFG; Project AI4Cropr, No. 877158.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Voyant, C.; Notton, G.; Kalogirou, S.; Nivet, M.L.; Paoli, C.; Motte, F.; Fouilloy, A. Machine learning methods for solar radiation forecasting: A review. *Renew. Energy* **2017**, *105*, 569–582. [[CrossRef](#)]
2. Raubitzek, S.; Neubauer, T. A fractal interpolation approach to improve neural network predictions for difficult time series data. *Expert Syst. Appl.* **2021**, *169*, 114474. [[CrossRef](#)]
3. Friedrich, J.; Gallon, S.; Pumir, A.; Grauer, R. Stochastic Interpolation of Sparsely Sampled Time Series via Multipoint Fractional Brownian Bridges. *Phys. Rev. Lett.* **2020**, *125*, 170602. [[CrossRef](#)] [[PubMed](#)]
4. Raubitzek, S.; Neubauer, T.; Friedrich, J.; Rauber, A. Interpolating Strange Attractors via Fractional Brownian Bridges. *Entropy* **2022**, *24*, 718. [[CrossRef](#)] [[PubMed](#)]
5. Raubitzek, S.; Neubauer, T. Taming the Chaos in Neural Network Time Series Predictions. *Entropy* **2021**, *23*, 1424. [[CrossRef](#)] [[PubMed](#)]
6. Raubitzek, S.; Neubauer, T. Combining Measures of Signal Complexity and Machine Learning for Time Series Analysis: A Review. *Entropy* **2021**, *23*, 1672. [[CrossRef](#)] [[PubMed](#)]
7. Pech-Pacheco, J.; Cristobal, G.; Chamorro-Martinez, J.; Fernandez-Valdivia, J. Diatom autofocusing in brightfield microscopy: A comparative study. In Proceedings of the 15th International Conference on Pattern Recognition, ICPR-2000, Barcelona, Spain, 3–7 September 2000; Volume 3, pp. 314–317. [[CrossRef](#)]
8. Takens, F. Detecting strange attractors in turbulence. In *Dynamical Systems and Turbulence, Warwick 1980, Lecture Notes in Mathematics*; Rand, D., Young, L.S., Eds.; Springer: Berlin/Heidelberg, Germany, 1981; Volume 898, pp. 366–381.
9. Packard, N.H.; Crutchfield, J.P.; Farmer, J.D.; Shaw, R.S. Geometry from a Time Series. *Phys. Rev. Lett.* **1980**, *45*, 712–716. [[CrossRef](#)]
10. Fraser, A.M.; Swinney, H.L. Independent coordinates for strange attractors from mutual information. *Phys. Rev. A* **1986**, *33*, 1134–1140. [[CrossRef](#)] [[PubMed](#)]

11. Rhodes, C.; Morari, M. The false nearest neighbors algorithm: An overview. *Comput. Chem. Eng.* **1997**, *21*, S1149–S1154. [[CrossRef](#)]
12. Delorme, M.; Wiese, K.J. Extreme-value statistics of fractional Brownian motion bridges. *Phys. Rev. E* **2016**, *94*, 052105. [[CrossRef](#)] [[PubMed](#)]
13. Sottinen, T.; Yazigi, A. Generalized Gaussian bridges. *Stoch. Process. Appl.* **2014**, *124*, 3084–3105. [[CrossRef](#)]
14. Quarteroni, A.; Sacco, R.; Saleri, F. *Numerical Mathematics*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2007; Volume 37. [[CrossRef](#)]
15. Hyndman, R.; Yang, Y. Time Series Data Library v0.1.0. 2018. Available online: pkg.yangzhuoranyang.com/tsdl (accessed on 1 July 2022).
16. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
17. Domingos, S.d.O.; de Oliveira, J.F.; de Mattos Neto, P.S. An intelligent hybridization of ARIMA with machine learning models for time series forecasting. *Knowl.-Based Syst.* **2019**, *175*, 72–86.
18. Shah, V. A Comparative Study of Univariate Time-Series Methods for Sales Forecasting. Master's Thesis, University of Waterloo, Waterloo, ON, Canada, 2020.