QUBAHAN ACADEMIC JOURNAL
VOL. 4, NO. 3, September 2024
https://doi.org/10.48161/qaj.v4n3a912

Multi-Step-Ahead Prediction of Chaotic Time Series: Self-Healing Algorithm for Restoring Values at Non-Predictable Points

Vasilii A. Gromov [1], Korney K. Tomashchuk [1], and Alexey A. Rukavishnikov [2*]

[1] International Laboratory for Intelligent Systems and Structural Analysis, Faculty of Computer Science, HSE University, Moscow, Russia;
[2] School of Data Analysis and Artificial Intelligence, Faculty of Computer Science, HSE University, Moscow, Russia.

Corresponding author: e-mail: rukavishnikov.hse@gmail.com.

ABSTRACT: This paper proposes a new algorithm for the multi-step-ahead prediction of chaotic time series within the framework of the clustering-based forecasting paradigm. The introduction of the concept of non-predictable points enabled to avoid the exponential growth of the prediction error as a function of the number of steps ahead, which made it possible to develop algorithms that predict many Lyapunov times (and many steps) ahead – the price for this turned out to be that some points remained non-predictable. This study proposes a **self-healing** algorithm, which is an iterative algorithm that takes the forecasts produced by the underlying prediction algorithm as input. At each iteration, the self-healing algorithm finds new possible predicted values, updates the status of the points from predictable to unpredictable, or vice versa, and calculates new single predicted values for the predictable points. This study proposes several new algorithms for calculating a single prediction value and algorithms for determining unpredictable points. Authors overviewed recent studies with larger goals and broader viewpoints on parameter estimations, relevance tracking, and predictive features. *Results*: Our research revealed that the new instruments in particular indicators increased RMSE (root mean squared error) from 0.11 to 0.06 and decreased MAPE (mean absolute percentage error) from 0.38 to 0.04. Research was conducted on the selection of parameters for the self-healing algorithm, its assessment, and the prediction quality compared to the existing prediction algorithm.

Keywords: Chaotic time series, Predicting over the prediction horizon, Many Lyapunov times, multi-step-ahead prediction, Self-healing algorithms, predictable and non-predictable values.

I. INTRODUCTION

The concept of non-predictable points [1] makes it possible to obtain a prediction for the number of steps ahead for the model and real chaotic time series, exceeding the manifold of the prediction horizon. Non-predictable points are those for which the algorithm cannot provide an adequate prediction, while predictable points are those for which reliable predictions can be made. The price for the large number of steps ahead for which we can obtain a prediction is a significant percentage (however, not reaching 100%) of non-predictable points – points at which the algorithm cannot obtain an adequate prediction. There are scientific questions regarding the prediction of multi-fuels in the work that is followed, especially with the new rate. This study proposes an algorithm for restoring values at non-predictable points from the values obtained for predictable points, in which the multi-step-ahead predictive-clustering prediction algorithms use the concept of unpredictable points. Gromov and Baranov [1] represent the zero iteration of the iterative process, during which the lacks (non-predictable points) are filled. The use of the algorithms proposed in this study made it possible to significantly reduce the number of non-predictable points while remaining within the same predictive clustering algorithm that was used for prediction. However, we present a correlation between the predictions and other registered measures of optimism. In this study, we suggest that the exception is a new prediction algorithm that can predict other signals.

Predictions of the signs of help and progress of lysine are presented using Poincaré's theorem. The high sensitivity of the trajectory (observed as a time series) to any, regardless of how small the disturbance is, is a basic characteristic of a chaotic dynamic system. This means that a small difference between the true and predictive trajectories of the

system, inevitable after the first prediction step, leads to an exponential increase in this difference over time (with an increase in the number of steps ahead for which it is necessary to obtain a prediction); the exponent here is the highest Lyapunov exponent. The latter allows for the estimation of the prediction horizon, the theoretical boundary beyond which the true and predictive trajectories diverge by an amount exceeding a given value.

In the study by Gromov and Baranov [1], within the clustering-based prediction paradigm, the concept of non-predictable points was proposed, which enables us to get away from the exponential "Lyapunov" growth of the prediction error as a function of the number of steps ahead, for which it is necessary to obtain a prediction; if we limit ourselves to only the average value of the error on predictable points, the error function does not grow, but exhibits behavior close to constant. According to the definition of the problem, it is necessary to study the proposed knowledge regarding the important properties of the prediction and their effects on the performance of the algorithm.

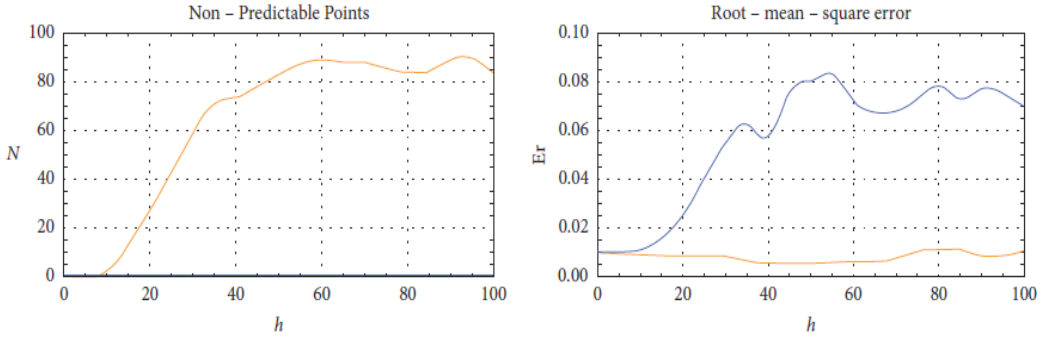


FIGURE 1. Number of unpredictable points (a) and root-mean-square deviation (b) depending on the prediction horizon for two extreme cases. The blue curves correspond to the first extreme case (intermediate non-predictable points are not highlighted); orange curves correspond to the second extreme case (intermediate non-predictable points are identified using a priori information) [1].

Figure 1 shows the characteristic dependence of the percentage of non-predictable points (left figure) and prediction errors per predictable point (right figure) for a prediction algorithm using the concept of non-predictable points (orange color) and the same algorithm that does not use it (blue color) (all dependencies correspond to the testing set). It can be seen that although the percentage of non-predictable points in the algorithm that uses the concept of non-predictable points increases with the number of steps ahead for which it is necessary to obtain a prediction (for an algorithm that does not use this concept, this percentage is naturally equal to zero), it does not reach 100%. (The problem of predicting a chaotic time series in these terms can be formulated as an optimization problem: minimizing the number of non-predictable points and minimizing the average prediction error over a set of common prediction values.). At the same time, the prediction error remains close to constant (does not grow exponentially) and is very small, in contrast to the same algorithm, which does not use the concept of non-predictable points.

The clustering-based prediction paradigm is underpinned by the Poincaré theorem for the return of trajectories [2], which guarantees the appearance of sections that are similar (but not completely identical!) to each other, in several areas. Consequently, the clustering of all sections of a time series makes it possible to identify characteristic sequences found in the observed part of the series (motifs). If the time series before the predictable point is similar to the beginning of a motif, there is a high probability that the predictable point will be close in value to the last point of the motif.

There are scientific questions about the barriers and the specific understanding of the parent's prediction that showed the progress of the revolution by following the specific opposition. Gromov and Baranov [1] introduced the concept of prediction patterns, according to which the sample for clustering is formed rather than observations whose distance between positions corresponds to a certain pattern than from sequential observations. The algorithm uses a significant number of patterns (in fact, all patterns that meet some reasonable restrictions), which enables us to obtain a set of predictions for a given predictable point (set of possible prediction values) using motifs that correspond to various t patterns. The available non-empty set of possible prediction values for a point makes it possible raises the question of its predictability, that is, the possibility of obtaining a single prediction value from multiple prediction values. For example, if 50% of the elements of the set of possible prediction values indicate a prediction value of +1 and 50% indicate a prediction value of-1, it is obvious that obtaining a single prediction value is complicated, and

such a point should be recognized as unpredictable. Moreover, the non-predictability of a series in a certain position does not mean that it is unpredictable in subsequent positions. This approach makes it possible to predict a very large number of steps ahead (in some cases, going beyond the prediction horizon); the price for this is a large percentage of non-predictable points (not reaching 100%, however).

In this study, we adopt a new framework for predicting the specific and general relationships in the pursuit of the original author. In this study, we adopt a new framework for predicting the specific and general relationships in the pursuit of the original author. This study aims to create a self-healing iterative algorithm that reduces the percentage of unpredictable points. It is expected that the zero iteration of the algorithm coincides with any prediction algorithm that assumes the ability to identify unpredictable points (see, for example, [1]). The result of its operation is a set of prediction values for a part of the section of the series of interest (predictable points). Naturally, in this case, some points remain unpredictable. In this context, we propose a new self-burning algorithm to reduce the number of non-prediction points. The main idea of the algorithm operation in iterations other than zero is as follows: if in the first iteration, the algorithm predicts a point using only points located to the left of the predictable one; in subsequent iterations, the algorithm uses points located both to the left and to the right of the predictable one. In both cases, the same algorithm is used, with the only difference being that in the zero iteration, the points predicted in the same zero iteration are used as input data, and in subsequent iterations, we use the prediction values obtained in previous iterations (admittedly, in both cases, the known values of the series can also be used). This observation suggests that the self-ignition algorithm typically increases the predicted frame relations in the global iterations. This observation suggests that the self-ignition algorithm generally increases the predicted frame relations in the global iterations. The main objectives of this study were as follows:

- Creating a self-ignition algorithm that can efficiently return values at unpredictable points.
- Minimizing the number of nonpredictive points also provides a more predictable framework for clustering.
- Enhance predictive reality management with new exclusive predictive global iterations.
- Here are the main contributions to this work:
- Development of a New Self-Burning Algorithm: This new algorithm provides a new self-burning algorithm for predicting chaotic interactions, which can efficiently return values at unpredictable points. The self-ignition algorithm in this work is necessary to increase the knowledge and accuracy of prediction, and especially to map and ignite chaotic relationships.
- Proposing New Methodologies: proposing new approaches to calculating predictive values and assigning non-predictive scores. These new methodologies also reduce the number of nonpredictive points and provide a more predictable framework for clustering. This new rate allows the detection of chaotic patterns and more predictable interactions.
- Relation of the Self-Burning Algorithm to Existing Prediction Algorithms: In this work, the self-burning algorithm suggests its relevance to existing prediction algorithms. This increases the relevance of current algorithm developments and monitoring of predictive methods in chaotic interactions.

Develop a self-healing algorithm that can effectively restore values at unpredictable points. The percentage of non-predictable points is reduced while maintaining the effectiveness of the clustering-based forecasting paradigm. Enhances prediction accuracy by iteratively refining predictions for predictable points.

The remainder of this paper is organized as follows. The first section provides a review and analysis of the literature. Section 2 presents the problem statement and Section 3 describes the proposed algorithm. Section 4 substantiates the choice of means for implementing the program for conducting research, and describes the methods for computational experiments. Section 5 presents the results of the computational experiments and a comparison with existing prediction algorithms. Finally, the conclusions are presented.

II. LITERATURE REVIEW

Despite the significant number of publications devoted to chaotic time-series prediction, here we can mention algorithms based on LSTM neural networks [3], support vector regression [4], and boosting [5] – the vast majority of them are devoted to prediction on one (few) steps ahead. The publications that make it possible to at least mitigate the exponential growth of the error and thereby enable multi-step-ahead prediction include (1) studies based on the creation of a certain prediction strategy within an existing prediction algorithm and (2) studies devoted to the development of a novel prediction algorithm, usually within a well-known strategy. Among the latter, we can

distinguish algorithms built within the machine (deep) learning paradigm from algorithms built within the clustering-based prediction paradigm.

To this end, it is necessary to present new forecasting methods under different headings and show the latest developments in forecasting relationships.

1. *Strategies for multi-step-ahead prediction*. The review article by Taieb et al. [6] analyzed various strategies for multistep-ahead prediction and identified five main classes of strategies: iterative, direct, DirRec, MIMO, and DIRMO strategies. The iterative (or recursive) strategy [7] implies that the one-step-ahead prediction algorithm is launched sequentially $\boldsymbol{h}$ times, where $\boldsymbol{h}$ is the number of steps for which it is necessary to obtain a prediction, whereas the values obtained in the previous steps are used to calculate the prediction values at the current step. The direct strategy [7] involves immediate prediction of the $\boldsymbol{h}$-th step without calculating the intermediate prediction values. The DirRec strategy [7] (Direct + Recursive) combines direct and iterative strategies: Predicted values are calculated independently by the algorithm, as in the direct strategy; however, (approximate) prediction values from previous steps are also used as input data. Within the MIMO strategy [8] (Multi-Input Multi-Output), the algorithm predicts a set of values for all points lying between the last observed point and the point at which the prediction should be obtained, rather than one value for the point at which the prediction should be obtained (distanced by $\boldsymbol{h}$ steps from the last observed point). The DIRMO strategy [9] (DIRect + miMO) divides the part of the series to be predicted into blocks and applies a MIMO-based algorithm for each block separately; thus, DIRMO combines the advantages of both strategies. Taieb et al. [6] conducted experiments on a large amount of time series data and concluded that MIMO-based algorithms are the most effective, and proposed several approaches to data processing before prediction, increasing the accuracy of the algorithms.
2. *Deep-learning methods in chaotic time series prediction*. Chandra et al. [10] compared prediction algorithms based on fuzzy, recurrent, convolutional neural networks, support vector regression, and others. The authors concluded that the encoder–decoder LSTM neural network and bidirectional LSTM neural network showed the best accuracy in both simulated and real-world time series.
3. *Clustering-based prediction.* The concept of clustering-based prediction [11-14] involves studying the dynamics in the space of z-vectors (a set of a fixed number of consecutive observations of a time series), which satisfies Takens' theorem [15] and corresponds to the same domain of the weird attractor. Gromov and Borisenko [16] proposed the concept of generalized z-vectors (a set of a fixed number of nonconsecutive observations of a time series), which made it possible to introduce the concepts of a set of possible prediction values and nonpredictable points.

An analysis of the literature reveals that in the vast majority of publications devoted to multi-step-ahead prediction of chaotic time series, there is, in one form or another, an exponential increase in the prediction error, which makes it important to construct and study methods for multi-step-ahead prediction, enabling us to avoid this exponential growth. This follow-up suggests that, in general predictive relationships, the increase in predictability determines the truth and the need for new communication methods to suggest general follow-up over prior prediction. Gromov and Baranov [1] proposed an approach to avoid exponential growth; the prediction error remained almost constant as the number of steps ahead, for which it is necessary to obtain a prediction, increased. The price for this is a large number of unpredictable points, for which the algorithm failed to obtain a prediction. In this paper, we propose an algorithm that makes it possible to fill in the values at points that remain unpredicted (self-healing algorithm).

III. PROBLEM STATEMENT

The chaotic time series $Y = \{y_1, y_2, \ldots, y_t, \ldots\}$ was considered. It is assumed that all transition processes in the corresponding dynamic system are completed, movement occurs in the vicinity of some weird attractors, and the series values are normalized. The series is divided into training Y_1 and testing Y_2 parts: $Y = Y_1 \cup Y_2, Y_1 \cap Y_2 = \emptyset$.

This study considers a modified statement of the problem of h-steps-ahead prediction problem for a (chaotic) time series, $h \in \mathbb{N}$, proposed in [1]. This problem statement assumes the possibility of the existence of unpredictable points, that is, points for which the algorithm failed to obtain a prediction.

Within this problem statement, the prediction algorithm for each point for which it is necessary to obtain a prediction, $t + h$, the set of possible prediction values is calculated $\hat{S}_{t+h} = \{\hat{y}_{t+h}^{(1)}, \ldots, \hat{y}_{t+h}^{(N_{t+h})}\}$. Here t is the last observed point in the series, h is the number of steps ahead for which we want to obtain a prediction, $\hat{y}_{t+h}^{(i)}$ is the i-th possible prediction value found by the algorithm, and N_{t+h} is the number of possible prediction values for the predictable point at position $t + h$. Within this problem statement, the following are also considered: operator ζ, which determines whether the point is predictable:

$$\hat{\zeta}(S_{t+h}) = \begin{cases} 0, \text{if the point is non} - \text{predictable} \\ 1, \text{if the point is predictable} \end{cases} \quad (1)$$

Where operator g, which calculates a single prediction value from a set of possible prediction values: $\hat{y}_{t+h} = g(S_{t+h})$.

To this end, it is necessary to think of the most modern form of management that passes through specific contexts between real appointments.

In these terms, we can formulate the prediction problem as a two-criteria optimization problem:

$$I_1 = min \sum_{t+h \in Y_2} (1 - \hat{\zeta}(S_{t+h})) \quad (2)$$

$$I_2 = min \frac{1}{|Y_2|} \sum_{t+h \in Y_2} \hat{\zeta}(S_{t+h}) \parallel g(\hat{S}_{t+h}) - y_{t+h} \parallel^2 \quad (3)$$

Here, functional I_1 minimizes the number of non-predictable points and functional I_2 minimizes the average prediction error on the predictable points.

IV. BASE FORECASTING ALGORITHM

1. MOTIF FORMATION ALGORITHM

Clustering of training samples constructed from generalized z-vectors was used to identify the motifs [13]. Generalized z-vectors are vectors composed of observations of a time series, the positions of which are separated from each other by certain distances corresponding to a certain pattern. Here the pattern is a vector of distances between observations $k_1, k_2, \ldots, k_{L-1}$, where L is the pattern length. Thus, a set of z-vectors for pattern $k_1, k_2, \ldots, k_{L-1}$ takes the form $\{(y_m, y_{m+k_1}, y_{m+k_1+k_2}, \ldots, y_{m+k_1+\ldots+k_{L-1}}) | 1 \le m < t\}$, where $t = |Y_1|$. For example, for pattern (3,2,4) a set of+ z-vectors is $(y_1, y_4, y_6, y_{10}), (y_2, y_5, y_7, y_{11}), \ldots, (y_{t-10}, y_{t-7}, y_{t-5}, y_{t-1})$ (see Figure 2). According to this definition, a transition z-vector (which is a specific set of time-series observations) is associated with patterns 1,1, \ldots, 1 (L-1) times.

In these terms, an ordinary z-vector (a set of sequential observations of a time series) corresponds to pattern 1,1…,1 $(L-1)$ times. We denote the set of all combinatorially possible patterns of length L, for which the maximum distance between observations does not exceed k_{max} as $\aleph(L, k_{max})$. For each pattern $\alpha \in \aleph(L, k_{max})$ a set of z-vectors is calculated, and motifs (cluster centers) are identified using the training set.

The clustering algorithms DBSCAN and Wishart were used in this study. The advantage of these algorithms is that they do not require a number of clusters in the input parameters. Given the definition that DBSCAN and Wishart are clustering algorithms, they also understand that a number of clusters must be set in the relational parameters. DBSCAN (density-based spatial clustering of applications with noise) is a density-based clustering algorithm. This algorithm was first proposed in 1996 [17] and is now one of the most commonly used clustering algorithms. The Wishart clustering algorithm [18], which was improved by Lapko and Chentsov [19], uses graph theory concepts and a nonparametric estimation of the probability density function r of the nearest neighbors.

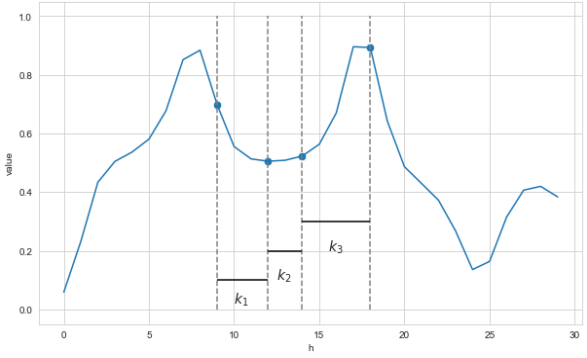


FIGURE 2. Pattern $(\mathbf{3}, \mathbf{2}, \mathbf{4})$ for Lorenz series.

Explanation of the definition on DBSCAN and Wishart:

- DBSCAN: Works under the "density-based" philosophy and was proposed in 1996.
- Wishart: Works under network theory and uses nonparametric estimates of the probability density function of nearest friends.

2. FORMATION OF A SET OF POSSIBLE PREDICTION VALUES

Let us denote a set of motifs for pattern $\alpha = \left(k_1^{(\alpha)}, \ldots, k_{L-1}^{(\alpha)}\right)$ as Ξ_α. For a predictable point in the position $t + h$ for each pattern $\alpha \in \aleph(L, k_{max})$, for each motif $C_\alpha \in \Xi_\alpha, C_\alpha = (\eta_1, \ldots, \eta_L)$ prediction algorithm forms vector $C = (y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_1^{(\alpha)}}, y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_2^{(\alpha)}}, \ldots, y_{t+h-k_{L-1}^{(\alpha)}})$. All the points of vector C are either known or predicted by the algorithm in the previous steps. If the Euclidean distance between vector C and truncated motif $Trunc(C_\alpha) = (\eta_1, \ldots, \eta_{L-1})$ is less than a small number ε, the last value of motif $C_\alpha - \eta_L -$ is added to the set of possible prediction values.

For the explanation of this definition, for the approximation of the Euclidean definition, the confirmation that ϵ is particularly definite, and the description of the method with the definition are given. In other words, the motif is "applied" to the head.

In other words, the motif is “applied” to the series so that its last point coincides with the predictable one. If the motif is sufficiently similar to the observed values, there is a high probability that the last value of the motif will be close to the real value of the predictable point in the series.

3. ALGORITHMS FOR DETERMINING NON-PREDICTABLE POINTS

Algorithms for determining unpredictable points use a set of possible predicted values, S_{t+h} as the input data. Here and below, the formulas are given for the prediction $t + h$, which corresponds to a point that is exactly on the prediction horizon h. For intermediate points, the algorithms for determining unpredictable points and calculating single predicted values are the same.

3.1 Forced Prediction (FP)

The algorithm is simple; all points with a non-empty majority of the predicted results are predictable. This algorithm is mainly used to evaluate the performance of other algorithms in determining unpredictable points as a boundary case. In this algorithm, all the lines that have a high percentage of proposals are available, especially all the lines that have the majority of proposals, so that they will be obtained in the definition of non-proposals in particular.

3.2 Large Spread (LS)

This method indicates that if the difference between the maximum and significant values of the application of possible predicted results is sufficiently large, this point cannot be considered predictable. The algorithm requires the validation part of the time series Y_3: $Y_1 \cap Y_3 = \emptyset$; $Y_2 \cap Y_3, = \emptyset$. For each value of Y_3, a set of possible predicted values S_i was calculated. For each S_i the average difference between the minimum and maximum values of the set (spread) was calculated as follows:

$$\bar{L} = \frac{1}{|Y_3|} \sum_{i=1}^{|Y_3|} \max(S_i) - \min(S_i) \qquad (4)$$

The point is unpredictable if the difference between the maximum and minimum values is greater than spread $\bar{L}$ (multiplied by coefficient ν). This method determines whether the learner has the largest and smallest of the recommended signs so that they do not make the line of determination. Therefore, it is necessary to follow the parameters of the recommendations in this offer.

3.3 Rapid Spread Growth (RG)

The algorithm is based on the assumption that if the spread increases monotonically over three or more points in a row, then the points after the third inclusion are unpredictable. Let S_h be the set of possible forecast values for a point on forecast horizon h; the spread for this point is $L_h = \max(S_h) - min(S_h)$. This point is unpredictable if the condition $L_{h-2} < L_{h-1} < L_h$ is true. Under real interpretations, they have their own definitions of this style, and it is clear that the reduction in general should be much more and better.

3.4 Rapid Growth DBSCAN (RD)

The algorithm is similar to the previous one, but it monitors the growth of the number of clusters obtained by clustering the DBSCAN method with a set of possible forecast values. Let us denote the number of clusters by for a point on forecast horizon h as N_h^D. This point is unpredictable if the condition $N_{h-2}^D < N_{h-1}^D < N_h^D$ is true. The algorithm creates lines of recommendation between related clusters and helps track them with the help of DBSCAN.

3.5 Rapid Growth Wishart (RW)

Similar to the previous algorithm, but as a clustering algorithm, the Wishart algorithm is used. The first algorithm was also used with Wishart's clustering algorithm.

3.6 Limit Cluster Size, (LCS)

This algorithm is used with simple predictive value algorithms to calculate the average over the largest cluster. If the largest cluster contains a certain proportion of points from the entire set of possible predicted values, and the number of clusters does not exceed the established value, the point is predictable.

The minimum percentage of points in the largest cluster γ can vary from 5% to 20-50% (in this case, points are predictable only if most of the points in the set of possible predicted values fall into this largest cluster). The maximum number of clusters N_{max} also varied. In the base case, it can be assumed that points with two or more clusters are not predictable because it cannot be unequivocally said that the largest cluster is the correct one. This algorithm is subject to the selection criteria required for large clusters and is used particularly with construction effort.

4. ALGORITHMS FOR CALCULATING A SINGLE PREDICTION VALUE

The set of possible prediction values S_{t+h}: $y_{t+h} = g_0(S_{t+h})$ is used to calculate single prediction value $\hat{y}_{t+h}$ for a calculation algorithm of a single prediction value g_0. Let us denote the i-th value of set S_{t+h} as $y_{t+h}^{(i)}$.

- Average value of the set of possible prediction values is:

$$\hat{y}_{t+h} = \frac{1}{|\hat{S}_{t+h}|} \sum_{i=1}^{|\hat{S}_{t+h}|} \hat{y}_{t+h}^{(i)} \qquad (5)$$

The first definition of a gift is given in the lines of suggestions, and the second definition is used independently.

- Average value weighted by distance to motif is:

$$\hat{y}_{t+h} = \frac{1}{\sum_{j=1}^{|\hat{S}_{t+h}|} \omega_j} \sum_{i=1}^{|\hat{S}_{t+h}|} \omega_i \hat{y}_{t+h}^{(i)}, \text{where } \omega_j = \| C - Trunc(C_\alpha) \|_2 \qquad (6)$$

We are implemented with specific cluster distances of the proposal.

- Center of the largest cluster DBSCAN (DB)

DBSCAN clustering for a set of possible prediction values returns a number of clusters, including the cluster labeled -1, which contains outliers. We denote the set of points that fall into cluster number j as Q_j. Let $j^{max} = \underset{j}{argmax}(|Q_j|)$ for clusters that are not outliers:

$$\hat{y}_{t+h} = \frac{1}{|Q_{jmax}|} \sum_{\hat{y} \in Q_{jmax}} \hat{y} \qquad (7).$$

This algorithm implicitly contains an algorithm for determining non-predictable points: if DBSCAN clustering does not identify any cluster, all points fall into the outlier cluster; thus, the point is non-predictable. The algorithm also suggests that if the DBSCAN cluster is created, not all the lines that are placed in the outlier cluster may be proposed.

- Center of the largest cluster Wishart (WI)

This algorithm is similar to Algorithm 3; however, the Wishart algorithm is used for clustering.

- Center of the largest cluster OPTICS (OP)

The algorithm is similar to that numbered 3, but the OPTICS algorithm is used for clustering. The basic idea of the OPTICS algorithm is similar to that of DBSCAN; however, it better solves the problem of finding meaningful clusters in data with different densities. It is assumed that this algorithm is more suitable for these possible prediction values because a denser cluster can be considered a more reliable cluster for calculating the center as a single prediction value.

During the computational experiments, the DBSCAN largest cluster center algorithm was mainly used. To design the algorithm, follow these steps:

Algorithm 1: Design Proposed Algorithm

Motif Formation Algorithm (Rectangle)
The general z-vectors form.
Cluster the z-vectors to assign motifs.
Use DBSCAN and Wishart algorithms.
Results: Motifs and cluster centers.
Set Formation for Prediction (Rectangle)
For each template, associate the prediction vectors.
Label the predicted vectors with the observed values.
Compute the Euclidean differences and write non-redundant vectors of truncated patterns.
Results: Group for predicted values.
Determine Non-Predictable Points (rectangles): Apply algorithms to determine whether points are predictable.
Hard Prediction (fp).
Large Difference (ls).
Fast Difference Valley (rg).
DBSCAN Predictive Valley (rd).
Wishart Predictive Valley (rw).
Cluster Size (lcs).
Results: Predicted or non-predicted scores.
Calculate Single Prediction Value (Rectangle)
Insertion: Group for predicted values.
Mean Value, Mean Value with Distance.
Large Cluster Center (DBSCAN, Wishart, OPTICS).
Conclusion: The value of a prediction.

V. SELF-HEALING ALGORITHM

The multistep prediction algorithm proposed in this study (a self-healing algorithm) is an iterative process of filling an increasingly large number of points in the interval over which we want to obtain a prediction (the general pseudocode of the algorithm is presented in Figure 3). It is assumed that in each iteration of the algorithm, we know the values of the series at a certain number of points (observed or predicted), and we want to restore the values at the remaining points.

To obtain a prediction in each iteration, a clustering-based prediction algorithm that relies on the concept of non-predictable points is used, this algorithm was proposed by Gromov and Baranov [1]. The algorithm assumes that many motifs (characteristic sequences observed in a series) are formulated prior to the start of the iterative process. In each iteration, based on known values, using motifs, the algorithm constructs (adds) set of possible prediction values for all points in the interval, both those that already have prediction values and those for which the algorithm has not yet been able to obtain prediction values. Next, the point predictability verification algorithm was applied to each set of possible prediction values (corresponding to each point in the interval for which we wanted to obtain a prediction). This algorithm determines whether it is possible to obtain a single prediction value for a given point (i.e., whether it is predictable) based on its set of possible prediction values. If a point is predictable, the algorithm for calculating a single prediction value from the set of possible prediction values is applied. Each of these algorithms is discussed in more detail in the subsequent sections (see also [1]). Interestingly, in iterations of the self-healing algorithm, two opposite situations are possible: a non-predictable point can become predictable, which is what we are striving for, and vice versa, a predictable point can become unpredictable.

Algorithm 2 Self-healing algorithm

procedure SELF-HEALING($Y_1, h, \hat{Y}_2, \hat{S}$)
  for $i \leftarrow 1, h$ **do**
    for $\alpha \in \beta\Re(L, k_{max})$ **do** ▷ here $\alpha = (k_1^{(\alpha)}, ..., k_{L-1}^{(\alpha)})$
      for $j \leftarrow 1, L$ **do**
        for $l \leftarrow 1, L$ **do**
          $\eta_l^{(\alpha)} \leftarrow y_{t-k_{L-1}^{(\alpha)}-\ldots-k_{L-i+1}^{(\alpha)}}$
        end for
        $C_\alpha \leftarrow (\eta_1^{(\alpha)}, ..., \eta_L^{(\alpha)})$
        $C' \leftarrow (y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_1^{(\alpha)}}, ..., y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_{j-1}^{(\alpha)}},$
          $y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_{j+1}^{(\alpha)}}, ..., y_{t+h})$
        $Trunc'(C_\alpha) \leftarrow (\eta_1, ..., \eta_{j-1}, \eta_{j+1}, ..., \eta_L)$
        if $\rho(C', Trunc'(C_\alpha)) < \varepsilon$ **then** ▷ ρ - Euclidian distance
          add $\eta_L^{(\alpha)}$ to $\hat{S}_{t+i}$
        end if
      end for
    end for
    if point $t+i$ is predictable **then** ▷ by corresponding algorithm
      calculate $\hat{y}_{t+h}$ ▷ by corresponding algorithm
    end if
  end for
end procedure

FIGURE 3. Pseudocode of the self-healing algorithm.

Similar to the basic prediction algorithm, the self-healing algorithm solves the problem of optimizing functions (1) and (2). The algorithm was iterative. In each iteration, the algorithm performs the following functions.

- searching for new possible prediction values for each point up to the prediction horizon,
- updating the status of points from predictable to non-predictable or vice versa,
- calculating new single prediction values for predictable points.

Algorithm stopping criterion:

- in the last three iterations the set of predictable points remains the same;
- The Euclidean distance between the vectors consisting of single predicted values in the last two iterations should be no more than a small number, ε_h.

1. FORMATION OF A SET OF POSSIBLE PREDICTION VALUES

The possible prediction values are calculated in almost the same manner as in the basic algorithm. However, the predictable point can be in any position in the motif with which the segment of the series containing the predicted point is compared. In other words, for each pattern $\alpha = (k_1^{(\alpha)}, \ldots, k_{L-1}^{(\alpha)})$, for each motif $C_\alpha \in \Xi_\alpha, C_\alpha = (\eta_1, \eta_2, \ldots, \eta_L)$, for each position in the motif $j = (1,2,\ldots,L)$ vector C' is compiled: $C' = (y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_1^{(\alpha)}}, \ldots, y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_{j-1}^{(\alpha)}}, y_{t+h-k_{L-1}^{(\alpha)}-\ldots-k_{j+1}^{(\alpha)}}, \ldots, y_{t+h})$. All values of vector C' are either known or calculated in previous iterations or by the main algorithm. If the Euclidean distance between C' and $Trunc'(C_\alpha) = (\eta_1, \ldots, \eta_{j-1}, \eta_{j+1}, \ldots, \eta_L)$ is less than a small number ε, the component of motif η_j is considered a novel possible prediction value for the point at position $t + h - k_{L-1}^{(\alpha)} - \ldots - k_p^{(\alpha)}$. The status of a point from predictable to unpredictable and vice versa is changed in accordance with the algorithms for determining unpredictable points, which are described in the corresponding section of this study. However, in some experiments, it is useful to record the status of points that are predictable as a result of the operation of the basic prediction algorithm.

2. PROPOSE ALGORITHMS FOR CALCULATING A SINGLE PREDICTION VALUE

In addition to the set of possible prediction values, in the self-healing algorithm, one more parameter is used to calculate a single prediction value, *in which iteration the possible prediction value was obtained*.

For example, it can be assumed that one should trust the predictions obtained by the basic algorithm, and as the iteration number increases, the possible prediction values become less trustworthy. In contrast, it should be considered that the predictions obtained by the basic algorithm initially contain a significant error, which can lead to an exponential increase in the error in subsequent positions. That is, the possible prediction values obtained by the self-healing algorithm should be considered more significant and the errors of the basic prediction should be corrected.

This study proposed several algorithms for calculating a single prediction value with regard to the iterations in which possible prediction values were obtained. We introduce the following notation: $\hat{S}_{t+h}^{[i]}$ is the set of possible prediction values obtained in the i-th iteration, $\hat{S}_{t+h}^{[i]}$ is a subset of the set of possible prediction values $\hat{S}_{t+h}$; i_{max} is the number of iterations in which the algorithm operated until the stopping criterion was met, and the zero iteration is the basic prediction algorithm, that is, $\hat{S}_{t+h}^{[0]}$ is the set of possible prediction values obtained by the basic algorithm.

2.1 Double Clustering (DC)

For each iteration single prediction value $\hat{y}_{t+h}^{[i]}$ is calculated individually using only values of set $\hat{S}_{t+h}^{[i]}$ for $0 \le i \le i_{max}$. The obtained values were then clustered using the DBSCAN algorithm with parameters $\varepsilon = 0.1, min\_samples = 2$. In other words, a cluster can contain at least two elements, and the minimum distance for elements within the cluster is much greater than that when clustering many possible prediction values. This makes it possible to obtain at least one cluster with a high probability that is different from the outlier cluster. A single predicted value was calculated as the average of the largest clusters.

The calculation algorithm $\hat{y}_{t+h}^{[i]}$ can be any one, but the algorithm calculating the center of the largest DBSCAN cluster is mainly used; hence, it is called double clustering.

2.2 Weighted Average (WA)

For each iteration single prediction value $\hat{y}_{t+h}^{[i]}$ is calculated individually using only values of set $\hat{S}_{t+h}^{[i]}$ for $0 \le i \le i_{max}$. single prediction value as weighted average: $\hat{y}_{t+h} = \frac{1}{\sum_{i=0}^{i_{max}} \omega_i} \sum_{i=0}^{i_{max}} \omega_i \hat{y}_{t+h}^{[i]}$. The weights are determined using the factor $f > 0$, and the weight of the intermediate prediction $\hat{y}_{t+h}^{[i]}$ is set as $\omega_i = f^i$.

3. The Center of the Cluster, which is the Largest Regarding the Weight of the Points in It

A point in the set of possible prediction values is assigned weight depending on the iteration in which it was obtained; for each point $\hat{y} \in \hat{S}_{t+h}^{[i]}$ weight is equal $\omega(\hat{y})$.

Clustering of the set of possible prediction values $\hat{S}_{t+h}$ is performed. The cluster size was calculated with regard to the weights of the points included in it. Let Q_j be the set of points that fall into cluster number j. The cluster size was equal $K_j = \sum_{\hat{y} \in Q_j} \omega(\hat{y})$. Then $\hat{y}_{t+h} = \frac{1}{|Q_{j^{max}}|} \sum_{\hat{y} \in Q_{j^{max}}} \hat{y}$, where $j^{max} = \underset{j}{argmax}(K_j)$.

Weights can have different structures: A factor $f > 0$ is chosen, and the weight of the point obtained in the j-th iteration is set as $\omega(\hat{y}) = f^j$. If $f < 1$, the values obtained in the early iterations have greater weights than those obtained in the later iterations. If $f > 1$, then we have the opposite situation. This method makes it possible to test hypotheses regarding the significance of minimizing the error of the iteration in which the value was obtained (factor). The weight is calculated as the inverse of the distance to the motif from which the value was obtained, that is, $\omega(\hat{y}) = \frac{1}{d(\hat{y})}$, where $d(\hat{y})$ is the distance to the corresponding motif. Points that were closer to the motif from which they were derived had more weight (dist). The weight was calculated as the inverse of the length of the pattern from which the value was obtained, that is, $\omega(\hat{y}) = \frac{1}{k_1^{(\hat{y})} + .. + k_{L-1}^{(\hat{y})}}$, where $k_1^{(\hat{y})}, \ldots, k_{L-1}^{(\hat{y})}$ are the distances in the pattern. This algorithm can

be used to test the hypothesis that the closer the motif points are to the predicted ones, the greater the trust in the prediction obtained with its help (pattern length, pl).

3. ALGORITHMS FOR DETERMINING NON-PREDICTABLE POINTS

As with the new algorithms for calculating simple predictive values, the new algorithms for identifying unpredictable points use one additional parameter, *in which iteration the possible predictive value was obtained.*

3.1 Big Leap (BL)

This algorithm achieves the result after the completion of the next iteration. The algorithm requires a validation part of the time series $Y_3 : Y_2, Y_1 \cap Y_3 = \emptyset$; $Y_2 \cap Y_3, = \emptyset$. For validation sample Y_3, the minimum and maximum differences of the adjacent values of the time series ($d_{min}, d_{\max}$). is calculated. If the difference from the previous value $d_{t+h-1,t+h}$is less than the minimum or greater than the maximum (with coefficients k_{min} and k_{max}), the point is unpredictable.

3.2 Big Leap Between Iterations (BLBI)

The algorithm assumes that the single predicted values change slightly from one iteration to another. If the single predicted value calculated at the current iteration is significantly different from the previous one, then the point is unpredictable. If the point is unpredictable at the previous iteration, then the algorithm considers no more than three iterations.

3.3 Weird Patterns (WP)

Computational experiments have shown that at intermediate iterations, sequences of observations are possible that definitely cannot be found in the current time series. The algorithm uses motives for certain patterns (e.g., {(1,1,1), (1,2,1)}). If, after the completion of the next iteration, the resulting time series contains generalized z-vectors for which no sufficiently close motive has been found, then all points of the generalized z-vector, except for the first, are unpredictable.

VI. CALCULATION EXPERIMENTS

The used patterns and their corresponding motifs were calculated using the training part of the Y_1 time series, and calculations were made for the algorithm to determine non-predictable points on the validation part of the Y_3 time series, if provided for by the algorithm.

The experiment was performed on the test part of the time series $Y_2 = (y_1^{(2)}, y_2^{(2)} \ldots, y_n^{(2)})$, where n is the size of the test part. For each prediction horizon $1 \le h \le h_{max}$ the average errors and average number of non-predictable points are calculated as follows:

Each point $y_i^{(2)} \in Y_2$ is designated by the algorithm as a point for prediction, the position of which coincides with the current prediction horizon h. In other words, Y_2 is divided into known and predictable parts: $(y_1^{(2)}, \ldots, y_{i-h}^{(2)})$ и $(y_{i-h+1}^{(2)}, \ldots, y_i^{(2)})$, respectively. The prediction algorithm is launched to obtain a prediction for the i-th point of the test part of the time series for the prediction horizon h — $\hat{y}_i^{(2)}(h)$. All predictions corresponding to one prediction horizon are combined into the set $\hat{Y}_2(h) = \{\hat{y}_i^{(2)}(h) | 1 \le i \le n\}$.

1. EXPERIMENT FOR A TIME SERIES WITH THROWN POINTS

To verify whether the self-healing algorithm restores intermediate points and its accuracy, an experiment with thrown points was used. From the test part of the series, the values of which are known, a certain number of points are "thrown": they are declared unpredictable, and they have an empty set of possible prediction values. The remaining points are predictable; for the correct operation of the algorithm, the set of possible prediction values is a cluster of 20 observations, each of which is the true value of a series with random noise $\sigma(\Delta)$, zero mathematical expectation and variance $\Delta > 0$.

One version of the experiment used a fixed status for the points that were not thrown. In this case, the assessment of the algorithm operation quality includes only the restoration of intermediate points rather than the correction of errors of previous iterations.

The limitations of the proposed method include the following. In general, the surface of this highly bare formula in the form of only private connectivity and shows only high connectivity.

This experiment can be conducted to visually evaluate the restoration of intermediate points and calculate the prediction errors in special cases of the number of thrown points, for example, for five thrown points. A more detailed study includes calculating prediction errors (mean square and mean absolute errors), the number of non-predictable points, and the number of iterations required by the self-healing algorithm for the number of thrown points $1 \leq k \leq h$, where h is the prediction horizon being studied.

Figure 4 shows plots of the number of non-predictable points versus the prediction horizon, which were obtained in an experiment assessing the quality of the model for a test sample size of 500 and maximum prediction horizon of $h = 200$.

For the forced prediction (fp) algorithm, unpredictable points are points for which the algorithm for calculating a single prediction value (the center of the largest cluster DBSCAN) cannot be determined. The percentage of non-predictable points did not become 1 as the prediction horizon increased. This implies that the main algorithm can predict some test sample points on a prediction horizon greater than 100. These results are comparable to those in [1], which indicates that the implementation of the main algorithm is correct.

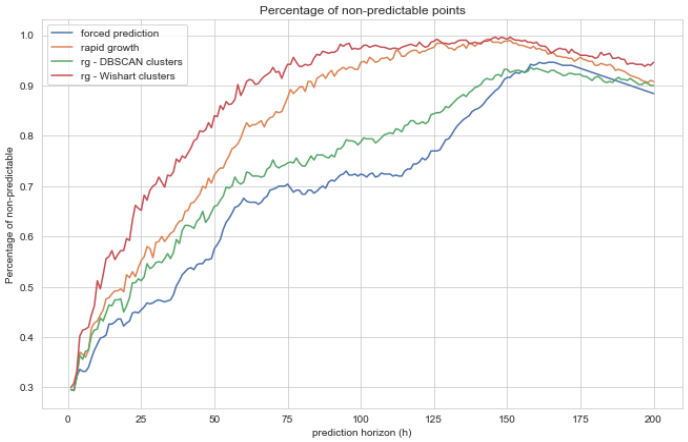


FIGURE 4. The percentage of unpredictable points (prediction horizon, h=200. Test sample is 500. Motif clustering (dB), 20%. ε=0.01. Algorithm for calculating a single prediction value, db. Algorithms for determining unpredictable points: fp (blue), rg (yellow), rd (green), and rw (red).

2. SELECTING THE SELF-HEALING ALGORITHM

The self-healing algorithm uses multiple possible prediction values and single prediction values of the base algorithm as input parameters. Trajectory prediction, compared with point-by-point prediction, has multiple possible prediction values of larger size and spread. This negatively affects the operation of the self-healing algorithm; a situation similar to retraining a machine learning model may arise. A large amount of information about possible prediction values, the error of which is large, leads to poor-quality results of the self-healing algorithm operation.

One limitation of the self-healing algorithm is its sensitivity to the distribution and spread of prediction values. In many ways, the true job of communicating a private communication big high gives the special meaning of a special high.

Algorithms for determining non-predictable points, such as rapid growth of spread (rg) and rapid growth of the number of clusters (rd, rw), were proposed specifically for trajectory prediction and showed good prediction results. Therefore, an algorithm for determining unpredictable points was studied based on the minimum proportion of points in the largest cluster γ and the maximum number of clusters of the set of possible prediction values N_{max} (lcs). Combinations of parameters $\gamma = 0.05,0.1,0.2,0.5,0.7$ and $N_{max} = 1,2,5,10$ examined and the algorithms were compared to the algorithms of forced prediction, rapid growth of spread and rapid growth of the number of clusters, test sample included 250 points. The following observations were made:

- $N_{max} = 5$ and $N_{max} = 10$ do not differ;
- $N_{max} = 2$ and $N_{max} = 5$ almost do not differ;

- $\gamma = 0.05,0.1,0.2$ do not differ with equal N_{max} and almost do not differ from forced prediction, which does not determine non-predictable points. However, the algorithm for calculating a single prediction value db cannot calculate it if the DBSCAN clustering algorithm does not find any cluster other than the cluster outliers.
- If we compare lcs $\gamma = 0.5, N_{max} = 1$ and the algorithm for rapid growth of spread (rg), we obtain approximately the same number of non-predictable points, but the error on average $h = 50$ is less for lcs $\gamma = 0.5, N_{max} = 1$ (Figure 5).
- If we compare lcs $\gamma = 0.5, N_{max} = 5$ and the forced prediction (fp) algorithm, they hardly differ in terms of the number of non-predictable points and average prediction errors (Figure 5).
- lcs $\gamma = 0.7, N_{max} = 1$ gives too many non-predictable points at $h \geq 50$, on average approximately 96-99%.

In the final experiment, to assess the quality of operation, two options were investigated: lcs with parameters $\gamma = 0.5, N_{max} = 1$ and lcs with parameters $\gamma = 0.7, N_{max} = 2$. The second algorithm produces fewer predictable points in the base algorithm than the first, but with a smaller prediction error, thus testing the hypothesis about what is more important: a larger number of predictable points or a smaller prediction error (Figure 5).

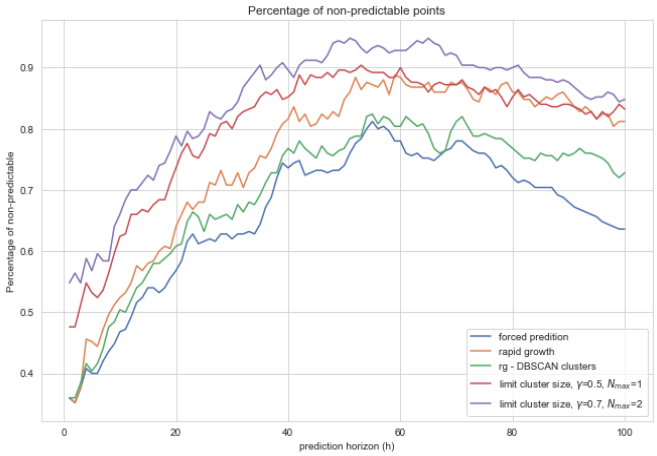
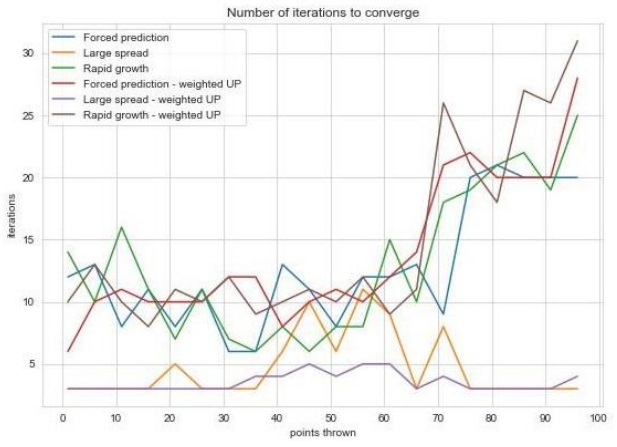
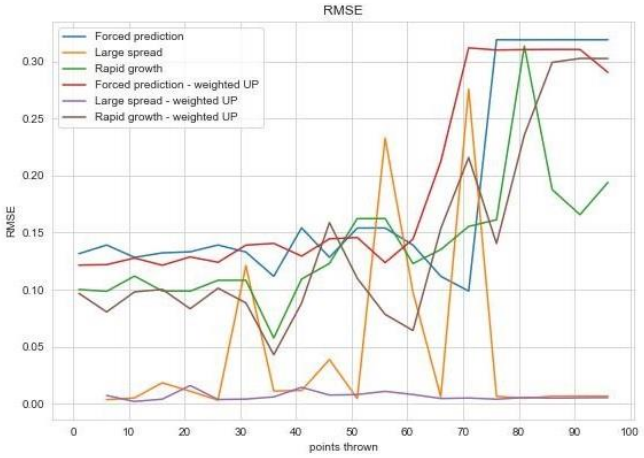


FIGURE 5. Percentage of non-predictable points from the prediction horizon. The prediction horizon was h=100. Test sample 250. Motif clustering (dB), 20%. ε=0.01. The algorithm used to calculate a single prediction value was DB. Algorithms for determining unpredictable points: fp (blue), rg (yellow), rd (green), lcs_0.5_1 (red), and lcs_0.7_2 (purple).

3. STUDY OF THE ALGORITHMS FOR DETERMINING NON-PREDICTABLE POINTS

Figure 6 shows plots of the number of iterations, number of non-predictable points, RMSE, and MAPE versus the algorithm for determining non-predictable points, and the algorithm for calculating a single predicted value. The status of all points can change from predictable to unpredictable, and vice versa. The number of thrown points was taken from 1 to 96 in increments of five.





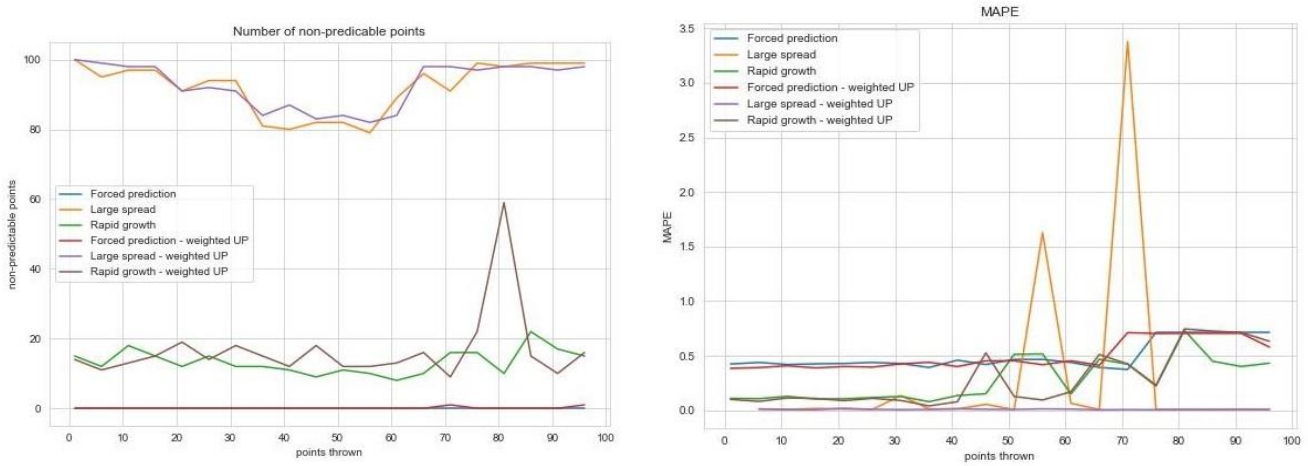


FIGURE 6. Plots of number of iterations, number of non-predictable points, RMSE, and MAPE vs. number of thrown points. The maximum prediction horizon was h=100. Step 5. Motif clustering-cl_db, β=20%. ε=0.01. The following combinations were presented: fp + db (blue), ls + db (yellow), rg + db (green), fp + factor (red), ls + factor (purple), and rg + factor (brown).

The main limitation of the algorithms for determining non-predictable points is their inability to maintain consistent prediction accuracy when the number of thrown points varies significantly. A special feature of this interface is that there are very few special payments that can be made in a special form of high communication.

The self-healing algorithm was expected to restore some of the throw points and slightly change the remaining points. In other words, it is expected that the plot of the number of non-predictable points will increase linearly with the number of thrown points. No algorithm satisfied this hypothesis. Moreover, the large spread algorithm defines 80-100% of the points as non-predictable, regardless of the number of thrown points, from which we can conclude that with these parameters, it is not suitable for the self-healing algorithm. The percentage of unpredictable points for the fast spread growth algorithm remains at approximately 20%, which again does not meet the hypothesis. For points 1, 6, and 11, the number of non-predictable points increases, although the remaining points are the true values of the series. The iteration-weighted average algorithm did not improve the prediction quality on an average.

It can be concluded that these algorithms for determining unpredictable points are not suitable for the self-healing algorithm.

4. STUDY OF NOVEL ALGORITHMS FOR CALCULATING A SINGLE PREDICTION VALUE

The algorithms for calculating a single prediction value, as listed in paragraph 3.2 of Chapter 2, were studied on a time series with five thrown points. This approach was chosen based on the following assumptions. If the self-healing algorithm does not restore intermediate points with good accuracy using real points in the time series, intermediate points will not be restored with good accuracy using predictions obtained by the basic algorithm and will have a non-zero prediction error.

To test the recovery of the throw points, experiments were conducted with a fixed starting point status. The RMSE and MAPE results are given for all predictable points to the prediction horizon $h = 100$.

Table 1. RMSE, MAPE and the number of non-predictable points depending on the algorithm for calculating a single prediction value. Motif clustering — db, 20%. ε=0.01.

Algorithm for determining non-predictable points	Algorithm for calculating a single prediction value	RMSE	MAPE	Number of non-predictable points
rd	db	0.11	0.38	4
rd	wa	0.14	0.18	11
rd	dc	0.13	0.15	9
rd	db, factor = 0.5	0.12	0.41	7
rd	db, factor = 0.6	0.13	0.41	9

rd	db, factor = 0.8	0.13	0.43	10
rd	db, dist	0.16	0.48	3
rd	db, pl	0.14	0.44	2
rd	db	0.12	0.39	0
rd	db, factor = 0.4	0.17	0.18	1
rd	db, factor = 0.5	0.12	0.38	0
rd	db, factor = 0.9	0.12	0.38	0
rd	db, dist	0.12	0.39	2
rd	db, pl	0.12	0.39	1

As the results in Table 1 show, the novel algorithms for calculating a single prediction value do not improve the quality of prediction. In some cases, the MAPE is reduced, but the RMSE is not improved compared with the algorithm without considering iteration when computing a single prediction value. This is associated with the incorrect adjustment of real points in section $h = [15,30]$. It was also noted that there were motifs that were not found in the present time series (Figure 7). None of the algorithms studied for calculating a single prediction value were able to correctly predict this section of the time series.

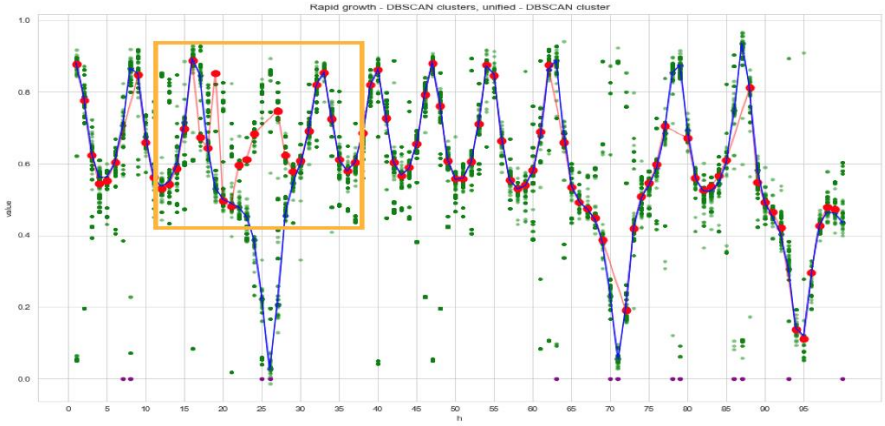


FIGURE 7. Last iteration of self-healing algorithm for experiment with five thrown points. Motif clustering (dB), 20%. ε=0.01. Algorithm for calculating a single prediction value, db. Algorithms for determining non-predictable points — rd. Blue indicates the real-time series, green points are possible prediction values, red points and lines are single prediction values, and purple points on the straight-line y=0 are non-predictable points. The section that was not similar to the motifs found in the real-time series is highlighted in orange.

In the experiment described in paragraph 3, an algorithm factor with a parameter of 0.7. is used, despite the fact that it does not significantly improve the quality of prediction in a series with five thrown points. This is justified by the intention to test the hypothesis that values in earlier iterations should be considered with greater weight for a larger sample size.

5. STUDY OF NOVEL ALGORITHMS FOR DETERMINING NON-PREDICTABLE POINTS

Algorithms for determining non-predictable points were also first studied on a time series with five thrown points. The main problem that these algorithms must solve is identifying points in the section $h = [15,30]$ as unpredictable. The big-leap (*bl*) algorithm examines the first differences of a time series; if the first difference is greater than the maximum in the current time series, the second difference point is not predictable. The large leap between iterations (*blbi*) does not allow the algorithm to dramatically adjust the predictions obtained in previous iterations. The (*wp*) algorithm examines the series obtained after the next iteration for the presence of motifs not found in the real series.

FIGURE 8. Plots of the number of iterations, number of non-predictable points, RMSE, and MAPE vs. the number of thrown points. The maximum prediction horizon was h=100. Increment – 5. Motif clustering-cl_db, β=20%. Algorithm for calculating a single prediction value: factor = 0.7. Algorithms for determining unpredictable points: fp (blue), ls (orange), rg (green), blbi (red), and wp (purple).

Table 2. RMSE, MAPE and the number of non-predictable points depending on the algorithm for calculating a single prediction value. Motif clustering — db, 20%. ε=0.01.

Algorithm for determining non-predictable points	Algorithm for calculating a single prediction value	RMSE	MAPE	Number of non-predictable points
rd	db	0.11	0.38	4
rd, bl	db	0.13	0.42	2
rd, blbi	db	0.04	0.04	7
rd, wp	db	0.06	0.04	16

Table 2 presents the results of the study of the novel algorithms for determining non-predictable points in a time series with five thrown points. The large leap between the iteration (*blbi*) algorithm and the weird pattern algorithm (*wp*) significantly improved the quality of prediction in terms of average prediction errors.

Figure 8 shows a comparison of algorithms for determining non-predictable points in the thrown-points experiment from 1 to 96 in increments of 5. On average, with a close percentage of predictable points, the error of the novel algorithms (*blbi* and *wp*) is less than that of the forced prediction algorithms (*fp*) and the rapid growth in spread (*rg*).

We can conclude that novel algorithms for determining non-predictable points improve the quality of prediction; therefore, the big-leap algorithm between iterations and the weird patterns algorithm is used in the next experiment.

6. STUDY OF THE SELF-HEALING ALGORITHM FOR CHAOTIC TIME SERIES PREDICTION

A quality assurance experiment was conducted to assess the quality of the self-healing algorithm, and a test sample included 20 observations. Tables 3 and 4 list the experimental results.

Two basic prediction algorithms with algorithms for determining non-predictable points lcs $\gamma = 0.5, N_{max} = 1$ and lcs $\gamma = 0.7, N_{max} = 2$ are compared with four prediction algorithms using self-healing: with the same algorithms for determining non-predictable points; however, the self-healing algorithm uses algorithms of big leap between iterations and weird patterns.

Table 3. RMSE and MAPE results averaged over the test sample, percentage of non-predictable points (n-p, %) for two basic algorithms and four self-healing algorithms. Motif clusterization — db, 20%. ε=0.01. Prediction horizon $h = 1,5,20$.

Algorithm type	DNPP algorithm	CSPV algorithm	h = 1			h = 5			h = 20		
			RMSE	MAPE	n-p, %	RMSE	MAPE	n-p, %	RMSE	MAPE	n-p, %
s	lcs_0.5_1	db	0.01	0.01	25	0.02	0.17	30	0.13	0.05	85
s, sh	lcs_0.5_1, wp	db, factor = 0.7	0.01	0.01	5	0.02	0.02	5	0.06	0.05	10
s, sh	lcs_0.5_1, blbi	db, factor = 0.7	0.01	0.01	5	0.02	0.02	5	0.06	0.05	10
s	lcs_0.7_2	db	0.01	0.01	35	0.02	0.02	30	0.13	0.17	85
s, sh	lcs_0.7_2, wp	db, factor = 0.7	0.01	0.01	5	0.02	0.02	5	0.06	0.14	15
s, sh	lcs_0.7_2, blbi	db, factor = 0.7	0.01	0.01	5	0.02	0.02	5	0.06	0.05	15

Note: DNPP algorithm – the algorithm for determining non-predictable points

CSPV algorithm – the algorithm for calculating a single prediction value

Table 4. RMSE and MAPE results averaged over the test sample, percentage of non-predictable points (n-p, %) for two basic algorithms and four self-healing algorithms. Motif clusterization — db, 20%. ε=0.01. Prediction horizon $h = 50,70,100$.

Algorithm type	DNPP algorithm	CSPV algorithm	h = 50			h = 70			h = 100		
			RMSE	MAPE	n-p, %	RMSE	MAPE	n-p, %	RMSE	MAPE	n-p, %
s	lcs_0.5_1	db	0.02	0.05	90	0.2	0.14	30	0.24	0.17	50
s, sh	lcs_0.5_1, wp	db, factor = 0.7	0.07	0.07	30	0.26	0.2	0	0.26	0.21	0
s, sh	lcs_0.5_1, blbi	db, factor = 0.7	0.25	0.18	5	0.26	0.2	0	0.26	0.21	0
s	lcs_0.7_2	db	—	—	100	0.22	0.17	25	0.08	0.1	45
s, sh	lcs_0.7_2, wp	db, factor = 0.7	0.2	0.15	20	0.26	0.2	0	**0.26**	**0.2**	**0**
s, sh	lcs_0.7_2, blbi	db, factor = 0.7	0.27	0.2	5	0.26	0.2	0	**0.26**	**0.2**	**0**

Note: DNPP algorithm – the algorithm for determining non-predictable points

CSPV algorithm – the algorithm for calculating a single prediction value

At prediction horizons $h = [1,5]$, the average errors RMSE and MAPE for prediction by the basic and self-healing algorithms did not differ; however, the number of unpredictable points was significantly lower. At the prediction horizon $h = 20$, the average prediction errors were twice as low, whereas the number of non-predictable points decreased significantly. At the prediction horizon $h = 50$, the average error was significantly larger, but the number of unpredictable points decreased. At prediction horizons $h = [70,100]$, the average error was larger, but the number of unpredictable points for the self-healing algorithm was zero.

Additionally, nonparametric statistical tests were conducted to assess the difference between the prediction errors, RMSE, MAPE, and the number of non-predictable points. The Mann-Whitney U test was used because it is suitable for small samples.

The null hypothesis of the Mann-Whitney U test is that the groups are drawn from the same population, and the alternative hypothesis is that there is a significant difference between the level of the trait in the samples under consideration.

Table 5. p-values for the Mann-Whitney U test, comparison of two basic prediction algorithms with four self-healing algorithms.

RMSE	lcs_0.5_1, wp	lcs_0.5_1, blbi	lcs_0.7_2, wp	lcs_0.7_2, blbi
lcs_0.5_1	0.94	0.59	0.82	0.59
lcs_0.7_2	1.0	1.0	1.0	1.0
MAPE	lcs_0.5_1, wp	lcs_0.5_1, blbi	lcs_0.7_2, wp	lcs_0.7_2, blbi
lcs_0.5_1	0.94	0.59	0.94	0.59
lcs_0.7_2	1.0	1.0	1.0	1.0
Percentage of non-predictable points	lcs_0.5_1, wp	lcs_0.5_1, blbi	lcs_0.7_2, wp	lcs_0.7_2, blbi
lcs_0.5_1	0.009	0.002	0.002	0.002
lcs_0.7_2	0.008	0.008	0.008	0.008

Table 5 shows the p-values of the statistics obtained by comparing the RMSE and MAPE. At the significance level α=0.05, the null hypothesis was accepted for the RMSE and MAPE; that is, the RMSE and MAPE were not statistically different when comparing the basic and self-healing algorithms. However, for the percentage of non-predictable points, an alternative hypothesis is accepted, which implies a statistically significant difference in the percentage of non-predictable points depending on the type of algorithm.

VII. CONCLUSION

In this study, a self-healing algorithm for clustering prediction was developed and analyzed. During this research, the subject area was studied, a basic clustering prediction algorithm was implemented, a self-healing algorithm was developed, and several new algorithms for determining non-predictable points and algorithms for calculating a single prediction value were proposed. The self-healing algorithm was used to predict the chaotic Lorentz time series, and based on the results of computational experiments, conclusions were drawn regarding the effectiveness of the self-healing algorithm and new algorithms for determining non-predictable points and algorithms for calculating a single predicted value. For the basic algorithm, an algorithm for limiting the minimum size of the largest cluster and the maximum number of clusters with parameters γ = 0.5, $N_{max} = 1$ and γ = 0.7, $N_{max} = 2$ turned out to be the most suitable for determining non-predictable points, considering the further use of the self-healing algorithm. The first algorithm, on average, determines more predictable points than the second; however, it has a larger average prediction error, which does not significantly affect the final quality of prediction. This study proposed several novel algorithms for calculating a single prediction value. They did not significantly affect the prediction quality in the experiment with five dropped points. Their research helped identify the problem of incorrect adjustment of the predictions obtained in previous iterations. To solve this problem, several new algorithms for determining non-predictable points were proposed, and good results were shown by the algorithm of big leap between iterations and the weird patterns algorithm. During the experiment evaluating the quality of the self-healing algorithm and comparing it with the basic prediction algorithm, the following results were obtained. The average prediction errors RMSE and MAPE of the basic algorithm and self-healing algorithm are not statistically different for the corresponding prediction horizons according to the Mann-Whitney U test. However, this criterion showed a statistically significant difference in the percentage of non-predictable points, depending on the type of algorithm. In other words, on average, more points were predictable, whereas the prediction errors remained almost at the same level. In future research, it will be possible to conduct experiments on larger samples and study the hyperparameters of the self-healing algorithm, algorithms for determining unpredictable points, and algorithms for calculating a single prediction value, which have already shown good prediction quality. It is also possible to study the prediction quality of the self-healing algorithm on other chaotic time series, such as financial series. One limitation of self-healing algorithms is that they may have difficulty adapting to sudden changes in the dataset or in the prediction environment. Typically, this formula is only high, and one of the shapes is specific to the high-high shape. The proposed self-healing algorithms were evaluated based on their performance with various types of prediction scenarios, including the number of thrown points and prediction horizon. A special

shape of this formula should be high in special high shapes that show particularly high values. Future research should focus on optimizing self-healing algorithms and evaluating their performances in more complex and dynamic scenarios.

Funding statement

The authors wish to acknowledge that no specific funding or support was provided for this study.

Author contribution

All authors made an equal contribution to the development and planning of the study.

Conflict of Interest

The authors declare no conflicts of interest.

Data Availability Statement

Data are available from the authors upon request.

Acknowledgements

This work/article is an output of a research project implemented as part of the Basic Research Program at the HSE University and Strategic Project "'Human Brain Resilience: Neurocognitive Technologies for Adaptation, Learning, Development and Rehabilitation in a Changing Environment'", which is part of Higher School of Economics' development program under the "Priority 2030" academic leadership initiative. The "Priority 2030" initiative was run by Russia's Ministry of Science and Higher Education as part of the National Project "Science and Universities."

REFERENCES

1. Gromov, V. A., & Baranov, P. S. (**2023**). Prediction after a horizon of predictability: Nonpredictable points and partial multistep prediction for chaotic time series. *Complex*, 1–21.
2. Malinetsky, G. G., & Potapov, A. B. (**2000**). *Modern problems of nonlinear dynamics*. Editorial URSS: Moscow, Russia.
3. Meng, X., & Yang, T. (**2021**). Entanglement-structured LSTM boosts chaotic time series forecasting. *Entropy*, 23, 1491.
4. Bao, Y., Xiong, T., & Hu, Z. (**2014**). Multi-step-ahead time series prediction using multiple-output support vector regression. *Neurocomputing*, 129, 482–493.
5. Assaad, M., Boné, R., & Cardot, H. (**2006**). Predicting chaotic time series by boosted recurrent neural networks. In *Lecture Notes in Computer Science* (pp. 831–840). Springer: Berlin, Germany.
6. Taieb, S. B., Bontempi, G., Atiya, A. F., & Sorjamaa, A. (**2012**). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Systems with Applications*, 39, 7067–7083.
7. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (**2007**). Methodology for long-term prediction of time series. *Neurocomputing*, 70, 2861–2869.
8. Bontempi, G. (**2008**). Long term time series prediction with multi-input multi-output local learning. *Proceedings of the 2nd European Symposium on Time Series Prediction (TSP)*, 145–154.
9. Taieb, S. B., Bontempi, G., Sorjamaa, A., & Lendasse, A. (**2009**). Long-term prediction of time series by combining direct and MIMO strategies. *Proceedings of the 2009 International Joint Conference on Neural Networks*, 3054-3061.
10. Chandra, R., Goyal, S., & Gupta, R. (**2021**). Evaluation of deep learning models for multi-step ahead time series prediction. *IEEE Access*, 9, 83105–83123.
11. Gromov, V. A., & Dang, Q. N. (**2023**). Semantic and sentiment trajectories of literary masterpieces.
12. Gromov, V. A., & Konev, A. S. (**2016**). Precocious identification of popular topics on Twitter with the employment of predictive clustering. *Neural Computing and Applications*, 28, 3317–3322.
13. Gromov, V. (**2021**). Chaotic time series prediction: Run for the horizon. In *Tools and methods of program analysis* (pp. 29-43). Springer: Cham, Switzerland.
14. Gromov, V. A., & Shulga, A. N. (**2012**). Chaotic time series prediction with employment of ant colony optimization. *Expert Systems with Applications*, 39, 8474–8478.
15. Takens, F. (**1981**). Detecting strange attractors in turbulence. In *Lecture Notes in Mathematics* (pp. 366–381). Springer: Berlin, Germany.
16. Gromov, V. A., & Borisenko, E. A. (**2015**). Chaotic time series prediction and clustering methods. *Neural Computing and Applications*, 2, 307–315.
17. Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (**1996**). A density-based algorithm for discovering clusters in large spatial databases with noise. *Knowledge Discovery and Data Mining*, 226–231.
18. Wishart, D. (**1969**). Numerical classification method for deriving natural classes. *Nature*, 221, 97–98.
19. Lapko, A. V., & Chentsov, S. V. (**2000**). *Nonparametric information processing systems*. Nauka: Novosibirsk, Russia.