Article:

Ayala Solares, J.R., Wei, H. orcid.org/0000-0002-4704-7346 and Billings, S.A. (2019) A novel logistic-NARX model as a classifier for dynamic binary classification. Neural Computing & Applications, 31 (1). pp. 11-25. ISSN 0941-0643

https://doi.org/10.1007/s00521-017-2976-x

Noname manuscript No.
(will be inserted by the editor)

Accepted Manuscript by Neural Computing and Applications

A novel logistic-NARX model as a classifier for dynamic binary classification

Jose Roberto Ayala Solares · Hua-Liang Wei · Stephen A. Billings

the date of receipt and acceptance should be inserted later

Abstract System identification and data driven modeling techniques have seen ubiquitous applications in past decades. In particular, parametric modelling methodologies such as linear and nonlinear autoregressive with exogenous inputs models (ARX and NARX) and other similar and related model types have been preferably applied to handle diverse data driven modeling problems due to their easy-to-compute linear-in-the-parameters structure, which allows the resultant models to be easily interpreted. In recent years, several variations of the NARX methodology have been proposed that improve the performance of the original algorithm. Nevertheless, in most cases, NARX models are applied to regression problems where all output variables involve continuous or discrete-time sequences sampled from a continuous process, and little attention has been paid to classification problems where the output signal is a binary sequence. Therefore, we developed a novel classification algorithm that combines the NARX methodology with logistic regression and the proposed method is referred to as logistic-NARX model. Such a combination is advantageous since the NARX methodology helps to deal with the multicollinearity problem while the logistic regression produces a model that predicts categorical outcomes. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models where users can identify which input variables play an important role individually and/or interactively in the classification process, something that is not achievable using other classification techniques like random forests, support vector machines and k-nearest neighbors. The efficiency of the proposed method is tested with five case studies.

Keywords Nonlinear system identification · Dynamic systems · Binary classification · NARX models · Logistic regression

1 Introduction

System identification focuses on finding models from data and use them to understand or analyze the properties or behaviours of the underlying systems [1].

Linear models have been widely used in many applications [2]. However, its applicability is limited since most of the real world problems may not be well presented using linear models [3]. Research on nonlinear system identification has been carried out and advanced since the 1980s [1]. One of the most popular methodologies is the Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) methodology, which has proved to be suitable for a wide class of nonlinear systems [1, 4–8]. The NARMAX approach can detect an appropriate model structure and select the most important model terms from a dictionary consisting of a great number of candidate model terms.

In recent years, several variants have been proposed that improve the performance of the original algorithm. Such variations include the use of more complex and flexible predefined functions [6, 9–13], novel dependency

Jose Roberto Ayala Solares
Department of Automatic Control and Systems Engineering
Faculty of Engineering
The University of Sheffield
United Kingdom
E-mail: jrayalasolares1@sheffield.ac.uk

Hua-Liang Wei (🖂)
E-mail: w.hualiang@sheffield.ac.uk

Stephen A. Billings
E-mail: s.billings@sheffield.ac.uk

metrics [5, 8, 14–21] or different search mechanisms [22–26]. Nevertheless, the different versions of the NARX methodology have been designed under the assumption that the variables involved are continuous.

Many real-life systems involve a mixed combination of continuous and discrete variables. In this work, we focus on systems with binary responses that depend on continuous time predictors. Binary responses are commonly studied in many situations such as the presence or absence of a disease, granting a loan, or detecting the failure of a process, system or product [27, 28]. However, the use of traditional regression techniques to deal with systems with a dichotomous response variable may not be appropriate given that they are sensitive to outliers and the distribution of the classes [27].

In this work, we propose a novel approach that combines logistic regression with the NARX methodology. The main motivation comes from the fact that logistic regression models are more suitable for binary classification problems given that they provide probabilities of belonging or not to a particular class. One important consideration when constructing a logistic regression model is multicollinearity. In general, it is important to always check for high inter-correlations among the predictor variables. In the ideal scenario, the predictor variables will have a strong relationship to the dependent variable but should not be strongly related to each other [29]. This problem is adequately solved using the NARX approach, since the model terms selected are orthogonal (uncorrelated) to each other. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models, something that is not achievable using other popular classification techniques like random forests [30], support vector machines [31] and k-nearest neighbors [32].

This work is organised as follows. Section 2 includes a brief summary of nonlinear system identification and a discussion of the Orthogonal Forward Regression algorithm. In section 3 our new methodology is described. Section 4 presents three numerical case studies that show the effectiveness of our new method. In section 5 the logistic-NARX model is applied to two real applications. Section 6 discusses advantages and disadvantages of the technique. The work is concluded in section 7.

2 Nonlinear System Identification

System identification, as a data based modelling approach, aims to find a model from available data that can represent as close as possible the system input and output relationship [1, 2]. While conventionally linear models have been applied in many applications, its applicability is limited as the linearity assumption may be violated for many nonlinear system modelling problems [3]. Nonlinear system identification techniques have been advanced since the 1980s [1]. In particular, the Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX) methodology has proved to be a powerful tool for nonlinear system identification [1, 12, 33, 34].

In general, system identification consists of several steps, including data collection and processing, selection of mathematical representation, model structure selection, parameter estimation, and model validation [2]. **Data processing is an important part given that data preparation plays a key role when training a model. Generally, this consists of dealing with missing values and outliers, data normalization and transformation, dimensionality reduction, and performing feature engineering. In [32, 35], these issues are widely discussed. Regarding the selection of mathematical representation, this work focuses on NARX models.** Model structure detection has been tackled using different methods like clustering [36, 37], the Least Absolute Shrinkage and Selection Operator (LASSO) [38, 39], elastic nets [40, 41], genetic programming [42, 43], the Orthogonal Forward Regression (OFR) using the Error Reduction Ratio (ERR) approach [33], and the bagging methodology [21]. Parameter estimation has been performed using the traditional least squares method, gradient descent and the Metropolis-Hastings algorithm [44, 45]. Finally, for model validation, a set of statistical correlation tests have been developed in [46] and can be used to test and verify the validity of the identified nonlinear input-output models. In summary, system identification is a process that builds a parsimonious model that satisfies a set of accuracy and validity tests [10].

2.1 Orthogonal Forward Regression Algorithm

The NARX model is a nonlinear recursive difference equation with the following general form:

$$\begin{aligned} y(k) =& f\Big(y(k-1), \ldots, y(k-n_y), \\ & u(k-1), \ldots, u(k-n_u)\Big) + e(k) \end{aligned} \tag{1}$$

where $f(\cdot)$ represents an unknown nonlinear mapping, $y(k)$, $u(k)$ and $e(k)$ are the output, input and prediction error sequences with $k = 1, 2, \ldots, N$, and the maximum lags for the output and input sequences are n_y

and n_u [8]. Most approaches assume that the function $f(\cdot)$ can be approximated by a linear combination of a predefined set of functions $\phi_m\Big(\boldsymbol{\varphi}(k)\Big)$, therefore equation (1) can be expressed in a linear-in-the-parameters form

$$y(k) = \sum_{m=1}^{M} \theta_m \phi_m\Big(\boldsymbol{\varphi}(k)\Big) + e(k) \tag{2}$$

where θ_m are the coefficients to be estimated, $\phi_m\Big(\boldsymbol{\varphi}(k)\Big)$ are the predefined functions that depend on the regressor vector $\boldsymbol{\varphi}(k) = \Big[y(k-1), \ldots, y(k-n_y), u(k-1), \ldots, u(k-n_u)\Big]^T$ of past outputs and inputs, and M is the number of functions in the set.

The most popular algorithm for NARX modelling is the Orthogonal Forward Regression (OFR) algorithm. As a greedy algorithm [4], it adopts a recursive partitioning procedure [47] to fit a parsimonious NARX model which can be represented as a generalized dynamic linear regression problem [8, 48]. One of the most commonly used NARX models is the polynomial NARX representation, where equation (2) can be explicitly written as

$$\begin{aligned} y(k) = \theta_0 + \sum_{i_1=1}^{n} \theta_{i_1} x_{i_1}(k) + \sum_{i_1=1}^{n}\sum_{i_2=i_1}^{n} \theta_{i_1 i_2} x_{i_1}(k) x_{i_2}(k) + \cdots \\ + \sum_{i_1=1}^{n} \cdots \sum_{i_\ell=i_{\ell-1}}^{n} \theta_{i_1 i_2 \ldots i_\ell} x_{i_1}(k) x_{i_2}(k) \ldots x_{i_\ell}(k) + e(k) \end{aligned} \tag{3}$$

where

$$x_m(k) = \begin{cases} y(k-m) & 1 \le m \le n_y \\ u(k-m+n_y) & n_y+1 \le m \le n = n_y + n_u \end{cases} \tag{4}$$

and ℓ is the nonlinear degree of the model. A NARX model of order ℓ means that the order of each term in the model is not higher than ℓ. The total number of potential terms in a polynomial NARX model is given by

$$M = \binom{n+\ell}{\ell} = \frac{(n+\ell)!}{n! \cdot \ell!} \tag{5}$$

The OFR algorithm implements a stepwise regression to select the most relevant regressors, one at a time; it uses the error reduction ratio (ERR) as an index to measure the significance of each candidate model term [1]. The OFR algorithm can be used to transform a number of selected model terms to a set of orthogonal vectors, for which ERR can be evaluated by calculating the non-centralised squared correlation coefficient $C(\mathbf{x}, \mathbf{y})$ between two associated vectors $\mathbf{x}$ and $\mathbf{y}$ [5]

$$C(\mathbf{x}, \mathbf{y}) = \frac{\Big(\mathbf{x}^T \mathbf{y}\Big)^2}{(\mathbf{x}^T \mathbf{x})(\mathbf{y}^T \mathbf{y})} \tag{6}$$

In recent years, several variants of the algorithm have been proposed that modify the predefined functions, the dependency metric or the search mechanism in order to enhance its performance. In particular, an important issue is that the non-centralised squared correlation only detects linear dependencies. To solve this, new metrics have been implemented that are able to capture nonlinear dependencies [5, 8]. Some of these new metrics are entropy, mutual information [5, 8, 14, 15], simulation error [20] and distance correlation [21]. Furthermore, more complex predefined functions have been used recently like wavelets [9–12], radial basis functions [6, 13], and ridge basis functions [49], together with improved search mechanism like common model structure selection [22–24], iterative search [25], and incorporation of weak derivatives information [26].

Most of these variants are able to obtain good one-step ahead (OSA) predictions,

$$\begin{aligned} \hat{y}(k) = & f\Big(y(k-1), \ldots, y(k-n_y), \\ & u(k-1), \ldots, u(k-n_u)\Big) \end{aligned} \tag{7}$$

However, because the NARX model (1) depends on past outputs, a more reliable way to check the validity of the model is through the model predicted output (MPO) [50], which uses past predicted outputs to estimate future ones,

$$\begin{aligned} \hat{y}(k) = & f\Big(\hat{y}(k-1), \ldots, \hat{y}(k-n_y), \\ & u(k-1), \ldots, u(k-n_u)\Big) \end{aligned} \tag{8}$$

The MPO can provide details about the stability and predictability range of the model. In [51], the authors developed a lower bound error for the MPO of polynomial NARMAX models, which can be used to detect when a model's simulation is not reliable and needs to be rejected.

In the literature, some authors have adapted the original OFR algorithm to optimize directly the MPO in order to obtain a better long-term prediction, however these modified versions tend to be computationally expensive during the feature selection step **given that equation (8) needs to be evaluated N times (where N is the size of the training set) before computing a single value of the dependency metric, i.e. ERR, for the selection of each single model term [1,20]. Furthermore, these versions are not easily extendable to large model searches which are often necessary when dealing with real systems or MIMO system identification [1].**

3 Logistic-NARX Modelling Approach

Classification problems ubiquitously exist in all areas of science and engineering, where the aim is to identify a model that is able to classify observations or measurements into different categories or classes. Many methods and algorithms are available which include logistic regression [27,28], random forest [30], support vector machines [31] and k-nearest neighbors [32]. The latter three are very popular but their major drawback is that they remain as black boxes for which the interpretation of the models may not be straightforward. Although it is possible to obtain an importance index for the predictors in the model, this does not help in understanding the possible inner dynamics of a system. On the other hand, logistic regression is an approach that produces a model to predict categorical outcomes. The predicted values are probabilities and are therefore restricted to values between 0 and 1 [29]. Logistic regression uses the logistic function defined as,

$$f\left(x\right)=\frac{1}{1+\exp\left(-x\right)} \tag{9}$$

where x has an unlimited range, i.e. $x\in\mathbb{R}$, and $f\left(x\right)$ is restricted to range from 0 to 1 [28]. One issue with logistic regression models is that they require the model terms and the interactions between them to be specified beforehand. This is problematic since it is important to always check for high inter-correlations among the predictor variables. In the ideal scenario, the predictor variables will be strongly related to the dependent variable but not strongly related to each other in order to avoid the multicollinearity problem [29].

The new approach proposed in this paper combines the logistic function with the NARX representation in order to obtain a probability model

$$p\left(k\right)=\frac{1}{1+\exp\left[-\sum_{m=1}^{M}\theta_m\phi_m\left(\boldsymbol{\varphi}\left(k\right)\right)\right]} \tag{10}$$

For convenience, let us assume that the output sequence $y\left(k\right)$ can be either $y\left(k\right)=1$ or $y\left(k\right)=0$ for $k=1,2,\ldots,N$, where $y\left(k\right)=1$ denotes the occurrence of the event of interest. It is important to mention that in contrast with the original OFR algorithm presented in [52], that requires a threshold for the total of ERR, the user needs to specify the maximum number of terms n_{max} that the algorithm will look for [7]. Furthermore, traditionally the OFR algorithm relies on the ERR index (6) to determine the significance of a model term with respect to the output sequence. However, this metric is no longer useful given that the output is a binary sequence and the information from the class denoted as 0 would be lost. To overcome this issue, the biserial correlation coefficient is used, which measures the strength of the association between a continuous variable and a dichotomous variable [29]. The biserial correlation coefficient is defined as

$$r\left(\mathbf{x},\mathbf{y}\right)=\frac{\overline{X}_1-\overline{X}_0}{\sigma_X}\sqrt{\frac{n_1n_0}{N^2}} \tag{11}$$

where $\overline{X}_0$ is the mean value on the continuous variable X for all the observations that belong to class 0, $\overline{X}_1$ is the mean value of variable X for all the observations that belong to class 1, σ_X is the standard deviation of variable X, n_0 is the number of observations that belong to class 0, n_1 is the number of observations that belong to class 1, and N is the total number of data points.

The pseudo-code of the proposed algorithm is described in Algorithm 1, where regressors (model terms) are selected using an OFR algorithm, which has been widely discussed in the literature [1]. In Algorithm 1, lines from 1 to 4 aim to find candidate model terms that make most significant contributions in explaining the variation of the system output measured by the biserial correlation coefficient. Once found, lines 5-8 create a simple logistic model using the maximum likelihood estimation method, and assess its performance using a k-fold cross-validation accuracy. New candidate terms are orthogonalised with respect to the model terms already chosen using the modified Gram-Schmidt method, and evaluated using the biserial correlation coefficient. This process is repeated in lines 10-22 until it reaches a pre-specified maximum number n_{max} of model terms to be included in the final

Algorithm 1 Orthogonal Forward Regression for Logistic-NARX models

Input: Dictionary of regressor vectors $D = \{\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_M\}$, output signal $\mathbf{y}$, maximum number of terms n_{max}
Output: Logistic NARX model with significant terms selected from D and corresponding parameters $\boldsymbol{\theta}$ estimated

```
 1: for all $\boldsymbol{\phi}_i$ in $D$ do
 2:     Define $\mathbf{w}_i = \phi_i / \|\phi_i\|_2$
 3:     Compute $r^{(i)}(\mathbf{w}_i, \mathbf{y})$
 4: Find $j = \underset{1 \le i \le M}{max} \left\{ r^{(i)}(\mathbf{w}_i, \mathbf{y}) \right\}$
 5: Define $\mathbf{q}_1 = \mathbf{w}_j$
 6: Define $\mathbf{p}_1 = \boldsymbol{\phi}_j$
 7: Train a logistic regression model using $\mathbf{y}$ and $\mathbf{p}_1$
 8: Compute the k-fold cross validation accuracy and store it
 9: Remove $\boldsymbol{\phi}_j$ from $D$
10: for $s = 2$ to $n_{\max}$ do
11:     for all $\boldsymbol{\phi}_i$ in $D$ do
12:         Orthonormalize $\boldsymbol{\phi}_i$ with respect to $[\mathbf{q}_1, \ldots, \mathbf{q}_{s-1}]$ to obtain $\mathbf{w}_i$
13:         if $\mathbf{w}_i^T \mathbf{w}_i < 10^{-10}$ then
14:             Remove $\boldsymbol{\phi}_j$ from $D$
15:             Go to next iteration
16:         Compute $r^{(i)}(\mathbf{w}_i, \mathbf{y})$
17:     Find $j = \underset{1 \le i \le M-s-1}{max} \left\{ r^{(i)}(\mathbf{w}_i, \mathbf{y}) \right\}$
18:     Define $\mathbf{q}_s = \mathbf{w}_j$
19:     Define $\mathbf{p}_s = \boldsymbol{\phi}_j$
20:     Train a logistic regression model using $\mathbf{y}$ and $\mathbf{p}_1, \ldots, \mathbf{p}_s$
21:     Compute the k-fold cross validation accuracy and store it
22:     Remove $\boldsymbol{\phi}_j$ from $D$
23: Using the stored k-fold cross validation accuracies, select the most parsimonious model with $n \le n_{max}$ terms with the
    best accuracy performance
24: Return matrix of terms selected $\mathbf{P} = \left[ \mathbf{p}_1 \ \mathbf{p}_2 \ \ldots \ \mathbf{p}_n \right]$ and vector of coefficients $\boldsymbol{\theta} = \left[ \theta_1 \ \theta_2 \ \ldots \ \theta_n \right]^T$
```

model, **where** $n_{max} \le M$. Lines 13-15 are used to calculate the squared norm-2 of each candidate model term, based on which it decides if a candidate term should be excluded to avoid any potential ill-conditional issue. When a new model term is included, a logistic regression model is trained and the k-fold cross-validation accuracy is computed in lines 20-21. When the iteration reaches the specified number n_{max}, a parsimonious model consisting of a total of up to n_{max} model terms is then selected in line 23 based on the best cross-validation accuracy obtained. Finally, the algorithm returns the parameters $\boldsymbol{\theta}$ together with the selected model terms. **Given that the optimal number of model terms is not known in advance,** the parameter n_{max} can be selected heuristically, by running Algorithm 1 several times, and checking the resulted cross-validation accuracy curve. **In case the best model contains exactly n_{max} model terms, this means that the appropriate number of model terms may be beyond this value; therefore it could be increased to find a better model.**

The proposed algorithm combines the transparency and efficiency of the NARX models with logistic regression to deal with classification problems. This combination is advantageous since the NARX methodology helps to deal with the multicollinearity problem because of the orthogonalisation process that takes places. Furthermore, the NARX approach allows for the inclusion of lagged terms and interactions between them in a straight forward manner resulting in interpretable models, something that is not achievable using random forests, support vector machines and k-nearest neighbors.

The time complexity of the logistic-NARX method is determined by three main parts: the assessment of feature relevancy to the class label, the computation of the logistic regression model, and the orthogonalisation operations. Feature relevancy assessment has a linear time complexity of $O(NM)$, where N is the number of observations and M is the number of candidate features. The computation of the regression model has a worst-case time complexity of $O(M^3 + NM)$ [53], while the orthogonalisation procedure has a complexity of $O(N(M-1))$ [54]. As a result, the overall time complexity takes the order of $O(M^3 + NM)$.

4 Numerical Simulation Case Studies

In this section, **three simulation examples** are provided to illustrate the effectiveness of the new logistic-NARX methodology. Data are generated from three nonlinear models driven by lagged input variables, along with some noise. In all cases, we compare the performance of our algorithm with traditional classification techniques. For simplicity, we focus on polynomial NARX models described in equation (3), although our algorithm can be applied to other NARX models using wavelets [10, 12] or radial basis functions [6, 13].

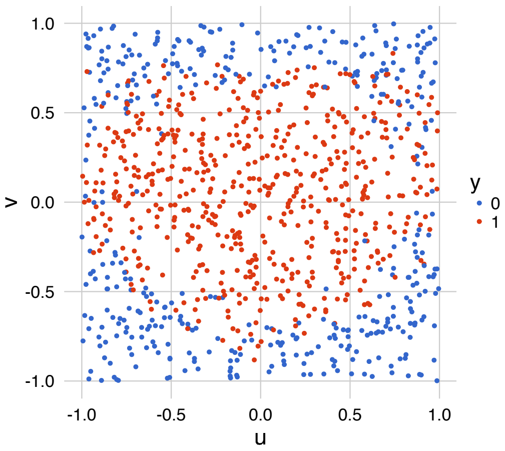

Fig. 1 Data points obtained from the input-output system given in equation (12)

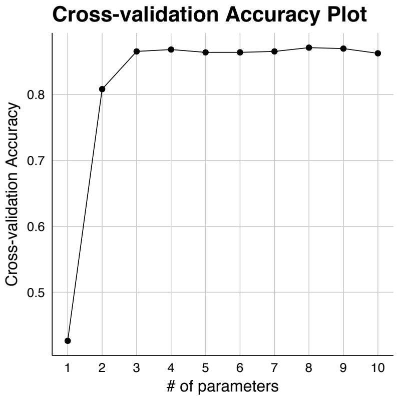

Fig. 2 Cross-validation accuracy plot obtained for (12) using Algorithm 1

4.1 Example 1

Assume we have the following input-output system:

$$y(k) = \begin{cases} 1 & \text{if } u^2(k) + 2v^2(k) \\ & \qquad -0.8u^2(k)\,v(k) + e(k) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where the inputs $u(k)$ and $v(k)$ are uniformly distributed between $[-1, 1]$, i.e. $u(k), v(k) \sim \mathcal{U}(-1, 1)$, and $e(k) \sim \mathcal{N}\left(0, 0.3^2\right)$. A total of 1000 input-output data points were collected. Plotting such points produces the figure shown in Fig. 1.

Most classification techniques are able to **perform static binary classification** with high accuracy. We apply our new algorithm to this dataset. The data is separated in a training set (700 points) and a testing set (300 points). Given that this is a static problem, no lags are used, and the nonlinear degree is chosen as $\ell = 3$, which results in a search space with 10 model terms. Therefore, the maximum number of terms is selected as $n_{max} = 10$, and 10 folds are used to compute the cross-validation accuracy. Fig. 2 shows the cross-validation accuracy plot obtained after applying Algorithm 1 and it suggests that no significant improvement is obtained in accuracy with models that have more than 4 models terms. Therefore, a model with 4 terms is chosen and these are shown in Table 1. Such results show that our algorithm was able to identify correctly all model terms involved in the decision boundary for (12). The parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function.

For comparison purposes, a regression-like NARX model based on the approach suggested in [7], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the **same** training set. All models are compared using the testing set and the classification accuracy.

Table 1 Identified model terms for (12) using Algorithm 1

Model Term	Parameter
$v^2(k)$	-12.297
constant	6.459
$u^2(k)$	-6.632
$u^2(k)\,v(k)$	4.470

Table 2 Comparison of accuracy performance between different methods for modelling of (12)

Method	Classification accuracy
Logistic-NARX	**0.8829**
Regression NARX	0.8763
Random Forest	0.8729
Support Vector Machine	0.8796
K-Nearest Neighbors	0.8428

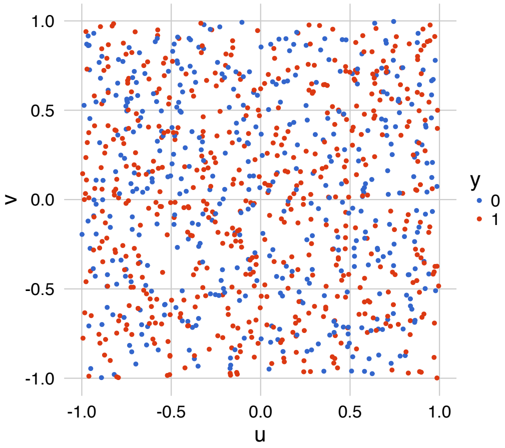

Fig. 3 Data points obtained from the input-output system given in equation (13)

The results are shown in Table 2. It can be seen that our new method has a comparable performance with the rest of the techniques, making it a feasible alternative for static binary classification problems.

4.2 Example 2

Let us consider a slightly different version of equation (12) as follows:

$$y(k) = \begin{cases} 1 & \text{if } u^2(k-1) + 2v^2(k-2) \\ & \quad -0.8u^2(k-2)\,v(k-1) + e(k) < 1 \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Plotting again the 1000 data points results in the figure shown in Fig. 3. As it can be observed, there is not a clear boundary between the two classes as in Fig. 1. This is a problem as it can be wrongly suggested that the two classes cannot be separated.

We apply our new algorithm to this dataset. The data is separated in a training set (the first 700 points) and a testing set (the last 300 points). The maximum lags for the inputs and output are chosen to be $n_u = n_y = 4$, and the nonlinear degree is $\ell = 3$, which results in a search space with 165 model terms. The maximum number of terms is selected as $n_{max} = 10$, and 10 folds are used to compute the cross-validation accuracy. Fig. 4 shows the cross-validation accuracy plot obtained after applying Algorithm 1 and it suggests that the most parsimonious model with the best accuracy has 4 models terms. These are shown in Table 3. Such results show that our algorithm was able to identify correctly all model terms involved in the decision boundary for (13). Again, the parameters obtained are log odds ratios, therefore they do not necessarily need to resemble the ones in the decision boundary function.

For comparison purposes, a regression-like NARX model based on the approach suggested in [7], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are

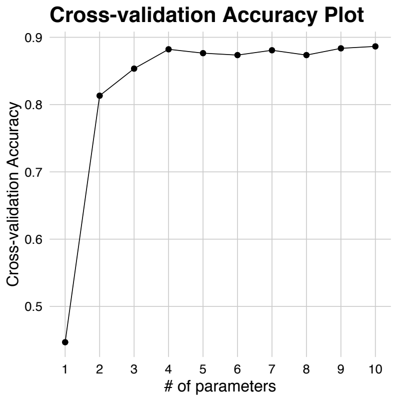

Fig. 4 Cross-validation accuracy plot obtained for (13) using Algorithm 1

Table 3 Identified model terms for (13) using Algorithm 1

Model Term	Parameter
$v^2(k-2)$	-12.508
constant	6.155
$u^2(k-1)$	-6.086
$u^2(k-2)\,v(k-1)$	4.582

Table 4 Comparison of accuracy performance between different methods for modelling of (13)

Method	**Classification accuracy**
Logistic-NARX	**0.8581**
Regression NARX	**0.8581**
Random Forest (without autoregressive inputs)	0.5034
Support Vector Machine (without autoregressive inputs)	0.5574
K-Nearest Neighbors (without autoregressive inputs)	0.5267
Random Forest (with autoregressive inputs)	0.8514
Support Vector Machine (with autoregressive inputs)	0.777
K-Nearest Neighbors (with autoregressive inputs)	0.6284

trained with the **same** training set. **In general, traditional classification techniques do not consider lagged variables unless these are explicitly included, therefore** two cases are considered: the first case assumes that no autoregressive terms are available, therefore only $u(k)$ and $v(k)$ are used. In the second one, **the same lagged input and output variables that were considered for the logistic-NARX model** are used with the maximum lags chosen to be $n_u = n_y = 4$ (the regression-like NARX model only considers the second case). All models are compared using the testing set and the OSA accuracy. The results are shown in Table 4. It can be seen that our new method has the best accuracy performance together with the regression-like NARX model. This is expected given the NARX-like structure that generates the data. Nevertheless, the regression-like NARX model produces real-valued outputs, which make them difficult to interpret for classification. On the other hand, the logistic-NARX model is preferred because its outputs are restricted to range from 0 to 1, and they can be used as classification probabilities. Furthermore, the random forest, support vector machine and k-nearest neighbors models are not able to generate reliable results if lagged variables (i.e. values observed in some previous time instants) are not taken into account when defining the feature vector, however their performance is increased when the autoregressive input variables are included. Although it may be argued that our method is just slightly better than the random forest with autoregressive inputs, it must be taken into consideration that the logistic NARX model is transparent and the role or contribution of individual regressors can be known.

4.3 Example 3

Assume we have the following input-output system:

$$y(k) = \begin{cases} 1 & \text{if } -u(k-1)\sqrt{|v(k-1)|} \\ & \qquad +0.5u^3(k-1) \\ & \qquad +\sin(v(k-2)) + e(k) < 0 \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where the inputs $u(k)$ and $v(k)$ are uniformly distributed between $[-1,1]$, i.e. $u(k), v(k) \sim \mathcal{U}(-1,1)$, the error sequence is given by $e(k) = w(k) + 0.3w(k-1) + 0.6w(k-2)$ and $w(k)$ is normally distributed with zero mean and variance of 0.01, i.e. $w(k) \sim \mathcal{N}(0, 0.01)$. A total of 1000 input-output data points were collected.

We apply our new algorithm to this dataset. The data is separated in a training set (the first 700 points) and a testing set (the last 300 points). The maximum lags for the inputs and output are chosen to be $n_u = n_y = 4$, and the nonlinear degree is $\ell = 3$, which results in a search space with 165 model terms. The maximum number of terms is selected as $n_{max} = 10$, and 10 folds are used to compute the cross-validation accuracy. Fig. 5 shows the cross-validation accuracy plot obtained after applying Algorithm 1 and it suggests that the most parsimonious model with the best accuracy has 8 models terms. These are shown in Table 5.

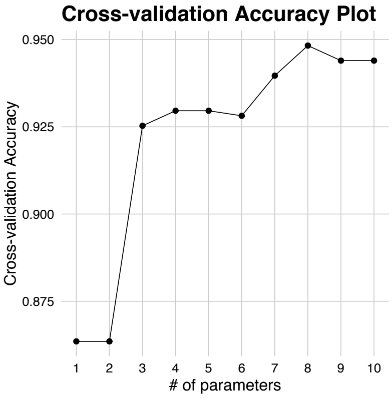

Fig. 5 Cross-validation accuracy plot obtained for (14) using Algorithm 1

Table 5 Identified model terms for (14) using Algorithm 1

Model Term	Parameter
$v(k-2)$	-12.755
constant	0.224
$u(k-1)v^2(k-1)$	8.488
$v^3(k-2)$	-15.323
$u(k-1)v^2(k-2)$	10.066
$u(k-1)$	9.047
$u^3(k-1)$	-8.715
$u(k-1)u^2(k-4)$	-3.285

Similarly to the previous case study, a regression-like NARX model based on the approach suggested in [7], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the **same** training set. **Again, two** cases are considered: the first case assumes that no autoregressive terms are available, therefore only $u(k)$ and $v(k)$ are used. In the second one, **the same lagged input and output variables that were considered for the logistic-NARX model** are used with the maximum lags chosen to be $n_u = n_y = 4$, and the nonlinear degree is $\ell = 3$, which results in a search space with 165 model terms. All models are compared using the testing set and the OSA accuracy. The results are shown in Table 6. It can be seen that our new method has the best accuracy performance, with a very similar result to the regression-like NARX model and the random forest with autoregressive inputs. Once more, the advantage over the random forest models is the transparency and interpretability about the role or contribution of individual regressors. Also, the advantage over the regression-like NARX model is a more interpretable output that is easily related to a classification probability.

5 Application to Real Data

In this section, two real scenarios are presented where the methodology is applied to the detection of cancerous cells in a breast cancer dataset [55–57], and the detection of human eye blinking using an electroencephalogram dataset [57]. Again, we compare the performance of our algorithm with traditional classification techniques. For simplicity, we focus on polynomial NARX models described in equation (3).

Table 6 Comparison of accuracy performance between different methods for modelling of (14)

Method	Classification accuracy
Logistic-NARX	**0.9392**
Regression NARX	0.9358
Random Forest (without autoregressive inputs)	0.527
Support Vector Machine (without autoregressive inputs)	0.4932
K-Nearest Neighbors (without autoregressive inputs)	0.47
Random Forest (with autoregressive inputs)	0.9223
Support Vector Machine (with autoregressive inputs)	0.8986
K-Nearest Neighbors (with autoregressive inputs)	0.7973

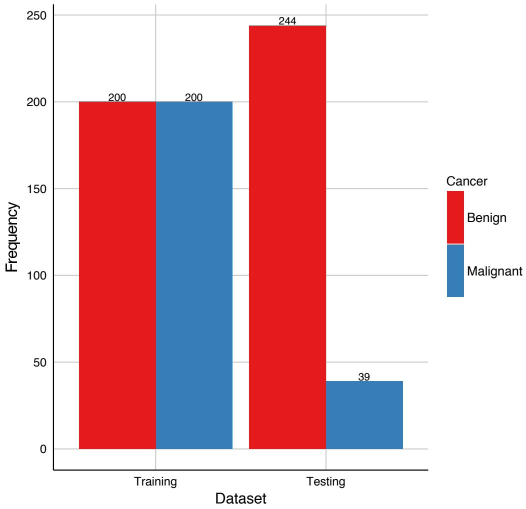

Fig. 6 Frequency of each cancer type for the training and testing sets.

5.1 Breast Cancer Classification

Breast cancer is the most common cancer in women worldwide [56]. Among the different prevention and control techniques, early detection is still the best method in order to improve breast cancer outcome and survival [58]. For this case study, we use the breast cancer dataset from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg [57]. This dataset contains 699 instances with the following 10 attributes:

- ID number
- Clump thickness (integer value between 1 and 10)
- Uniformity of cell size (integer value between 1 and 10)
- Uniformity of cell shape (integer value between 1 and 10)
- Marginal adhesion (integer value between 1 and 10)
- Single epithelial cell size (integer value between 1 and 10)
- Bare nuclei (integer value between 1 and 10)
- Bland chromatin (integer value between 1 and 10)
- Normal nucleoli (integer value between 1 and 10)
- Mitoses (integer value between 1 and 10)
- Class (2 for benign, 4 for malignant)

The bare nuclei attribute contains 16 missing values. Such instances were removed from the analysis. Also, the ID number attribute does not provide any meaningful information for the classification task, so it is removed from the dataset. The class attribute is recoded with ’0’ for a benign case and ’1’ for a malignant. The rest of the attributes are divided by 10 in order to have feature values ranging from 0.1 to 1.

The data is separated in a training set (400 instances with 200 samples from each class) and a testing set (283 instances). The frequency of the class for each set is shown in Figure 6 where it can be noticed that each cancer type has the same frequency in the training set, however, this is not the case in the testing set. Nevertheless, this is not a significant issue as the training phase has access to a good balance of the two classes that need to be identified, while the imbalanced testing set can be used to check the performance of the trained model.

Given that this is a static problem, no lags are used, and the nonlinear degree is chosen as $\ell = 2$ based on [33]. These results in a search space with 55 model terms. Therefore, the maximum number of terms to search is selected as $n_{max} = 10$, and 10 folds are used to compute the cross-validation accuracy. Fig. 2 shows the cross-validation

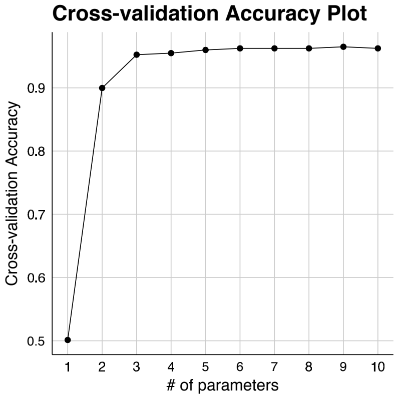

Fig. 7 Cross-validation accuracy plot obtained for the Breast Cancer dataset using Algorithm 1

accuracy plot obtained after applying Algorithm 1 and it suggests that no significant improvement is obtained in accuracy with models that have more than 3 models terms. Therefore, a model with 3 terms is chosen and these are shown in Table 7.

Table 7 Identified model terms for the Breast Cancer dataset using Algorithm 1

Model Term	Parameter
Bare nuclei	6.430
constant	-5.774
Uniformity of cell size	11.338

Once more, a regression-like NARX model based on the approach suggested in [7], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the **same** training set. All models are compared using the testing set and the classification accuracy. The results are shown in Table 8. All the methods are able to obtain a good classification accuracy. Although the logistic NARX has not the best accuracy, the difference with the best ones is negligible. This makes the logistic NARX model a competitive alternative to other classification techniques.

Table 8 Comparison of accuracy performance between different methods for modelling of the Cancer Breast dataset

Method	**Classification accuracy**
Logistic-NARX	0.9716
Regression NARX	**0.9787**
Random Forest	**0.9787**
Support Vector Machine	0.9681
K-Nearest Neighbors	0.9716

5.2 Electroencephalography Eye State Identification

Recently, electroencephalography eye state classification has become a popular research topic with several applications in areas like stress features identification, epileptic seizure detection, human eye blinking detection, among others [59]. For this case study, we use the EEG Eye State dataset found at the UCI Machine Learning Repository [57]. This dataset contains 14,980 EEG measurements from 14 different variables taken with the Emotiv EEG neuroheadset during 117 seconds. The eye state of the patient was detected with the aid of a camera during the experiment. If the eye is closed, it is coded as a '1', otherwise it is coded as '0'.

For this analysis, the first 80% of the dataset is used for training, while the rest is used for testing. The frequency of the eye state for each dataset is shown in Figure 8. Similar to the breast cancer scenario, the two eye states have roughly the same frequency in the training set, however, this is not the case in the testing set. Once more, this is not a significant issue as the training phase can be performed with enough information from both eye states, while the imbalanced testing set can be used to check the performance of the trained model.

Furthermore, two preprocessing steps are performed in the training set. First of all, several outliers are detected within the 14 variables. **These were detected using data visualization techniques (i.e. boxplots, histograms and line plots) and summary statistics on each of the variables.** The outliers are replaced

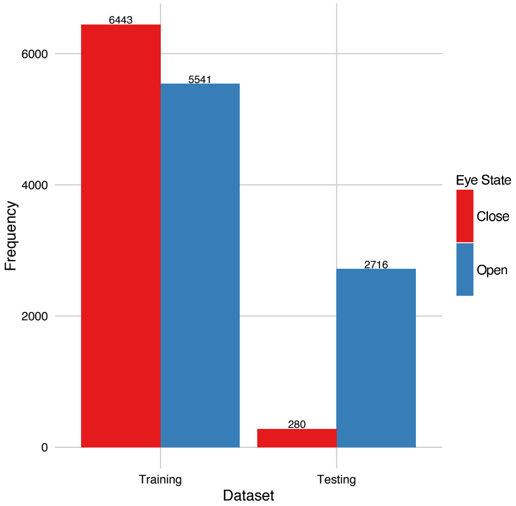

Fig. 8 Frequency of the eye state for the training and testing sets.

with the mean value of the remaining measurements for each variable. The eye state time series, together with the 14 cleaned variables, are shown in Figure 9. Second, **an attempt to train a model using the original dataset was done. However,** given the high variability and dependency between the variables measured, the model does not perform well enough. **Because of this**, a principal component analysis (PCA) is performed in order to reduce the dimensionality of both the data and model space, and in this study the 5 most important principal components (PCs) were used to represent the features of the original data. The PC time series are shown in Figure 10. Each PC is treated to be a new input variable; lagged PC variables were then used to built a logistic-NARX model. For this analysis, the variables are transformed using scaling, centering and Box-Cox transformations. Therefore, the PCs summarise the main variability of the dataset and simplify the identification process. The preprocessing parameters obtained during the training phases are directly used on the testing set in order to avoid the data snooping problem.

We apply the logistic-NARX modelling approach to this dataset. The output variable is the eye state signal, and the input variables are the 5 PCs computed in the preprocessing phase. **For this scenario, no lagged variables of the output signal are used in order to ensure that the model captures a pattern with the exogenous inputs only.** The maximum lag for the inputs is chosen to be $n_u = 50$, and the nonlinear degree is $\ell = 1$ **based on the results of previous works in** [33,59]. The search space is made up of 251 model terms. The maximum number of terms to look for is chosen as $n_{max} = 30$, and 10 folds are used to compute the cross-validation accuracy. Fig. 11 shows the cross-validation accuracy plot obtained after applying Algorithm 1 and it suggests that the most parsimonious model with the best accuracy has 9 models terms. These are shown in Table 9.

Table 9 Identified model terms for the EEG Eye State dataset using Algorithm 1

Model Term	Parameter
$PC_2\,(k-43)$	0.1545
constant	0.2123
$PC_3\,(k-50)$	0.5776
$PC_1\,(k-43)$	-0.1384
$PC_2\,(k-1)$	-0.2593
$PC_2\,(k-38)$	0.1766
$PC_2\,(k-50)$	0.3606
$PC_3\,(k-1)$	-0.1214
$PC_2\,(k-32)$	0.1536

In order to assess the performance of the resultant logistic NARX model, a regression-like NARX model based on the approach suggested in [7], a random forest with 500 trees, a support vector machine with a radial basis kernel, and a k-nearest neighbors model are trained with the **same** training set. **Similar to the previous examples, two cases are considered. One were the current input values are used, i.e. without lags, and another were** the same lagged variables that were used for the logistic NARX model **are employed**. All models are compared using the testing set and the OSA accuracy. The results are shown in Table 10. In this case, our method has the best accuracy performance and identifies the most significant lagged PCs that

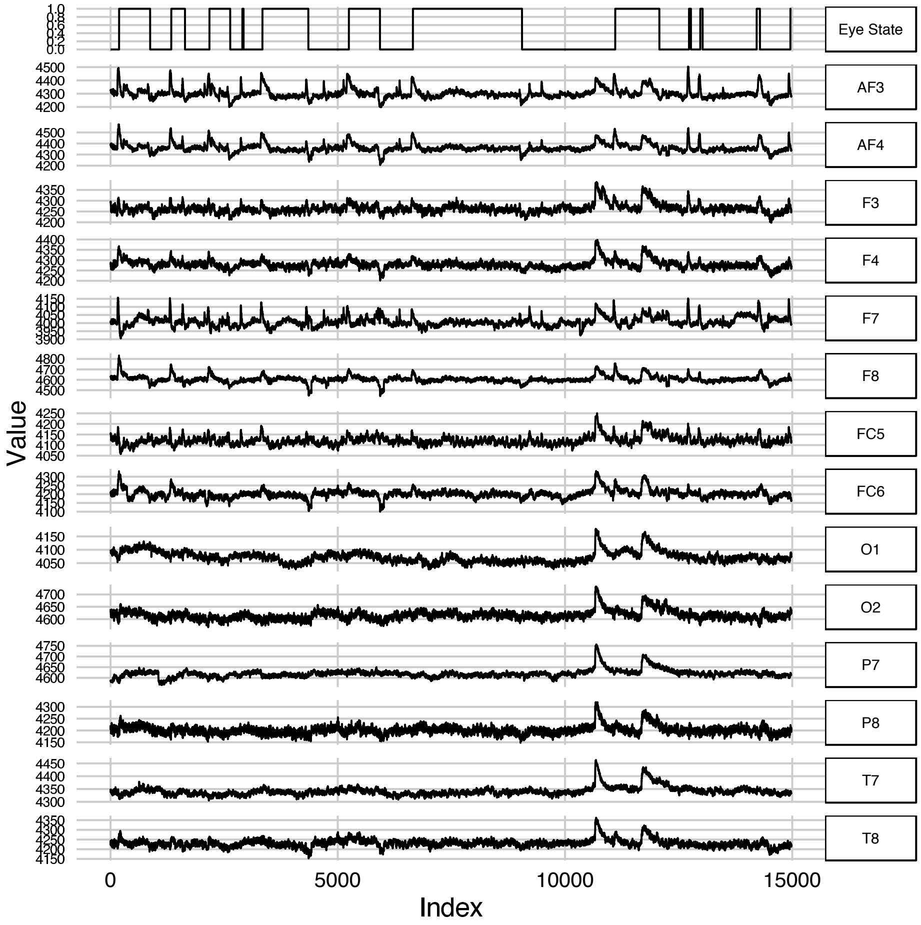

Fig. 9 Time series of all variables in the EEG Eye State dataset found at the UCI Machine Learning Repository [57].

Table 10 Comparison of accuracy performance between different methods for modelling of the EEG Eye State dataset

Method	Classification accuracy
Logistic-NARX	**0.7199**
Regression NARX	0.6643
Random Forest (without autoregressive inputs)	0.5475
Support Vector Machine (without autoregressive inputs)	0.6029
K-Nearest Neighbors (without autoregressive inputs)	0.5041
Random Forest (with autoregressive inputs)	0.6365
Support Vector Machine (with autoregressive inputs)	0.6473
K-Nearest Neighbors (with autoregressive inputs)	0.5662

contribute to the classification of the eye state. The models that are trained without autoregressive inputs have a poor classification accuracy. This is improved when autoregressive information is included. However, they do not achieve a classification accuracy like the one obtained by the logistic NARX model.

6 Discussion

The proposed logistic-NARX algorithm shows a new approach to deal with classification problems. The new method has a similar performance with other classification techniques when dealing with static data, but it outperforms other methods when there is a dynamic component, and lagged versions of input and output variables are required. Our algorithm is able to produce interpretable models where the contribution of each model term can be analysed. It is superior to random forests, support vector machines and k-nearest neighbors; these techniques are considered black-box models, so little information about the inner dynamics of the model can be obtained. Additionally, when using random forests, it is possible to get the variable importance, which may provide some insight about which variables are contributing the most to explain the output. However, this only ranks the variables, and does not explain how the variables are contributing to the modeling process.

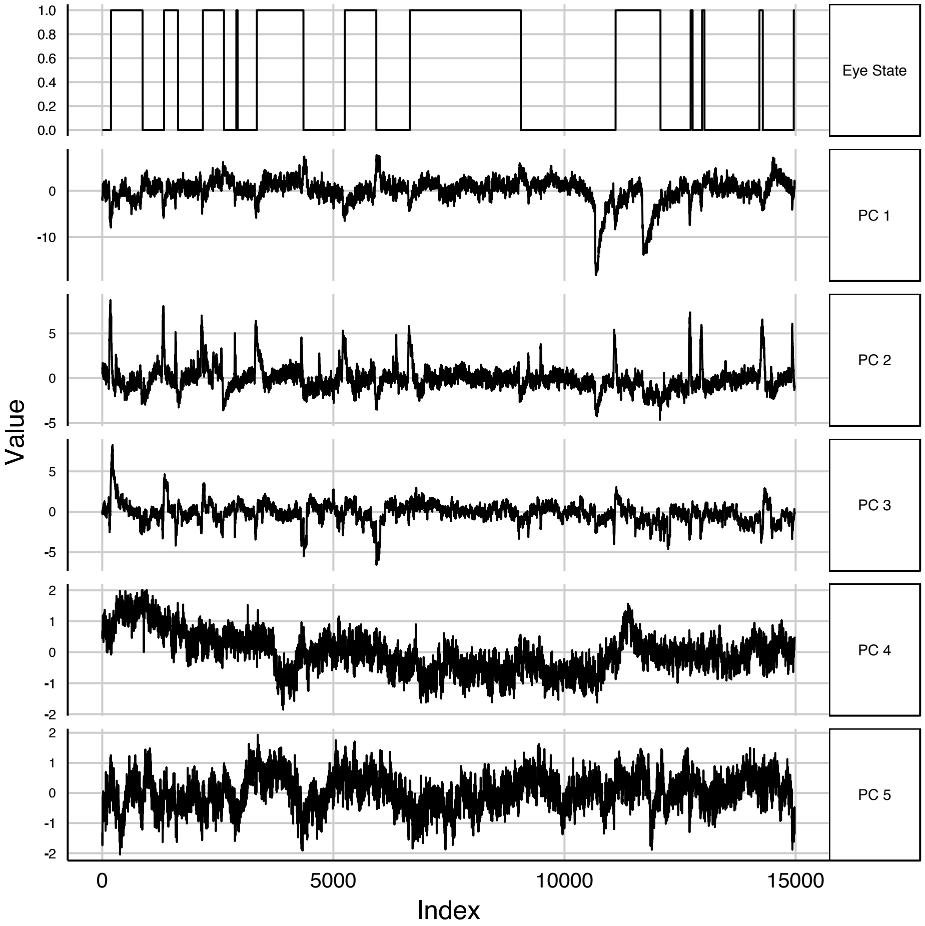

Fig. 10 Time series of the 5 most important principal components.

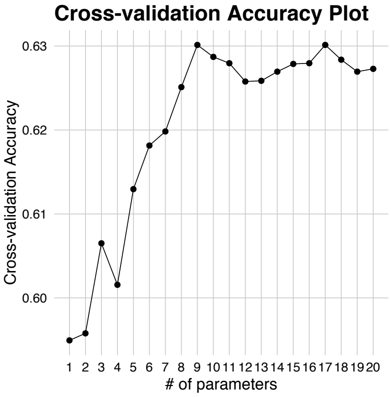

Fig. 11 Cross-validation accuracy plot obtained for the EEG Eye State dataset using Algorithm 1

The logistic-NARX model overcomes this by providing variable importance and interpretability about how the variables are interacting. Nevertheless, there are some limitations to the proposed algorithm. First of all, this work focuses on polynomial-like structures, therefore, severe nonlinearities may not be modeled properly. To overcome this, other structures can be considered (e.g. radial basis functions, wavelets), and this will be considered in a future extension of this work. Another issue, is the selection of the maximum lags for the output and input sequences (n_y and n_u). This is an open research problem where several interesting approaches have been proposed to tackle it [33, 37]. It would become more difficult when the lags become large, as the model search space has a factorial growth which makes it intractable. Also, the logistic-NARX approach may be affected by severe correlation between the inputs, which results in poor performance models. Some alternatives to overcome this include the iterative OFR [25] and the ultra OFR [26]. Finally,

the performance of the logistic-NARX model can be affected if the data are not balanced (especially when the output data are imbalanced). The scenario of imbalanced data is typical in many real applications where the minority class is dominated or buried by the majority class. Several approaches are available for dealing with imbalanced data problem, readers are referred to [32, 60] for details.

7 Conclusion

In this work we developed a novel algorithm that combines logistic regression with the NARX methodology. This allows to tackle classification problems where the output signal is a binary sequence and the regressors are continuous lagged variables. Our approach can deal with the multicollinearity problem while producing models that predicts binary outcomes. From the five case studies, the performance of the proposed logistic NARX models is preferable to that of the other compared methods when dealing with binary-label prediction, where it is sometimes highly desirable to know which input variables play an important role individually and/or interactively in the classification process. The results obtained are promising, and future research may extend this method to multi-class problems.

8 Acknowledgement

The authors acknowledge the financial support to J. R. Ayala Solares from the University of Sheffield and the Mexican National Council of Science and Technology (CONACYT). The authors gratefully acknowledge that part of this work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/I011056/1 and Platform Grant EP/H00453X/1, and ERC Horizon 2020 Research and Innovation Action Framework Programme under Grant No 637302 (PROGRESS).

References

1. S.A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains* (Wiley, 2013)
2. T. Söderström, P. Stoica, *System Identification* (Prentice Hall, 1989)
3. K.J. Pope, P.J.W. Rayner, in *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on*, vol. IV (1994), vol. IV, pp. 457 – 460
4. S.A. Billings, S. Chen, R.J. Backhouse, Mechanical Systems and Signal Processing **3**(2), 123 (1989)
5. S.A. Billings, H.L. Wei, IEEE Transactions on Neural Networks **18**(1), 306 (2007)
6. H.L. Wei, D.Q. Zhu, S.A. Billings, M.A. Balikhin, Advances in Space Research **40**(12), 1863 (2007). URL http://www.sciencedirect.com/science/article/pii/S0273117707002086
7. S.A. Billings, H.L. Wei, International Journal of Control **81**(5), 714 (2008)
8. H.L. Wei, S.A. Billings, International Journal of Modelling, Identification and Control **3**(4), 341 (2008)
9. A.K. Alexandridis, A.D. Zapranis, Neural Networks **42**(0), 1 (2013). DOI http://dx.doi.org/10.1016/j.neunet.2013.01.008. URL http://www.sciencedirect.com/science/article/pii/S0893608013000129
10. S.A. Billings, H.L. Wei, International Journal of Systems Science **36**(3), 137 (2005)
11. S.A. Billings, H.L. Wei, Neural Networks, IEEE Transactions on **16**(4), 862 (2005)
12. H.L. Wei, S.A. Billings, Y. Zhao, L. Guo, Neural Networks, IEEE Transactions on **20**(1), 181 (2009)
13. S.A. Billings, H.L. Wei, M.A. Balikhin, Neural Networks **20**(10), 1081 (2007). URL http://www.sciencedirect.com/science/article/pii/S0893608007001876
14. D. Koller, M. Sahami, in *In 13th International Conference on Machine Learning* (1995)
15. S. Wang, H.L. Wei, D. Coca, S.A. Billings, International Journal of Systems Science **44**(2), 223 (2013)
16. T. Speed, Science **334**(6062), 1502 (2011). DOI 10.1126/science.1215894. URL http://www.sciencemag.org/content/334/6062/1502.short
17. D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Science **334**(6062), 1518 (2011). DOI 10.1126/science.1205438. URL http://www.sciencemag.org/content/334/6062/1518.abstract
18. G.J. Székely, M.L. Rizzo, N.K. Bakirov, The Annals of Statistics **35**(6), 2769 (2007)
19. G.J. Székely, M.L. Rizzo, Journal of Statistical Planning and Inference **143**(8), 1249 (2013)
20. L. Piroddi, W. Spinelli, International Journal of Control **76**(17), 1767 (2003). URL http://dx.doi.org/10.1080/00207170310001635419
21. J. Ayala Solares, H.L. Wei, Nonlinear Dynamics pp. 1–15 (2015). URL http://dx.doi.org/10.1007/s11071-015-2149-3
22. H.L. Wei, S.A. Billings, International Journal of Modelling, Identification and Control **5**(2), 93 (2008)
23. P. Li, H.L. Wei, S.A. Billings, M.A. Balikhin, R. Boynton, Journal of Computational and Nonlinear Dynamics **8**(4), 10 (2013)
24. Y. Li, H.L. Wei, S.A. Billings, P. Sarrigiannis, International Journal of Systems Science pp. 1–11 (2015). URL http://dx.doi.org/10.1080/00207721.2015.1014448
25. Y. Guo, L. Guo, S. Billings, H.L. Wei, International Journal of Systems Science **46**(5), 776 (2015). DOI 10.1080/00207721.2014.981237. URL http://dx.doi.org/10.1080/00207721.2014.981237
26. Y. Guo, L.Z. Guo, S.A. Billings, H.L. Wei, Neurocomputing (2015). URL http://www.sciencedirect.com/science/article/pii/S0925231215011741

27. G. James, D. Witten, T. Hastie, R. Tibshirani, *An Introduction to Statistical Learning with Application in R*, *Springer Texts in Statistics*, vol. 103 (Springer, 2013)
28. F. Harrell, *Regression Modeling Strategies: with Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis* (Springer, 2015)
29. J. Pallant, *SPSS Survival Manual* (McGraw-Hill Education (UK), 2013)
30. L. Breiman, Machine Learning **45**(1), 5 (2001). URL http://dx.doi.org/10.1023/A%3A1010933404324
31. V.N. Vapnik, *Statistical Learning Theory* (Wiley, 1998)
32. M. Kuhn, K. Johnson, *Applied Predictive Modeling* (Springer, 2013)
33. H.L. Wei, S.A. Billings, J. Liu, International Journal of Control **77**(1), 86 (2004)
34. M.T. Rashid, M. Frasca, A.A. Ali, R.S. Ali, L. Fortuna, M.G. Xibilia, Nonlinear Dynamics **69**(4), 2237 (2012). DOI 10.1007/s11071-012-0422-2. URL http://dx.doi.org/10.1007/s11071-012-0422-2
35. H. Wickham, G. Grolemund, *R for Data Science* (O'Reilly Media, 2016)
36. L.A. Aguirre, C. Jácôme, in *Control Theory and Applications, IEE Proceedings-*, vol. 145 (IET, 1998), vol. 145, pp. 409–414
37. B. Feil, J. Abonyi, F. Szeifert, Journal of Process Control **14**(6), 593 (2004)
38. S.L. Kukreja, J. Lofberg, M.J. Brenner, in *System Identification*, vol. 14 (2006), vol. 14, pp. 814–819
39. P. Qin, R. Nishii, Z.J. Yang, Nonlinear Dynamics **70**(3), 1831 (2012). DOI 10.1007/s11071-012-0576-y. URL http://dx.doi.org/10.1007/s11071-012-0576-y
40. H. Zou, T. Hastie, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67**(2), 301 (2005)
41. X. Hong, S. Chen, in *16th IFAC Symposium on System Identification* (2012), pp. 1814–1819
42. S. Sette, L. Boullart, Engineering Applications of Artificial Intelligence **14**(6), 727 (2001)
43. J. Madár, J. Abonyi, F. Szeifert, Industrial & Engineering Chemistry Research **44**(9), 3178 (2005)
44. T. Baldacchino, S.R. Anderson, V. Kadirkamanathan, Automatica **48**(5), 857 (2012)
45. B.O. Teixeira, L.A. Aguirre, Journal of Process Control **21**(1), 82 (2011)
46. S.A. Billings, W.S.F. Voon, International Journal of Control **44**(1), 235 (1986)
47. T.G. Dietterich, in *Structural, Syntactic, and Statistical Pattern Recognition* (Springer, 2002), pp. 15–30
48. L.A. Aguirre, C. Letellier, Mathematical Problems in Engineering **2009**(35) (2009)
49. H.L. Wei, M.A. Balikhin, S.N. Walker, in *Computer Science & Education (ICCSE), 2015 10th International Conference on* (IEEE, 2015), pp. 125–130
50. S. Billings, K. Mao, Model identification and assessment based on model predicted output. Tech. rep., Department of Automatic Control and Systems Engineering, The University of Sheffield, UK (1998)
51. E.G. Nepomuceno, S.A.M. Martins, Systems Science & Control Engineering **4**(1), 50 (2016). DOI 10.1080/21642583.2016.1163296. URL http://dx.doi.org/10.1080/21642583.2016.1163296
52. S. Chen, S. Billings, W. Luo, International Journal of Control **50**(5), 1873 (1989)
53. P. Komarek, Logistic regression for data mining and high-dimensional classification. Master's thesis, Robotics Institute - School of Computer Science, Carnegie Mellon University, USA (2004)
54. A. Senawi, H.L. Wei, S.A. Billings, Pattern Recognition (2017). Accepted
55. K.P. Bennett, O.L. Mangasarian, Optimization methods and software **1**(1), 23 (1992)
56. O.L. Mangasarian, W.N. Street, W.H. Wolberg, Operations Research **43**(4), 570 (1995)
57. M. Lichman. UCI Machine Learning Repository (2013). URL http://archive.ics.uci.edu/ml
58. WHO. Breast cancer: prevention and control. URL http://www.who.int/cancer/detection/breastcancer/en/
59. T. Wang, S.U. Guan, K.L. Man, T.O. Ting, Mathematical Problems in Engineering **2014** (2014)
60. N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Journal of Artificial Intelligence Research **16**, 321 (2002)