Pattern Recognition Letters 128 (2019) 278–282

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Evolvable fuzzy systems from data streams with missing values: With application to temporal pattern recognition and cryptocurrency prediction

Cristiano Garcia[a], Ahmed Esmin[b], Daniel Leite[c,*], Igor Škrjanc[d]

[a] Department of IT Management, Federal University of Lavras, Lavras, Brazil
[b] Department of Computer Science, Federal University of Lavras, Lavras, Brazil
[c] Department of Engineering, Federal University of Lavras, Lavras, Brazil
[d] Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 26 August 2018
Revised 7 September 2019
Accepted 17 September 2019
Available online 18 September 2019

Keywords:
Temporal pattern recognition
On-line algorithm
Real-time system
Fuzzy system
Machine learning

ABSTRACT

Data streams with missing values are common in real-world applications. This paper presents an evolving granular fuzzy-rule-based model for temporal pattern recognition and time series prediction in online nonstationary context, where values may be missing. The model has a modified rule structure that includes reduced-term consequent polynomials, and is supplied by an incremental learning algorithm that simultaneously impute missing data and update model parameters and structure. The evolving Fuzzy Granular Predictor (eFGP) handles single and multiple Missing At Random (MAR) and Missing Completely At Random (MCAR) values in nonstationary data streams. Experiments on cryptocurrency prediction show the usefulness, accuracy, processing speed, and eFGP robustness to missing values. Results were compared to those provided by fuzzy and neuro-fuzzy evolving modeling methods.

1. Introduction

Detecting patterns, trends, seasonalities, nonstationarities in temporal data may help human decision-making in a variety of situations and endeavors. Pattern recognition, machine learning and computational intelligence methods have been considered to discover useful information in datasets. In particular, models to deal with time-varying data streams must take into account that: (i) samples cannot be permanently stored; (ii) data sets are potentially unbounded; (iii) processing time should not scale exponentially with the number of samples, attributes and parameters; and (iv) the data distribution may change and new patterns may emerge [1,5,9,10].

The vast majority of stream-oriented learning methods require the values of all attributes to be available to work properly [4,15]. However, missing values are common in real-world applications. Missing data arise due to incomplete observations, data transfer problems, and malfunction of sensors [3]. Four ways for handling missing data can be mentioned: (i) discard incomplete samples, or attributes with more than a predefined percentage of their values missing; (ii) impute zeros, mean, or median for an attribute; (iii) impute values by maximum likelihood and parameter estimation procedures; and (iv) identify relationships among attributes and previous values to estimate the missing values.

Statistical and intelligent methods have been proposed to deal with missing data in offline settings [13]. These methods rely on techniques such as Fuzzy C-Means Clustering [2], Support Vector Regression [2,16], Weighted K-Nearest-Neighbor and Auto-Encoder [3], Regularized Expectation-Maximization and Bayesian Principal Component Analysis [13], Time Warping [14], and Neural Networks [6]. These techniques may be combined with machine-learning procedures. Often, meta-heuristics, e.g. multi-objective Genetic Algorithm [2,12] or Bio-Inspired [6] techniques are used for parameter adaptation. Intelligent models provide a nonlinear way to impute missing data based on information uncovered from the data.

Limitations of the aforementioned methods include: (i) prior availability of a closed dataset is generally necessary; (ii) in spite of the use of nonlinear models, offline learning methods require multiple passes over the data to determine the parameters of the imputation model; and (iii) the resulting predictor does not deal with typical changes of online time-varying environment. A survey on evolving fuzzy and neuro-fuzzy systems that deal with these issues and additional challenges of online modeling and learning was published in [17].

An evolving, fuzzy-rule-based model, modified to include a recurrent learning algorithm for missing-data imputation and

* Corresponding author.
E-mail addresses: cristiano.garcia@dgti.ufla.br (C. Garcia), ahmed@dcc.ufla.br (A. Esmin), daniel.leite@deg.ufla.br (D. Leite), igor.skrjanc@fe.uni-lj.si (I. Škrjanc).

https://doi.org/10.1016/j.patrec.2019.09.012
0167-8655/

model adaptation, which we call Evolving Fuzzy Granular Predictor, is presented in this paper. eFGP is useful for prediction in time-varying environment. Its rule base is built from scratch based on a data stream that may contain MAR and MCAR values. A single missing value on a data sample is handled using reduced-term consequent polynomials, whereas multiple missing values are dealt with using the essence of time-varying fuzzy granules evolved in the data space. Therefore, eFGP is different from any other evolving intelligent approach because it addresses single and multiple MAR and MCAR values in time-varying online environment.

2. Missing data

A datum is missing if no value is available for the underlying attribute of a sample. *Case deletion* and *imputation* transform an incomplete dataset into a fully-populated rectangular format. However, the first option is useful only if the number of missing values is small. Depending on the nature of the missing data, they are classified as MCAR, when the probability of any value being missing is equal to the probability of any other value being missing; and MAR, when the propensity of the values of a specific attribute to be missing is higher than those of other attributes, e.g., a sensor is not working properly, or one of the questions in a survey is harder to answer than the others. Slightly different definitions and interpretations may be found across research communities.

Model-based imputation is grounded on statistical and machine-learning fundamentals. While linear and stationary interpolation for imputation may make the completed dataset biased depending on the level of nonlinearity and nonstationarity involved, nonlinear model-based imputation replaces the missing value with the best estimate based on previously sampled data [11]. Data streams impose further challenges to nonlinear modeling such as reasonably limited time and memory.

3. Evolving fuzzy granular predictor

A rule-based model may have its parameters and structure updated when the data stream changes. Handling multiple MAR and MCAR values incrementally for the purpose of predicting future values of a data stream is a concern of eFGP. eFGP provides pointwise and granular prediction of nonstationary functions, and linguistic description of the behavior of a system by means of IF-THEN rules. The granular prediction encloses output data, and may help decision making and improve model acceptability. The enclosure may be interpreted as optimistic and pessimistic estimates in an application, which can be more important than the numerical estimates since, with pointwise values, we have no idea about the error or uncertainty involved in the estimation. Online learning of eFGP models can start from scratch. Rules are developed as new information is uncovered from the data.

Let $(\mathbf{x}, y)^{[h]}$, $h = 1, \ldots,$ be the hth observation of a data stream. $\mathbf{x} \in \Re^n$ is an input multi-dimensional vector, and $y \in \Re$ is the actual output. The actual output $y^{[h]}$ will be known after the input $\mathbf{x}^{[h]}$ arrives and a prediction $\hat{y}^{[h]}$ is given. An attribute x_j of $\mathbf{x} = (x_1, \ldots, x_n)$ is a real value. The same holds for y. The pair $(\mathbf{x}, y)$ is a point in the product space $X \times Y$. Let $\gamma^i \in X \times Y$, $i = 1, \ldots, c$, be the current set of eFGP granules built on the basis of $(\mathbf{x}, y)$. Rules R^i governing granules γ^i are given as

$$R^i: \text{IF } (x_1 \text{ is } A_1^i) \text{ AND } \ldots \text{ AND } (x_n \text{ is } A_n^i)$$
$$\text{THEN } \underbrace{(y \text{ is } B^i)}_{\text{Linguistic}} \text{ AND} \Big(\underbrace{\hat{y} = p^i(x_1, \ldots, x_n)}_{\text{Functional}}$$
$$\text{OR } \underbrace{\hat{y} = q_\theta^i(x_1, \ldots, x_{\theta-1}, x_{\theta+1}, \ldots, x_n),\ \theta = 1, \ldots, n}_{\text{Functional with reduced argument-list } (x_\theta \text{ omitted})} \Big)$$

where x_θ is a missing value; $A_j^i = (l_j^i, \lambda_j^i, \Lambda_j^i, L_j^i)$ and $B^i = (u^i, v^i, \Upsilon^i, U^i)$ are trapezoidal membership functions related to the ith rule; and p^i and q_k^i are affine functions. The core of a membership function, say A_j^i, is the region $[\lambda_j^i, \Lambda_j^i]$ of the universe of x_j characterized by elements with full membership in the set A_j^i. The boundary parameters of a membership function form its support. The support of A_j^i is the region $[l_j^i, L_j^i]$ of the universe x_j characterized by elements with nonzero membership in the set A_j^i. Notice that q_k^i has one term less than p^i and that a disjunction operator (OR) relates the terms. The set of rules R^i, $i = 1, \ldots, c$, is a fuzzy granular description of a system. Initially, $c = 0$, i.e., no prior knowledge is assumed. A rule provides a granular (by means of active output fuzzy sets B^i) and a pointwise (by means of p^i or q_θ^i) prediction. The functional consequent is given by either p^i, in case $\mathbf{x}^{[h]}$ is complete, or q_θ^i, in case x_θ is missing. The linguistic consequent offers prediction bounds and interpretability, since trapezoids B^i can be joined to linguistic values.

Affine functions are given as

$$p^i(\mathbf{x}) = \alpha_0^i + \sum_{j=1}^{n} \alpha_j^i x_j \quad \text{and} \quad q_\theta^i(\mathbf{x}) = \beta_{0\theta}^i + \sum_{j=1, j\neq\theta}^{n} \beta_{j\theta}^i x_j, \tag{1}$$

$\theta = 1, \ldots, n$. Consequent functions, p^i and q_θ^i $\forall\theta$, are updated using the Recursive Least Squares algorithm [7]. As trapezoids A_j^i may overlap, eFGP pointwise prediction is found as the weighted mean value,

$$\hat{y} = \frac{\sum_{i=1}^{c} \Psi_{\text{com}}^i \; p^i(x_1, \ldots, x_n)}{\sum_{i=1}^{c} \Psi_{\text{com}}^i}, \tag{2}$$

for a complete $\mathbf{x}$, or

$$\hat{y} = \frac{\sum_{i=1}^{c} \Psi_{\text{inc}}^i \; q_\theta^i(x_1, \ldots, x_{\theta-1}, x_{\theta+1}, \ldots, x_n)}{\sum_{i=1}^{c} \Psi_{\text{inc}}^i}, \tag{3}$$

if x_θ is missing; where $\Psi_{\text{com}}^i = T(A_1^i(x_1), \ldots, A_n^i(x_n))$ or $\Psi_{\text{inc}}^i = T(A_1^i(x_1), \ldots, A_{\theta-1}^i(x_{\theta-1}), A_{\theta+1}^i(x_{\theta+1}), \ldots, A_n^i(x_n))$ is the activation degree of R^i for $\mathbf{x}$ complete or not; T is any triangular norm, e.g. the minimum operator.

Granular prediction is given by the convex hull of sets B^{i^*}, where i^* are indices of active granules for $\mathbf{x}^{[h]}$. The convex hull of a set of trapezoids, say $B^1, \ldots, B^c$, is

$$\text{ch}(B^1, \ldots, B^c) = (T(u^1, \ldots, u^c), T(v^1, \ldots, v^c), S(\Upsilon^1, \ldots, \Upsilon^c), S(U^1, \ldots, U^c)), \tag{4}$$

where S is a triangular conorm, e.g. the maximum operator.

Suppose a sample $\mathbf{x}^{[h]}$ has multiple missing values, say x_{θ_1} and x_{θ_2}. A straightforward, but not practical, approach to deal with multiple missing data is to consider additional consequent functions with fewer terms. However, the number of parameters to be updated would scale exponentially with the number of attributes. An effective approach for multiple missing values consists in imputing the midpoint of membership functions related to the most active rule for the missing values. In this case, the activation level of the rule R^i, $i = 1, \ldots, c$, is calculated as

$$\Psi_{\text{inc}}^i = T(A_1^i(x_1), \ldots, A_{\theta_1-1}^i(x_{\theta_1-1}), A_{\theta_1+1}^i(x_{\theta_1+1}), \ldots, A_{\theta_2-1}^i(x_{\theta_2-1}), A_{\theta_2+1}^i(x_{\theta_2+1}), \ldots, A_n^i(x_n)). \tag{5}$$

If R^i is the most active rule for $\mathbf{x}^{[h]}$ according to Ψ_{inc}^i, then the midpoint of its membership functions related to the missing values are used for imputation. The midpoint is the mean value of the core parameters, that is,

$$x_{\theta_1}^{[h]} = \frac{(\lambda_{\theta_1}^i + \Lambda_{\theta_1}^i)}{2} \quad \text{and} \quad x_{\theta_2}^{[h]} = \frac{(\lambda_{\theta_2}^i + \Lambda_{\theta_2}^i)}{2}. \tag{6}$$

The imputed sample is used by the fuzzy model to provide numerical and granular predictions at the time step h. The multiple-imputation procedure extends straightforwardly to larger amounts of missing data per sample.

Let ρ_j and σ be the maximum length a granule can assume along the jth input dimension and output dimension. Parameter ρ_j delimits the maximum expansion region of trapezoidal membership functions around their midpoints, namely, A_j^i must not assume values lower than $\text{mp}(A_j^i) - \rho_j/2$ nor values greater than $\text{mp}(A_j^i) + \rho_j/2$ at any time step. Different values of ρ_j produce different models of the same data stream in different granularities. $\rho_j\,\forall j$ and σ assume a single value in [0, 1]. If they are equal to 0, granules are not expanded. Learning creates a new rule for each sample, which causes overfitting. If they are equal to 1, a single granule covers all the data. Evolution is attained by choosing intermediate values. The higher the value of ρ_j, the more compact tends to be the structure of the eFGP model.

A granule γ^{c+1} is created by adding a rule R^{c+1} to the set of rules $R = \{R^1, \ldots, R^c\}$ whenever either an input vector, $\mathbf{x}^{[h]}$, contains at least one element, $x_j^{[h]},\ j = 1, \ldots, n$, that is not in the expansion region of $A_j^i,\ i = 1, \ldots, c$, or $y^{[h]}$ is not in the expansion region of $B^i,\ i = 1, \ldots, c$. Formally, $x_j^{[h]}$ must belong to $[\text{mp}(A_j^i) - \rho/2,\ \text{mp}(A_j^i) + \rho/2],\ j = 1, \ldots, n$, to be considered by the ith granule. Additionally, $y^{[h]}$ must belong to $[\text{mp}(B^i) - \sigma/2, \text{mp}(B^i) + \sigma/2]$. The new γ^{c+1} has trapezoidal memberships functions A_j^{c+1} and B^{c+1}, in which $l_j^{c+1} = \lambda_j^{c+1} = \Lambda_j^{c+1} = L_j^{c+1} = x_j^{[h]},\ \forall j$, and $u^{c+1} = \upsilon^{c+1} = \Upsilon^{c+1} = U^{c+1} = y^{[h]}$. Therefore, the new granule, γ^{c+1}, is initially a point in the data space, the point $(\mathbf{x}, y)$. Initially, the coefficients of p^{c+1} are $\alpha_j^{c+1} = 0, j \neq 0$, and $\alpha_0^{c+1} = y^{[h]}$; the coefficients of $q_\theta^{c+1},\ \forall\theta$, are set as $\beta_{j\theta}^{c+1} = 0,\ j \neq 0$, and $\beta_{0\theta}^{c+1} = y^{[h]}$.

Adapting R^i consists in expanding or contracting the support and the core of A_j^i and B^i, and the coefficients of p^i and q_θ^i to fit new data. If a data sample, $(\mathbf{x}, y)^{[h]}$, belongs to the expansion region of a granule, γ^i, then its membership functions are enlarged to cover the sample. If the sample is within γ^i, parameters can be changed in the sense of contracting or expanding the core of its membership functions. The following situations may happen according to the position of a sample in relation to a granule:

- If $x_j^{[h]} \in [\text{mp}(A_j^i) - \frac{\rho_j}{2}, l_j^i]$, then $l_j^i(\text{new}) = x_j^{[h]}$ (support expansion)
- If $x_j^{[h]} \in [l_j^i, \lambda_j^i]$, then $\lambda_j^i(\text{new}) = x_j^{[h]}$ (core expansion)
- If $x_j^{[h]} \in [\lambda_j^i, \text{mp}(A_j^i)]$, then $\lambda_j^i(\text{new}) = x_j^{[h]}$ (core contraction)
- If $x_j^{[h]} \in [\text{mp}(A_j^i), \Lambda_j^i]$, then $\Lambda_j^i(\text{new}) = x_j^{[h]}$ (core contraction)
- If $x_j^{[h]} \in [\Lambda_j^i, L_j^i]$, then $\Lambda_j^i(\text{new}) = x_j^{[h]}$ (core expansion)
- If $x_j^{[h]} \in [L_j^i, \text{mp}(A_j^i) + \frac{\rho_j}{2}]$, then $L_j^i(\text{new}) = x_j^{[h]}$ (support expansion)

When operating on core parameters, λ_j^i and Λ_j^i, midpoints of γ^i are updated. As a consequence, support contraction may be necessary, thus

- If $\text{mp}(A_j^i) - \frac{\rho_j}{2} > l_j^i$, then $l_j^i(\text{new}) = \text{mp}(A_j^i) - \frac{\rho_j}{2}$
- If $\text{mp}(A_j^i) + \frac{\rho_j}{2} < L_j^i$, then $L_j^i(\text{new}) = \text{mp}(A_j^i) + \frac{\rho_j}{2}$

Adaptation of B^i uses data $y^{[h]}$ and the same relations above. Only the most active granule, γ^i, is chosen to be adapted for a sample $(\mathbf{x}, y)^{[h]}$.

All consequent functions, p^i and $q_\theta^i\ \forall\theta$, are updated using the Recursive Least Squares algorithm [9] in case R^i is the most active rule for a complete $\mathbf{x}^{[h]}$. However, for a complete $\mathbf{x}^{[h]}$, coefficients $\beta_{j\theta}^i,\ j = 0, 1, \ldots, n;\ j \neq \theta$, are computed ignoring the attribute $x_\theta^{[h]},\ \theta = 1, \ldots, n$. For an incomplete $\mathbf{x}^{[h]}$, with a unique missing element, $x_\theta^{[h]}$, only the coefficients of q_θ^i are updated. In case $\mathbf{x}^{[h]}$ contains multiple missing values, consequent coefficients are not updated.

After a number of time steps, h_r, merging and deleting rules may help to keep the model updated. Merging neighbor granules, say γ^1 and γ^2, into a single granule formed by their convex hull, $\gamma^\Psi = \text{ch}(\gamma^1, \gamma^2)$, happens when they are placed close to each other so that γ^Ψ respects ρ and σ. Parameters of consequent functions of merged rules are obtained from

$$\alpha_j^\Psi = \frac{\alpha_j^1 + \alpha_j^2}{2},\ \ j = 0, \ldots, n, \tag{7}$$

and

$$\beta_{j\theta}^\Psi = \frac{\beta_{j\theta}^1 + \beta_{j\theta}^2}{2},\ \ j = 0, \ldots, \theta - 1, \theta + 1, \ldots, n;\ \forall\theta. \tag{8}$$

Concept changes may cause rules to become inactive. Rules are removed if they are not activated during h_r time steps. The value chosen for h_r depends on how long we want to keep inactive rules in the memory of the model.

Notice that if the number of attributes of an application is large such that the processing time of one sample is higher than the sampling rate, then there exist time-granulation [10] and online feature-selection methods [17] that can be used prior to eFGP. Otherwise, the eFGP parameters, h_r and ρ, can be made smaller and larger, respectively, so that wider granules, and the rule-merging and deleting procedures can keep a more compact summary of the data stream as a set of rules. Therefore, streaming data can be processed in a feasible amount of time by the more compact fuzzy model.

The learning procedure to evolve an eFGP model is given in Algorithm 1. Steps 3 and 15 suggest that samples are received and discarded one at a time, an essential feature in online environment. $|\theta|$ denotes the number of missing elements in a given $\mathbf{x}^{[h]}$. The resulting eFGP model is available at any time. The model is

Algorithm 1 eFGP Online Incremental Learning.

```
1: Define ρ, σ and h_r;
2: while 1 do
3:    Read input x^[h], h = 1, ...;
4:    if |θ| = 0 then
5:       Give prediction ŷ using complete functions p^i;
6:    else if |θ| = 1 then
7:       Give prediction ŷ using reduced-term functions q_θ^i;
8:    else if |θ| > 1 then
9:       Choose the most active rule for x^[h];
10:      Multiple imputation using the midpoints of the most active granule;
11:      Use complete functions p^i to give the prediction ŷ;
12:   end if
13:   Provide the granular prediction;
14:   Create rule or adapt the most active rule to accommodate (x, y)^[h];
15:   Delete sample (x, y)^[h];
16:   if h = zh_r, z = 1, ... then
17:      Merge neighbor granules;
18:      Delete inactive rules;
19:   end if
20: end while
```

robust to single and multiple MAR and MCAR data using a modified rule structure and an inherent (nonlinear and nonstationary) mechanism of learning and data imputation.

4. Application example

A Bitcoin dataset (bitcoincharts.com/charts) is used to evaluate eFGP. The lowest and highest prices during a day as well as closing prices of previous days are used to predict the closing price of the next trading day. 4-attribute daily data samples over a 4-year period (starting on Feb. 9, 2014) are considered. Models were constructed considering a percentage of missing values. The MAR cases are: 5% – 1% (which means 5% of chance that the value of a chosen attribute is missing, and 1% of chance that each of the remaining values is missing); 10% – 1%; 10% – 5%; 20% – 5%; 20% – 10%; 30% – 5%; and 30% – 10%.

The Bitcoin price is given in US Dollars at the end of the day. The minimum and maximum prices at the end of the day, observed from February 9, 2014 to February 8, 2018, are USD 171.41 and USD 19,187.78. We normalized (reescaled linearly) the data within [0,1]. Prediction accuracy is quantified using the Root Mean Square Error (*RMSE*) index [9],

$$RMSE = \frac{1}{H}\sum_{h=1}^{H}\sqrt{(\hat{y}^{[h]} - y^{[h]})^2}, \tag{9}$$

where H is the number of iterations. Given the current input sample, a prediction $\hat{y}^{[h]}$ of the Bitcoin price for the following day is provided. Then, the actual output value $y^{[h]}$ becomes available, and an incremental training step is given using the input-output pair. This learning approach is referred to as test-before-the-training learning approach [10]. There is no division of the data in training, validation, and test fractions. Test and training are continuously performed, sample per sample, on the fly. The *RMSE* is, therefore, calculated for test data only.

Average results for the Bitcoin price with different fractions of MCAR and MAR data considering 10 runs of eFGP are shown in Table 1. eFGP uses $\rho = \sigma = 0.15$, and $h_r = 50$. With the increase of the amount of missing data, the *RMSE* suggests that the estimation performance tends to reduce slightly; the number of rules tends to increase. MCAR data requires a greater number of rules whereas parametric adaptation prevails in MAR cases. The behavior of eFGP has been stable in the different scenarios.

Fig. 1 shows the numerical and granular eFGP estimates for the hardest, 30%, MCAR case. An accurate tracking of the actual data can be seen by using the eFGP pointwise estimate $\hat{y}$. Online learning and creation of new local granules were key points to keep a reasonable prediction accuracy as the currency achieved a quite high price followed by an unprecedented fall. Bounds of granules give a range of values around $\hat{y}$ (an outer approximation of the time-varying and unknown Bitcoin function) that may help decision making regarding Bitcoin exchanges and purchases.

Table 1
eFGP results for the Bitcoin price: MCAR and MAR data.

MCAR	RMSE	Mean # of rules
0%	0.0219 +/− 0.0044	14.4 +/− 0.0
1%	0.0225 +/− 0.0033	14.3 +/− 0.8
5%	0.0220 +/− 0.0023	14.1 +/− 0.2
10%	0.0213 +/− 0.0034	6.0 +/− 0.3
15%	0.0231 +/− 0.0047	8.2 +/− 0.4
20%	0.0234 +/− 0.0041	18.9 +/− 1.9
30%	0.0252 +/− 0.0054	16.3 +/− 1.8
MAR	**RMSE**	**Mean # of rules**
5% – 1%	0.0226 +/− 0.0037	8.8 +/− 0.1
10% – 1%	0.0229 +/− 0.0023	8.8 +/− 0.2
10% – 5%	0.0241 +/− 0.0054	8.1 +/− 0.1
20% – 5%	0.0237 +/− 0.0044	5.5 +/− 0.3
20% – 10%	0.0251 +/− 0.0039	5.8 +/− 0.5
30% – 5%	0.0244 +/− 0.0024	5.4 +/− 0.3
30% – 10%	0.0237 +/− 0.0035	10.9 +/− 0.4

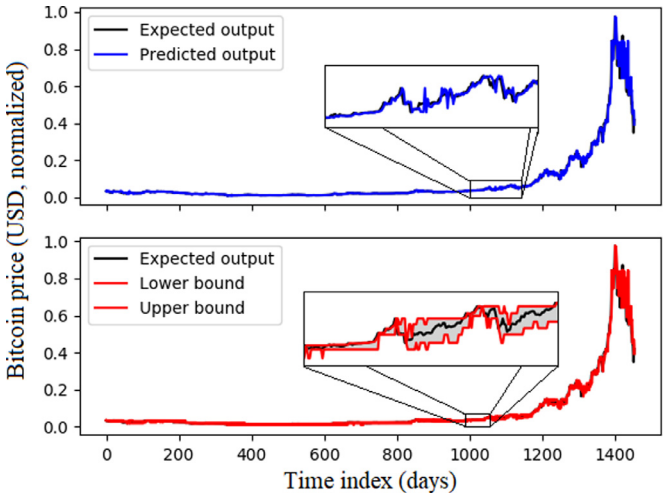

Fig. 1. Numerical and granular Bitcoin prediction.

eFGP was compared to an evolving Granular Neural Network (eGNN) [8] using min-max neurons, $\rho = 0.02$, $\zeta = 0.9$, and $h_r = 1500$; evolving Takagi–Sugeno (eTS) [1] using $\Omega = 350$, and $r = 0.4$; and extended Takagi–Sugeno (xTS) [1] employing $\Omega = 100$. These parameters provided the highest average-accuracy of each method based on 10 runs. As alternative methods are not supplied with mechanisms to impute missing data, a zero-order-hold approach is employed, i.e., the last prediction is replicated. Fig. 2 shows that eFGP is more robust to MCAR and MAR data across the range of analyzed values. Notice that eTS and xTS give the lowest average RMSE for the complete dataset in this application. However, removing a small percentage of values from the data stream is enough for eFGP to overcome them.

The average time spent by eFGP to process a sample on a Intel i7 3.6GHz processor with 16GB RAM using Python-Ubuntu 18.04 on Windows 10 was 9.0 milliseconds (ms). eGNN spent 10.9 ms; eTS consumed 9.8 ms; and xTS spent 8.4 ms. eFGP is competitive with the other methods in processing time.

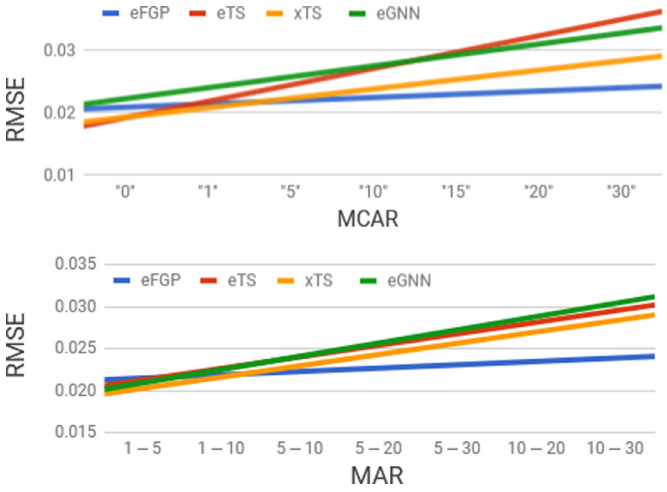

Fig. 2. Performance comparison on the prediction of the Bitcoin price.

5. Conclusion

We discussed missing values in time-varying data streams. We presented an evolving fuzzy model with modified rule structure, and an incremental learning algorithm for model construction, data imputation, and prediction in online settings where values may be missing at random and missing completely at random. eFGP deals with multiple missing data using reduced-term consequent functions and time-varying granules. Experimental results on cryptocurrency prediction considering from 1% to 30% of missing values have shown that eFGP overcomes other fuzzy and neuro-fuzzy evolving methods that utilize sample deletion and replication of the last output. A particular characteristic of eFGP concerns the provision of an enclosure of the output data, which may help decision making in a variety of applications. Further work will discuss missing-data imputation in semi-supervised classification of data streams.

Declaration of Competing Interest

The authors certify that they have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakersbureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

Acknowledgment

This work was supported by the Serrapilheira Institute (grant number Serra-1812-26777). Igor Škrjanc is grateful to the Slovenian Research Agency - Program P2-0219: Modeling, Simulation and Control.

References

[1] P. Angelov, Autonomous Learning Systems, Wiley, 2013.
[2] I. Aydilek, A. Arslan, A hybrid method for imputation of missing values using optimized fuzzy c-means with support vector regression and a genetic algorithm, Inf. Sci. 233 (2013) 25–35.
[3] I. Aydilek, A. Arslan, A novel hybrid approach to estimating missing values in databases using k-nearest neighbors and neural networks, Int. J. Innov. Comput. I 7 (8) (2012) 4705–4717.
[4] M. Das, S.K. Ghosh, Data-driven approaches for meteorological time series prediction: a comparative study of the state-of-the-art computational intelligence techniques, Pattern Recognit. Lett. 105 (2018) 155–164.
[5] T. Al-Khateeb, M. Masud, K. Al-Naami, S. Seker, A. Mustafa, L. Khan, Z. Trabelsi, C. Aggarwal, J. Han, Recurring and novel class detection using class-based ensemble for evolving data stream, IEEE Trans. Knowl. Data Eng. 28 (10) (2016) 2752–2764.
[6] V. Kuppusamy, I. Paramasivam, Grey fuzzy neural network-based hybrid model for missing data imputation in mixed database, Int. J. Intell. Eng. Syst. 10 (3) (2017) 146–155.
[7] D. Leite, R. Palhares, V. Campos, F. Gomide, Evolving granular fuzzy model-based control of nonlinear dynamic systems, IEEE Trans. Fuzzy Syst. 23 (4) (2015) 923–938.
[8] D. Leite, P. Costa, F. Gomide, Evolving granular neural networks from fuzzy data streams, Neural Netw. 38 (2013) 1–16.
[9] D. Leite, R. Ballini, P. Costa, F. Gomide, Evolving fuzzy granular modeling from nonstationary fuzzy data streams, Evol. Syst. 3 (2) (2012) 65–79.
[10] D. Leite, Evolving Granular Systems, State University of Campinas, School of Electrical and Computer Engineering, 2012 Phd thesis.
[11] R. Little, D. Rubin, Statistical Analysis with Missing Data, Wiley, NY, 1986.
[12] F. Lobato, C. Salesa, I. Araujo, V. Tadaiesky, L. Dias, L. Ramos, A. Santana, Multi-objective genetic algorithm for missing data imputation, Pattern Recognit. Lett. 68 (1) (2015) 126–131.
[13] J. Luengo, S. Garcia, F. Herrera, On the choice of the best imputation methods for missing values considering three groups of classification methods, Knowl. Inf. Syst. 32 (1) (2012) 77–108.
[14] T.-T.-H. Phan, E.P. Caillault, A. Lefebvre, A. Bigand, Dynamic time warping-based imputation for univariate time series data, Pattern Recognit. Lett. (2017). (In Press) Available online, Aug.
[15] R. Ranjan, R. Arya, S. Fernandes, E. Sravya, V. Jain, A fuzzy neural network approach for automatic k-complex detection in sleep EEG signal, Pattern Recognit. Lett. (2018). (In Press) Available online, Jan.
[16] P. Saravanan, P. Sailakshmi, Missing value imputation using fuzzy possibilistic c means optimized with support vector regression and genetic algorithm, J. Theor. Appl. Inf. Technol. 72 (1) (2015) 34–39.
[17] I. Škrjanc, J. Iglesias, A. Sanchis, D. Leite, E. Lughofer, F. Gomide, Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: a survey, Inf. Sci. 490 (2019) 344–368.