



ORIGINAL RESEARCH PAPER

Insurance cycles detection using neural networks

Hamza Hanbali

Department of Economics, Centre for Actuarial Studies, University of Melbourne, Melbourne, Australia
Email: hamza.hanbali@unimelb.edu.au

(Received 14 November 2024; revised 11 March 2025; accepted 01 April 2025)

Abstract

This paper utilizes neural networks (NNs) for cycle detection in the insurance industry. The efficacy of NNs is compared on simulated data to the standard methods used in the underwriting cycles literature. The results show that NN models perform well in detecting cycles even in the presence of outliers and structural breaks. The methodology is applied to a granular data set of prices per risk profile from the Brazilian insurance industry.

Keywords: Cycles; neural networks; property-liability insurance

1. Introduction

The underwriting cycle refers to the repetitive patterns of ups and downs in property and casualty insurance. Industry insiders and academics have reported cycles of profitability in various countries and lines of business (Chen et al., 1999; Cummins & Outreville, 1987; Meier & Outreville 2006; Pentikäinen & Rantala, 1982; Stewart et al., 1991; Venezian, 1985). While existing literature has mostly focused on the causes of cycles (Bruneau & Sghaier, 2015; Chen et al., 1999; Choi et al., 2002; Doherty & Kang, 1988; Fun et al., 1998; Gron, 1994; Higgins & Thistle, 2000; Lamm-Tennant & Weiss, 1997; Niehaus & Terry, 1993; Wang & Murdock 2019; Winter, 1988;), research on the consequences of the cycles is underdeveloped, but it is often argued that they could negatively impact the affordability and availability of policies during the peaks of the cycle and that they could be challenging for policyholders in terms of financial planning. Insurers' insolvencies are believed to be higher following the troughs of the cycle, and the existence of cycles was shown in Trufin et al. (2009) to have an increasing effect on insolvency risk. Furthermore, since the seminal paper of Taylor (1986), insurance cycles have gained popularity in the actuarial literature where several contributions have used insights from control theory and game theory to develop pricing models able to produce them (Emms, 2012; Malinovskii, 2010, 2013a, b). Indeed, as noted by Feldblum (2001), cycles may not be just an inefficiency of the insurance market, and to the extent that they exist and that they are nurtured by competitive forces, pricing actuaries should incorporate them into their models.

The present study is motivated by the fact that the tools usually applied to detect cyclicity were questioned, raising doubts on the very existence of cycles in the insurance industry. The typical approach in existing studies consists in fitting a second-order autoregressive process (AR(2)), and verifying whether the characteristic equation of the estimated model has complex roots. Boyer et al. (2012) and Boyer & Owadally (2015) argue that researchers have studied spurious cycles in what could simply be random walks. Specifically, Boyer et al. (2012) noted that the probability of observing cycles given random distributions of the parameters of an AR(2) process is

two-thirds, and they observe that most papers that studied multiple time series of insurance profitability report cycles in about that proportion. Boyer et al. (2012) conclude that “*cycles in property and casualty insurance industry exist only because we expect them to exist.*”

The present paper addresses this concern in a simulation study by comparing the performance of AR(p) processes, the g -test for the significance of spectral density, and neural networks (NNs) trained on data simulated from AR(p) processes. The data are simulated either from AR(p) with $p = 0, 1$ or 2 , or from noisy sinusoids, and with different cycle periods, distributions of the error terms, as well as with or without outliers or structural breaks in the mean. This means that the simulation study evaluates the ability of each methodology (g -test, AR processes, and NNs) to differentiate between AR(2) and noisy sinusoids, versus AR(1) and AR(0) series. The focus on AR processes in the detection of cycles is motivated by the fact that, besides the few applications of the g -test, fitting an AR(2) is the most common approach in the literature. That is, in line with the literature on the underwriting cycle, the definition of cyclicity adopted in this paper is whether a time series is better modeled with either a cyclical AR(2) or a noisy sinusoid, rather than an AR(1) or AR(0) process. However, it is worth mentioning that other processes can generate cyclicity, such as AR(p) processes with $p \geq 3$, but they are not included in this study.

The results of the simulation study show that AR(p) processes and the g -test can adequately distinguish between cyclical AR(2) series and non-cyclical AR(1)/AR(0) ones, but their performance deteriorates in the presence of outliers or structural breaks in the mean, which is due to over-rejection of cyclicity. On the other hand, NN models prove to be significantly better in the classification and are more robust to the presence of outliers and structural breaks in the mean. In other words, NNs trained on a mix of AR(1)/AR(0) and cyclical AR(2) are better at identifying the cyclical AR(2) series than a direct estimation of an AR(p) model.

At the core of this analysis is a novel data set from the Brazilian car insurance market. The data set comprises bi-annual time series over 14 years of average market prices for 410 risk profiles segregated by region, sex, and age group. This is a noteworthy novelty, as in most countries, historical data on insurance prices are not publicly available, and insurance companies usually do not report price and quantity separately. This has forced existing studies to focus on accounting data, but little is known about the patterns of prices, which are at the heart of underwriting cycle explanations. The granularity of this data set allows to build stronger and more efficient NNs, as it offers a wider range of parameters for cyclical processes.

The NNs trained on simulated data classify prices of almost all risk profiles as cyclical. The classification of the series using the NN models is in many cases consistent with that of AR processes and the g -test, but the two traditional methods suggest lower proportions of cyclical series. This evidence of cyclicity provided in the present paper is new. More importantly, the prevalence of cyclicity in price indicates that insurance cycles are not only a phenomenon that can affect insurers, but that it can also be experienced by consumers. Thus, taking the results presented here at face value prompts the question of whether price cycles are harmful to consumers; a topic that will be explored in future research.

Two key considerations must be kept in mind when interpreting the results of the empirical part of the paper. Firstly, the NN models employed are trained to distinguish between cyclical AR(2) and cosine-based series from AR(1)/AR(0), but they do not identify all forms of cyclicity. The simulation study does not explore how NNs would classify series with complex, non-cyclical temporal structures, such as non-cyclical AR(3) models or heteroscedastic series. Nevertheless, preliminary tests generally indicate the absence of heteroscedasticity and i.i.d. residuals from AR(1) or AR(2) models. Furthermore, AR(3) models were estimated, and the results suggest that they only outperformed AR(1) and AR(2) models in a few cases and that they support cyclicity. Secondly, the 14-year data set encompasses only about 3 cycles, making it challenging to identify cyclicity from an econometric perspective. However, the simulation study shows that NNs can detect cyclical AR(2) patterns and noisy sinusoids with the same number of observations as the empirical study, suggesting that the number of observations is not a major concern.

The remainder of the paper is organized as follows. Section 2 provides some background on the Brazilian insurance market and describes the data. Section 3 discusses the methodologies for cycle detection, namely, the *g*-test, autoregressive processes, and NNs. Section 4 presents the design of the simulation study and its results highlighting the performance of NNs over the competing methods. Section 5 contains the empirical results from the analysis of the Brazilian data on price. Section 6 closes the paper.

2. Data

This section discusses the data used in this paper, with some background on the Brazilian insurance industry, and a description of the data set of premiums.

Insurance in Brazil has witnessed a fast evolution over the past years. The total assets of insurance companies represented 9.5% of the country's GDP in 2010. According to the IMF (2012) report, the Brazilian insurance market is the largest among Latin American countries, with 50% of the region's total collected premiums in 2010. Further, the total collected premiums for all insurance lines combined was equivalent to 1.09% of the OECD's total in 2018, which makes it comparable to Australia (1.32%), Spain (1.48%), the Netherlands (1.64%) or Canada (1.64%) in terms of collected premiums (<https://stats.oecd.org/>).

Insurance in Brazil is regulated by the Superintendency of Private Insurance (Superintendência de Seguros Privados – SUSEP), which is the executive arm of the National Council for Private Insurance, and has jurisdiction over all states (Frazão 2020; IMF, 2012). Motor insurers operating in Brazil are free in setting their premiums (Frazão, 2020). Moreover, entering the market is relatively easy, and insurers are subject to the standard requirements in terms of technical provisions and minimum solvency capital. Thus, the Brazilian motor insurance industry resembles that of most developed economies.

All insurance companies operating in Brazil must comply with SUSEP's reporting and disclosure rules by submitting statistics conforming to the SUSEP Periodic Information Form. For motor insurance, insurers report their prices, exposure, and insured sums for different risk profiles. Unlike in most countries, the data on premiums per risk profile are freely accessible online, on the AUTOSEG database (see the AUTOSEG database web page). The website also has an active forum where users can request clarification on the data, with high participation and fast responses from the forum's administrators (see SUSEP's forum web page).

The AUTOSEG database contains time series for different risk profiles for supplementary motor insurance covering accident, fire, and theft. The variables used in the analysis are the total exposure, the market average premiums, and market average insured sums. The market averages are weighted sums of the relevant quantity across all insurance companies, where the weights are determined from the proportion of policies for each company. The total exposure is a proxy for the number of policies over a given semester and accounts for the fact that some policies may be active only over a fraction of a given time period. The market average premiums represent the average amount paid by policyholders. Whereas this amount varies by region, sex, and age, it also depends on other time-varying factors such as inflation, the value of the car, and the limit applied to the policy. Those time-varying factors are captured in the market average insured sums, which is the maximum amount payable to the policyholder, and can be understood as a monetary amount representing the financial exposure of the insurer. In particular, standardizing the market average premiums by the market average insured sums gives the insurance price for each insured Brazilian Real.

The data set used in this study consists of time series for 410 different risk profiles, which are categorized by region (41 regions), sex (male or female), and age group (18–25, 26–35, 36–45, 46–55 and 55+). The observations are available at a biannual frequency, from July–December 2006 to January–June 2020, which gives a total of 28 observations for each of the 410 risk profiles. This means that each data point on premiums corresponds to the average premium paid by policyholders having a specific risk profile in a given semester.

The time series analyzed here are the ratio between the market average premium and the market average insured sum for each risk profile and each observation time, on a logarithmic scale, which gives the logarithm of the price for each insured Brazilian Real. In particular, as mentioned above, the standardization by the insured sum leads to prices per unit of risk, and hence, allows for a better comparison across the different risk profiles and rules out the effect of other time-varying factors such as inflation. Note that while “price” and “premium” may be used interchangeably in the insurance industry, here the term “premium” refers to the original commercial premium time series, and the term “price” refers to the premiums standardized by the insured sum.

From the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test of stationarity, time series for some risk profiles have a p -value less than 0.05, which is mostly due to trend non-stationarity. Following Brockwell & Davis (2002), these series are de-trended by fitting individual regressions on time and subtracting the trend line from the original series. As a result, the KPSS test leads to p -values greater than 0.05 for all series. An overall p -value for all risk profiles can be determined by combining all null hypotheses of stationarity using Hartung (1999)’s test. The overall p -value with the conventional tuning parameter 0.2 equals 0.09374, which suggests that prices are stationary. The ARCH test’s p -values on the de-trended series are above 0.05 for 88.78% of the risk profiles. The ARCH test on the residuals of AR(1) and AR(2) models lead to p -values above 0.05 for 93.9% and 96.1% of the risk profiles, respectively, thus rejecting an ARCH effect and suggesting homoscedasticity for most risk profiles. The Ljung-Box test on the de-trended series leads to p -values below 0.05 for 98.5% of risk profiles, as well as for 61.65% of risk profiles when the test is performed on the residuals of an AR(1), and for 0.49% of the risk profiles when the test is performed on the residuals of an AR(2) model. In particular, for the AR(2) model, the Ljung-Box test fails to reject the null hypothesis of i.i.d. residuals for all risk profiles except two. These results already suggest that for most risk profiles, it is appropriate to consider AR processes of order 2.

Finally, all series were normalized by removing the means and dividing by the standard deviation. This procedure is recommended when working with NNs.

Fig. 1 displays (a) the original market average premiums, (b) the premiums divided by the insured sums (i.e. the price) on a logarithmic scale, (c) the price from (b) de-trended, and (d) the price from (c) normalized. All figures hint at the presence of cycles in the prices for the Brazilian motor insurance market. However, for some risk profiles, the data appears to be rather noisy. For instance, risk profiles from Rio Grande do Sul experienced a sharp drop in prices in the second semester of 2008, and males aged 18-25 from Rio de Janeiro and São Paulo experienced a milder drop in the second semester of 2016. Therefore, a systematic and automated method for cycle detection is required to ascertain the existence of cycles in insurance prices.

3. Cycle detection methodologies

In this paper, a time series is considered cyclical if it is better modeled by an AR(2) process with complex roots or by a noisy sinusoid, rather than by an AR(0) or AR(1) process. This definition follows the standard approach in the underwriting cycle literature and is consistent with the methodologies evaluated in this study. Non-standard approaches such as AR processes with orders higher than 2, non-linear time series, or wavelet analysis, are not considered. Furthermore, heteroscedasticity is not accounted for, which is not a major omission for the paper’s application given that the results of the ARCH test do not provide strong support for the presence of heteroscedasticity in the time series.

This section outlines three methodologies used to detect cycles in times series. Subsection 3.1 describes the g -test. Subsection 3.2 describes the methodology based on autoregressive models. Subsection 3.3 describes different NN models. The focus on the g -test and the AR models is motivated by the fact that they are the main approaches for cycle detection in the underwriting cycle literature, especially AR processes of orders up to 2.

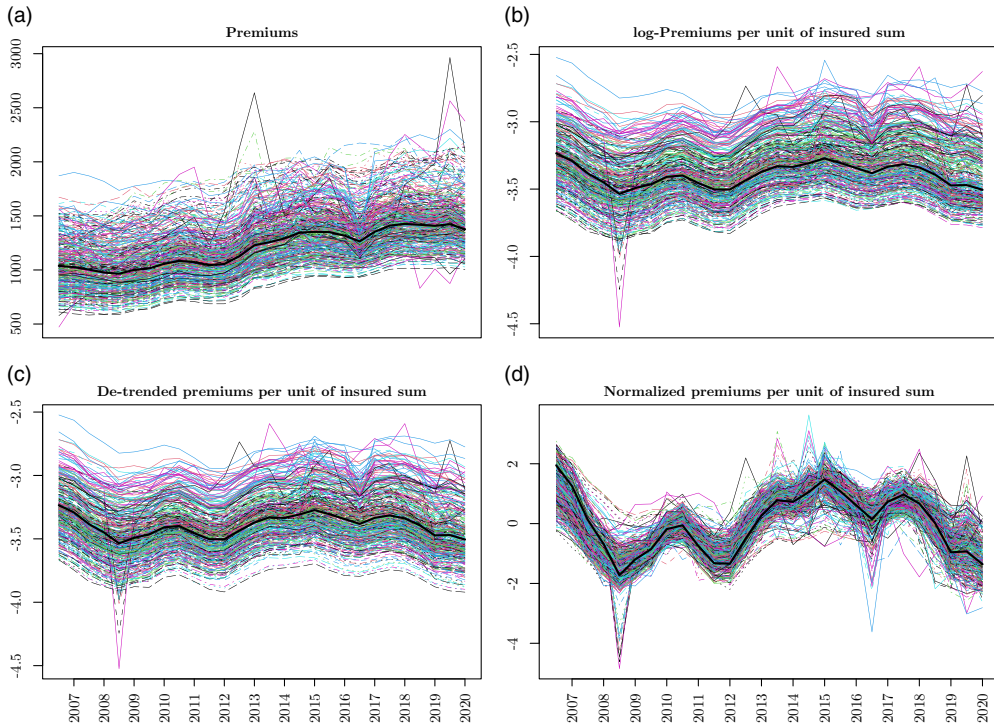


Figure 1. Premiums and prices per risk profile over time. *Notes:* (a) Top-left panel: original average premiums for 410 risk profiles from the AUTOSEG data set. (b) Top-right panel: corresponding premiums per unit of insured sums on a logarithmic scale. (c) Bottom-left panel: corresponding premiums per unit of insured sums (i.e. price) on a logarithmic scale, where prices for some risk profiles were de-trended, after which all time series passed the KPSS test of stationarity. (d) Bottom-right panel: corresponding prices after normalizing with the means and standard deviations.

3.1 Significance test for spectral peaks

One method to detect cyclicity in time series consists in visualizing their spectral densities; see Grace & Hotchkiss (1995) and Venezian & Leng (2006) for applications in the context of the underwriting cycle. A spectral density function is a projection of the auto-correlation function from the time domain onto the frequency domain and is derived from a representation of time series using combinations of sinusoidal functions (Brockwell & Davis, 2002). It is possible to detect cyclicity in time series if the spectral density has a peak at some given frequency.

When performing spectral analysis, a problem that may arise is that a spectral density function may have multiple peaks, and in such cases, it may be hard to draw sensible conclusions. Furthermore, carrying out a graphical analysis for each time series in the present analysis would be too cumbersome and not necessarily accurate. One way to automate the process, and to address the issue of spurious cycle detection, is to test the significance of spectral peaks (Lazar & Denuit, 2012; Venezian, 2006). The *g*-test proposed in Fisher (1929) is one of the earliest methods applied for cycle detection, and it is still in use; see e.g. Lazar & Denuit (2012) for a review and application to the underwriting cycle.

The point estimate of the *g*-test statistic is given by:

$$\bar{g} = \frac{\max_k I(\omega_k)}{\sum_{k=1}^n I(\omega_k)},$$

with $n = \lfloor \frac{T-1}{2} \rfloor$, and T is the number of observations, where $I(\omega_k)$ is the periodogram at the discrete Fourier frequencies $\omega_k = \frac{2\pi k}{T}$, for $k = 0, 1, \dots, \lfloor \frac{T}{2} \rfloor$, such that:

$$I(\omega) = \frac{1}{T} \left| \sum_{t=1}^T y_t \exp(-i\omega t) \right|^2, \quad \omega \in [0, \pi],$$

and y_t is the time series under interest. Under the null hypothesis, y_t is a Gaussian white noise process. Large values of \bar{g} indicate that the highest peak of the spectral density, which is estimated by the periodogram, is significant, and hence lead to the rejection of the null hypothesis. Under the null hypothesis, the exact distribution of the g -test statistic is given by:

$$\mathbb{P}[g > \bar{g}] = \sum_{k=1}^a (-1)^{k-1} \frac{n!}{k! (n-k)!} (1 - k\bar{g})^{n-1},$$

where a is the maximum integer less than $\frac{1}{\bar{g}}$.

3.2 Autoregressive models

Detecting cyclical behavior using second-order autoregressive processes, or AR(2), is the most common way in studies of the underwriting cycle (Cummins & Outreille, 1987; Venezian, 1985). It is also the methodology criticized by Boyer et al. (2012). In general, autoregressive processes exhibit cyclical behavior, provided the characteristic equation has complex roots. It follows that an autoregressive process of order 1 or less is not cyclical.

An AR(2) process has the following representation:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \epsilon_t, \quad (3.1)$$

where ϵ_t is a white noise. The root of the characteristic equation is $\phi_1^2 + 4\phi_2$, and hence, the process y_t is cyclical if $\phi_1^2 + 4\phi_2 < 0$. In this case, the period of the cycle is $\frac{2\pi}{f}$, where the expression of the frequency f is $\arccos\left(\frac{0.5\phi_1}{\sqrt{-\phi_2}}\right)$; see Cummins & Outreille (1987).

Based on the properties of the process in (3.1), the classification of the time series into cyclical and non-cyclical premiums is achieved by fitting model (3.1) to the time series of interest, as well as the same model with $\phi_2 = 0$ or $\phi_2 = \phi_1 = 0$. The optimal model is chosen according to the value of the corrected Akaike Information Criteria (AICc):

$$\text{AICc} = T \log \left(\frac{1}{T} \sum \hat{\epsilon}_t^2 \right) + 2(k+1) + \frac{2(k+1)(k+2)}{T-k-2},$$

where T is the number of observations, and k is the number of free parameters. For a given time series and for AR(p) with $p \leq 2$, if the model with the lowest AICc is not an AR(2) model, then the series is classified as non-cyclical. Otherwise, if the optimal model is an AR(2), the series is classified as cyclical if $\phi_1^2 + 4\phi_2 < 0$, and as non-cyclical otherwise.

3.3 Neural networks

NNs are computational graphs that can extract mappings between inputs and targets. The extraction of mappings, or learning, occurs when the model is exposed to a sufficiently large amount of input-target pairs. NN models typically operate in layers, where each layer transforms the information distilled by the previous one. Each layer is parameterized by a set of weights, which are estimated during the learning phase. The estimation of the parameters is generally performed through a feedback mechanism, most commonly a gradient-based optimization with backpropagation, to determine the weights that minimize the error between the predictions and the actual

targets (Aggrawal, 2018; Chollet & Allaire, 2018; Rumelhart et al., 1986). NNs have proven efficient in many actuarial applications, including forecasting mortality rates and modeling claims in non-life insurance for pricing and reserving (Gabrielli, 2020; Hainaut, 2018; Li, 2022; Lindholm et al., 2022; Shi and Shi, 2022). In the present context, the goal is to classify the series as either cyclical or non-cyclical.

NN models need to be trained using data for which both input and output are known before they are applied for predictions, i.e. in situations where the output is unknown. The data for which both input and output are known are used to update the parameters of the model and to tune its hyperparameters (i.e. the set of defining features of the network, including its structure and the way it learns). The next section describes how several data sets are simulated. The simulated data are used to train the NN models, as well as to compare the performance of the three cycle detection methodologies (the *g*-test, autoregressive processes, and NN models).

Four NN architectures are considered in this study. The architectures are determined using the simulated data described in the next subsection. Hyperparameter tuning is based on standard best-practice methods by monitoring the accuracy of the validation sets. The first model is a fully connected NN (FNN) and consists of fully connected layers only. It has four hidden layers, with number of nodes 512, 512, 64, and 8, and dropout rates of 0.4 at each layer. The second model is a recurrent NN (RNN). It has two identical bidirectional Gated Recurrent Unit (GRU) layers with 256 units each, a dropout rate of 0.1, and a recurrent dropout rate of 0.5. The output of the recurrent base of the network is flattened and then processed through a fully connected layer with 256 units, and a dropout rate of 0.1. The third model is a one-dimensional convolutional NN (CNN). It has two convolution layers with 512 and 256 filters of size 25, respectively. A stride of 2 is used with zero padding. The output of the convolution base is flattened into a first fully connected layer with 128 units and a dropout rate of 0.1, and then a second fully connected layer with 64 units and a dropout rate of 0.2. The fourth model is a hybrid convolutional-recurrent architecture. It starts with a one-dimensional convolution layer with 512 filters with size 20, and a stride of 2 with zero-padding. The convolution layer is followed by a bidirectional GRU layer with 128 units, a dropout rate of 0.1, and a recurrent dropout rate of 0.5. The recurrent layer of the hybrid NN is connected to the final single-node output layer through a fully connected layer with 32 units and a dropout rate of 0.1.

For all architectures, the target function for calibration is binary cross-entropy, which is standard for classification tasks. The output layer consists of a single neuron with a sigmoid activation function, producing a probability score for each series. A threshold of 0.5 is used to classify a series as cyclical or non-cyclical. The input to the NN consists of the full-time series, represented as a vector of 28 observations corresponding to the length of each sample in the data set. Models are trained using the Adam optimizer with early stopping to prevent overfitting, and the number of epochs is set to 30 based on validation performance.

It is worth noting that given the number of parameters in NN models, overfitting is a potential concern. However, several measures are implemented to mitigate this risk. First, dropout regularization, as described above, prevents the networks from relying too heavily on specific neurons. Second, the size of the training set is sufficiently large relative to the number of parameters, with thousands of time series providing diverse patterns for robust learning. Last, the performance of the models is evaluated on completely unseen test data, confirming that improvements in cycle detection are not artifacts of overfitting but rather genuine enhancements in identifying cyclical patterns.

4. Simulation study

The goal of this subsection is to compare the performance of the different methodologies on simulated data. It is structured in three parts. Subsection 4.1 describes the design of the simulation study. Subsection 4.2 provides details on the procedure for simulating different data sets. Subsection 4.3 reports the results of the simulation study. Recall that the simulated data sets serve

two purposes. The first purpose of the simulated data sets is to compare the efficacy of all methodologies (g-test, AR processes, and the four NN models) in cycle detection. The results of this analysis are reported in Subsection 4.3. The second purpose of the simulated data sets is to train the NN models for the empirical application of the paper. Specifically, the NN's models *learn* to detect cyclicity in the simulated data before they are applied to the Brazilian data. These latter results are reported in Section 5.

4.1 Design of the simulation

Standard best-practice methods to evaluate the efficacy of NN models prescribe using three different data sets. The training and validation sets, which are used to train and fine-tune the architectures, and the test data set, which is used to compare the performance of the models. In order to ensure that the performance of the NNs on the test data is not due to over-fitting or to the similarity of that data set with the training data, the three data sets should be disjoint.

The simulation study in this paper uses five sets of data generating processes \mathcal{G}_i , $i = 1, \dots, 5$, to generate five data sets \mathcal{S}_i . Each data set \mathcal{S}_i contains a mix of cyclical AR(2) and non-cyclical AR(1) and AR(0). The series across the different sets \mathcal{S}_i are grouped by the parameters of the AR(p) processes. Namely, each set contains cyclical time series with specific range of values of cycle periods, AR(1) time series with specific range of values of the parameter ϕ_1 , and AR(0) time-series with specific range of values of standard deviations. In particular, \mathcal{S}_1 is associated with “low” values, \mathcal{S}_2 is associated with “medium-low” values, \mathcal{S}_3 is associated with “medium-high” values and \mathcal{S}_4 is associated with “high” values. All intermediary values, as well as “very low” and “very high” values are included in \mathcal{S}_5 . This procedure ensures that the overlap between the sets \mathcal{S}_1 , \mathcal{S}_2 , \mathcal{S}_3 and \mathcal{S}_4 is significantly reduced.

The first step is to determine the hyperparameters of the NN models (e.g. the number and size of layers). Note that fixing the hyperparameters and estimating the weights of the NN models are two different tasks, and the latter task is referred to here as “training.” In this first step, where the hyperparameters are determined, \mathcal{S}_1 and \mathcal{S}_4 are used as training data, and \mathcal{S}_3 is used as validation data to tune the hyperparameters of the NNs. The second step is to study the performance of all methodologies (g-test, autoregressive models, and NNs) in detecting different types of cyclicity. In particular, using the NN architectures obtained from Step 1, the simulation study is structured as follows:

- (a) train the NN models on \mathcal{S}_2 , \mathcal{S}_3 and \mathcal{S}_4 , and test all methods on \mathcal{S}_1 (low).
- (b) train the NN models on \mathcal{S}_1 , \mathcal{S}_3 and \mathcal{S}_4 and test all methods on \mathcal{S}_2 (medium-low).
- (c) train the NN models on \mathcal{S}_1 , \mathcal{S}_2 and \mathcal{S}_4 and test all methods on \mathcal{S}_3 (medium-high).
- (d) train the NN models on \mathcal{S}_1 , \mathcal{S}_2 and \mathcal{S}_3 and test all methods on \mathcal{S}_4 (high).
- (e) train the NN models on \mathcal{S}_1 , \mathcal{S}_2 , \mathcal{S}_3 and \mathcal{S}_4 and test all methods on \mathcal{S}_5 (intermediary and extreme).

Note again that the architectures of the NN models are determined using \mathcal{S}_1 , \mathcal{S}_3 and \mathcal{S}_4 , but the parameters of each architecture are different for each test and depend on the training data used. Note also that in each test, the data used in the comparison of the methodologies are different from those on which the NN models are trained.

4.2 Simulated data

This subsection explains how the simulated data are obtained. First, a decomposition procedure is applied to the AUTOSEG data to obtain a larger sample of parameters of both cyclical and non-cyclical AR(p) models. This is because the NN models will be applied to detect cyclicity in the Brazilian insurance industry, and a resemblance between the simulated and the available data will increase the reliability of the NNs. Second, the sets of estimated parameters are augmented by

fitting a multivariate model to the estimated ones and simulating a large number of new parameters from that multivariate distribution. Third, the data-generating processes are assigned to the sets \mathcal{G}_i using a splitting approach that attempts to avoid overlap between the sets. Fourth, within each of these five sets, the cyclical data-generating processes based on AR(2) models are supplemented with cosine-based models. Fifth, the data are simulated and normalized using the means and standard deviations. The remainder of this subsection provides details on each of these five steps.

4.2.1 Step 1 – STL decomposition

In order to increase the available data, each of the AUTOSEG time series of prices is filtered using Cleveland et al. (1990) seasonal-trend (STL) procedure to separate the trend and the remainder components. Note that since the data is bi-annual, the seasonal component is insignificant, and hence, the outcome of the STL decomposition for each series of the AUTOSEG is one cyclical series and a remainder non-cyclical series. The autoregressive process in (3.1), as well as the nested models with $\phi_2 = 0$ or $\phi_2 = \phi_1 = 0$, are estimated for each series, and the optimal model is selected based on the AICc. All of the 410 trend components series except two are better modeled with AR(2) cyclical processes, whereas all of the 410 remainder components series are better modeled with either AR(1) or AR(0) processes. This results in 818 sets of parameters $(\hat{\phi}_0, \hat{\phi}_1, \hat{\phi}_2, \hat{\sigma})$ with no duplicates, where $\hat{\sigma}$ is the standard deviation of the residuals, and $\hat{\phi}_2 < 0$ and $\hat{\phi}_1 + 4\hat{\phi}_2 < 0$ for all AR(2) processes. Since all series are normalized, ϕ_0 is set to 0.

4.2.2 Step 2 – Multivariate distribution for simulating new parameters

The vector $(\log(\hat{\phi}_1), \log(-\hat{\phi}_2), \sqrt{\hat{\sigma}})$ of estimated parameters from the AR(2) processes and its counterpart $(\log(\hat{\phi}_1), \sqrt{\hat{\sigma}})$ from the AR(1) processes are modeled using a variance gamma for the marginal distributions and a vine copula for the dependence structure. These models are used to generate 10,000 new sets of AR(2) parameters $(\tilde{\phi}_1, \tilde{\phi}_2, \tilde{\sigma})$ satisfying the condition of cyclicity, and another 10,000 of AR(1) parameters. For the AR(0) processes, 10,000 parameters $\tilde{\sigma}$ are simulated assuming that the estimated $\sqrt{\hat{\sigma}}$ follow a variance gamma distribution. The choice of the marginal distributions and log transformations was based on a visual inspection of the Q-Q plots, and the selection of the copulas is based on a comparison of the AIC.

4.2.3 Step 3 – Assignment of the parameters

In order to ensure that the training, validation, and test data are disjoint, the parameters $(\tilde{\phi}_1, \tilde{\phi}_2, \tilde{\sigma})$ must be split before the simulations. The typical approach for splitting the data is to randomly assign proportions for each set. In the present context, this approach could assign similar parameters across the three data sets, which would violate the requirement of disjoint sets.

The approach followed here for splitting the data-generating processes attempts to ensure that the sets are disjoint and allows to evaluate the performance of the models in detecting different types of cyclicity (e.g. cycles with low or high periods). For the AR(2) processes, this is achieved by assigning the parameters based on the value of their associated cycle period. Those values are first split into several bins based on their quantile values. The bins are then allocated to the sets \mathcal{G}_i for $i = 1, \dots, 5$. A similar procedure is performed for the sets of parameters of the non-cyclical processes, where the reference quantities used for the partitioning are the values of $\tilde{\phi}_1$ for the AR(1), and $\tilde{\sigma}$ for the AR(0).

Fig. 2 shows how the partitioning is performed for the AR(2) processes. On the left panel, the figure displays the estimated cycle periods $\frac{2\pi}{f}$ (gray circles) and the assignment of the simulated

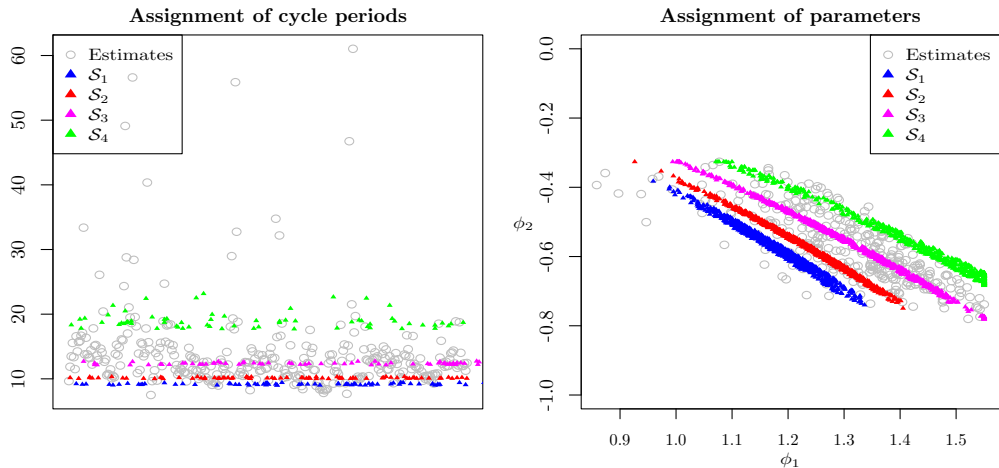


Figure 2. Assignment of parameters for the simulated data. *Notes:* This figure displays the split of the parameters into \mathcal{G}_i for $i = 1, \dots, 4$. On the left panel, the figure displays the estimated cycle periods $\frac{2\pi}{f}$ (gray circles) of the cyclical trend components, and the assignment of the simulated cycle periods $\frac{2\pi}{f}$ into \mathcal{G}_1 (blue triangles), \mathcal{G}_2 (red triangles), \mathcal{G}_3 (magenta triangles) and \mathcal{G}_4 (green triangles). On the right panel, the figure displays the corresponding estimated parameters $(\hat{\phi}_1, \hat{\phi}_2)$ (gray circles) and the assignment of the simulated parameters $(\tilde{\phi}_1, \tilde{\phi}_2)$ into \mathcal{G}_1 (blue triangles), \mathcal{G}_2 (red triangles), \mathcal{G}_3 (magenta triangles) and \mathcal{G}_4 (green triangles).

cycle periods $\frac{2\pi}{f}$ into \mathcal{G}_1 (blue triangles), \mathcal{G}_2 (red triangles), \mathcal{G}_3 (magenta triangles) and \mathcal{G}_4 (green triangles). On the right panel, the figure displays the corresponding estimated parameters $(\hat{\phi}_1, \hat{\phi}_2)$ (gray circles) and the assignment of the simulated parameters \mathcal{G}_1 (blue triangles), \mathcal{G}_2 (red triangles), \mathcal{G}_3 (magenta triangles) and \mathcal{G}_4 (green triangles). The figure shows that there is no overlap between the sets in terms of cycle periods. The figure also shows that the simulated parameters are consistent with those estimated from the Brazilian data sets.

4.2.4 Step 4 – Cyclical data generating processes using cosines

A cyclical data generating process given by $\beta \cos(\omega t) + \epsilon_t$ is added, where ϵ_t has a standard deviation $\tilde{\sigma}$. The standard deviation $\tilde{\sigma}$ is sampled from those of the AR(2), and β is uniformly drawn from $[0.5, 5.5]$. The parameter ω is such that the period $\frac{2\pi}{\omega}$ is consistent with $\frac{2\pi}{f}$ from the relevant set \mathcal{G}_i .

4.2.5 Step 5 – Data simulation

For each set of parameters, several paths are simulated, assuming that ϵ_t has either a normal distribution with standard deviation $\tilde{\sigma}$, or a t distribution with degree of freedom $\nu = 2.5$ and scaled by $\frac{\tilde{\sigma}}{\sqrt{5}}$. Scaling by $\frac{\tilde{\sigma}}{\sqrt{5}}$ in case of t distributed errors ensures the standard deviation of the process is equal to $\tilde{\sigma}$. For each path, the number of observations is randomly chosen between 50 and 150, but only the last 28 observations are selected. This allows us to obtain asynchronous paths. Each data set \mathcal{S}_i for $i = 1, \dots, 5$ contains 2,000 paths from the cyclical AR processes with normally distributed errors, 2,000 paths from the cyclical AR processes with t distributed errors, 2,000 paths from the cosine process, 3,000 paths from the non-cyclical AR processes (either AR(0) or AR(1)) with normally distributed errors, and 3,000 paths from the non-cyclical AR processes (either AR(0) or AR(1)) with t distributed errors. In particular, each data set contains the same number of time series from each class, namely 6,000 cyclical time series and 6,000 non-cyclical

Table 1. Accuracy, sensitivity, and specificity of all methodologies on different simulated test data

Training data: Test data:	S_2, S_3, S_4 S_1	S_1, S_3, S_4 S_2	S_1, S_2, S_4 S_3	S_1, S_2, S_3 S_4	S_1, S_2, S_3, S_4 S_5
<i>Accuracy rate:</i>					
<i>g</i> -test	80.07%	79.52%	80.23%	79.06%	80.41%
<i>g</i> -test with Bonferroni correction	56.49%	59.61%	56.92%	57.37%	58.1%
Autoregressive model	88.83%	88.08%	88.63%	88.02%	88.85%
Fully connected NN	93.52%	90.54%	94.19%	92.49%	94.05%
Recurrent NN	96.54%	87.88%	95.20%	93.23%	94.88%
Convolutional NN	95.09%	90.71%	95.13%	93.67%	95.03%
Hybrid conv-rec NN	95.71%	92.03%	95.62%	93.89%	95.45%
<i>True positive rate:</i>					
<i>g</i> -test	71.47%	89.23%	76.02%	81.27%	80.57%
<i>g</i> -test with Bonferroni correction	13.03%	25.90%	14.00%	16.12%	18.07%
Autoregressive model	83.38%	81.65%	83.07%	81.52%	82.63%
Fully connected NN	88.23%	93.57%	90.30%	89.42%	90.90%
Recurrent NN	96.43%	98.68%	97.13%	96.17%	97.18%
Convolutional NN	92.03%	97.23%	93.32%	93.6%	94.07%
Hybrid conv-rec NN	92.82%	96.72%	94.18%	93.03%	95.07%
<i>True negative rate:</i>					
<i>g</i> -test	88.67%	69.80%	84.43%	76.85%	80.25%
<i>g</i> -test with Bonferroni correction	99.95%	93.32%	99.83%	98.62%	98.13%
Autoregressive model	94.27%	94.50%	94.20%	94.52%	95.07%
Fully connected NN	98.80%	87.52%	98.08%	95.57%	97.20%
Recurrent NN	96.65%	77.07%	93.27%	90.28%	92.58%
Convolutional NN	98.15%	84.18%	96.95%	93.75%	95.98%
Hybrid conv-rec NN	98.60%	87.35%	97.05%	94.75%	95.83%

Notes: This table reports the accuracy rate, the true positive rate (sensitivity) and the true negative rate (specificity) of all methodologies on the test data S_j when the training data set is $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, for $j = 1, 2, 3, 4$, as well as the performance on the test data S_5 when the training data set is $\cup_{1 \leq i \leq 4} S_i$. The highest value of each measure and data set across all models is highlighted in bold.

ones. This means that the binary classification problem is balanced (i.e. 50% of series in each class), and hence, that the classification accuracy is an appropriate measure to evaluate the performance of the models. Each simulated path is normalized individually by subtracting its mean and dividing by its standard deviation.

4.2.6 Software use

The analysis is conducted on R using the packages `tseries` and `seasonal` for time series analyses, the packages `VarianceGamma` and `rvinecopulib` for the multivariate modeling of the parameters, and the packages `keras`, `kerasR` and `tensorflow` for fitting the NNs. The code that supports the findings of this study is available on request from the author.

4.3 Results of the simulation study

4.3.1 Results on the test data sets

Table 1 reports the accuracy rate, the true positive rate (sensitivity), and the true negative rate (specificity) of all methodologies for different levels of cyclicity. Note that when the test data set S_j for $j = 1, \dots, 4$ is evaluated, the training data set for the NN models is $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, whereas when the test data set S_5 is evaluated, the training data set is $\cup_{1 \leq i \leq 4} S_i$. The results for the *g*-test are reported with and without Bonferroni correction for multiple-comparison, where the threshold for the *p*-value is set to the conventional significance level of 0.05.

Looking first at the accuracy rate, overall, the four NN models outperform both the *g*-test and autoregressive processes in terms of cycle detection on all data sets. Furthermore, the hybrid convolutional-recurrent model outperforms the other NN models with higher accuracy on all

Table 2. Estimated weights from ensembling all models

Training data: Test data:	S_2, S_3, S_4 S_1	S_1, S_3, S_4 S_2	S_1, S_2, S_4 S_3	S_1, S_2, S_3 S_4	S_1, S_2, S_3, S_4 S_5
<i>g</i> -test	0%	0%	0%	0%	0%
Autoregressive model	0%	0%	0%	0%	0%
Fully connected NN	0.87%	0%	0%	0%	0%
Recurrent NN	0%	0%	0%	19.15%	0%
Convolutional NN	82.05%	74.32%	97.80%	56.15%	99.66%
Hybrid conv-rec NN	17.08%	25.68%	2.20%	24.70%	0.34%
Accuracy rate	95.37%	91.02%	95.17%	94.33%	95.04%

Notes: This table reports the estimated weights on each method (*g*-test, autoregressive model, and NN's) obtained by minimizing the sum of squared differences between the predictions from the ensemble and the targets from the relevant training set $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, for $j = 1, 2, 3, 4$, and $\cup_{1 \leq i \leq 4} S_i$.

data sets except for S_1 (i.e. low cycle period). The accuracy rates of the top-performing model vary across the data sets, where the lowest value is 92.03% on S_2 (i.e. medium-low cycle period), and the highest value is 96.54% on S_1 (i.e. low cycle period).

Regarding the *g*-test and autoregressive processes, two observations are noteworthy. The first observation is that, whereas the *g*-test has the lowest accuracy rates, those values tend to decrease significantly when the Bonferroni correction is applied. This result suggests that Bonferroni correction is not necessarily useful in the present context. The second observation is that despite the criticism expressed by Boyer et al. (2012), the methodology based on AR processes performs satisfactorily with an average accuracy of about 88.5% across the five data sets. Nevertheless, as it is shown later in this section, the performance of the AR(*p*) model deteriorates significantly when outliers and structural breaks are present in the data.

The true positive rate is highest for recurrent NN model, where its performance is closely matched by the convolutional and hybrid networks. That performance is also associated with the lowest true negative rate among the NN models. On the other hand, the true negative rate is highest for the *g*-test with Bonferroni correction, which is close to 100%, but it comes at the cost of too low true positive rates. This is because the *g*-test with Bonferroni correction classifies most series as non-cyclical, which leads to a high specificity but a low sensitivity.

The results from Table 1 based on simulated data highlight the high performance of NNs in distinguishing between cosines with noise or AR(2) from AR(1)/AR(0) process, which substantially exceeds that of the *g*-test and autoregressive models. Furthermore, among the four NN models, the hybrid convolutional-recurrent architecture has a higher efficacy in cycle detection, even though the performance measures of all four architectures are close. Interestingly, the NN models perform better than AR(*p*) models despite the data being generated from AR(*p*) and cosine-based models.

4.3.2 Ensembling

An additional test is performed, consisting of fitting a weighted average of the predictions from the *g*-test, autoregressive processes, and the four NN models. This ensembling is performed for each of the five data sets $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, for $j = 1, 2, 3, 4$ and $\cup_{1 \leq i \leq 4} S_i$. The goal is to verify whether combining the different methodologies could help to further improve the detection of cycles. The estimated weights are reported in Table 2, where the values are estimated by minimizing the sum of squared differences between the predictions from the ensemble and the targets from the relevant training set. Note that the ensembling performed using the binary cross-entropy leads to similar conclusions.

The first observation from Table 2 is that for all data sets, the *g*-test and the autoregressive model are assigned weights of 0%. This suggests that not only are the NN models able to detect

cyclical with high accuracy, but that additional input from the g -test and autoregressive processes is not necessary on the training data. The second observation is that, among the NN models, the convolutional NN is assigned the highest weight on all training data sets. The hybrid model comes second, and the recurrent NN comes third, where it is assigned a positive weight for one data set only. The FNN is not used at all.

It is worth mentioning that these weights are determined using the training data, and hence, this result does not necessarily mean that the convolutional NN is the best model on the unseen test data. Indeed, as discussed in the previous subsection, Table 1 shows that the hybrid convolutional-recurrent NN outperforms the other models on 4 out of the 5 test data sets, whereas the recurrent NN outperforms the other models on the remaining data set. In Table 2, the accuracy rates on the test data show that the ensembles outperform the convolutional NN, but the accuracy rates are still lower compared to those of the hybrid convolutional-recurrent NN.

4.3.3 Performance in the presence of outliers and structural breaks

In order to provide deeper understanding of the differences between the methodologies in terms of cycle detection, this subsection investigates the cycle detection power of all methodologies when the time series contain either outliers or structural breaks in the mean. Understanding the cycle detection ability of all models in such cases is important, as Fig. 1 suggests that some time series exhibit those features.

Let y_t be a simulated time series from any of the data sets S_i . Recall that the series were standardized, such that their mean is 0 and their standard deviation is 1. The corresponding time series $y_t^{\text{outlier}}(s, \delta)$ is the transformation of y_t with an outlier at a unique time $s \in \{1, \dots, T\}$ and size δ , such that:

$$y_t^{\text{outlier}}(s, \delta) = y_t + \delta \times P \times \mathbb{I}_{[t=s]}.$$

where P is equal to either -1 or 1 each with probability 0.5 , and $\mathbb{I}_{[t=s]}$ is equal to 1 at time $t = s$. The value of s is sampled from $\{1, \dots, T\}$ with equal probability. Analogously, $y_t^{\text{break}}(s, \delta)$ is the transformation of y_t with a structural break in the mean that occurs at a unique time $s \in \{3, \dots, T-2\}$ and size δ , such that:

$$y_t^{\text{break}}(s, \delta) = y_t + \delta \times P \times \mathbb{I}_{[t \geq s]},$$

where P is again equal to either -1 or 1 each with probability 0.5 , and $\mathbb{I}_{[t \geq s]}$ is equal to 1 at times $t \geq s$. The value of s for structural breaks is sampled from $\{3, \dots, T-2\}$ with equal probability. Note that s is not sampled from $\{1, 2, T-1, T\}$ to avoid redundancy. For instance, for $s = 1$, there are no structural breaks, and $s = 2$ leads to series with an outlier at time 1 .

To conclude, each time series y_t^{outlier} has a single outlier. Half of the time series y_t^{outlier} has positive outliers, and the other half has negative ones. Among the y_t^{outlier} 's, $\frac{1}{T}$ of the time series has an outlier at the s -th observation for $s \in \{1, \dots, T\}$. Analogously, each time series y_t^{break} has a single structural break. Half of the time series y_t^{break} has positive breaks, and the other half has negative ones. Among the y_t^{break} 's, $\frac{1}{T-4}$ of the time series has a structural break from time $s \in \{3, \dots, T-2\}$. The size of the outlier or the break is equal to δ , where $\delta = 0$ (i.e. no outliers or structural breaks) or $\delta \in \{1, \dots, 5\}$. Fig. 3 displays an example of the transformation for a non-cyclical and a cyclical time series from the data set S_1 .

When the NN models are trained using data containing outliers or structural breaks, the training sets include series with $\delta \in \{0, 1, 2, 3\}$ but not with $\delta \in \{4, 5\}$. However, the testing data sets include series with all values of δ , i.e. $\delta \in \{0, \dots, 5\}$. This allows us to determine how the NN models generalize their detection of cycles in the presence of outliers and breaks on which they were not trained, and in particular to $\delta = 4$ or 5 . In the remainder of this section, a data set S_i to which outliers and structural breaks were added is denoted by S_i^* . The set S_i^* includes a balanced

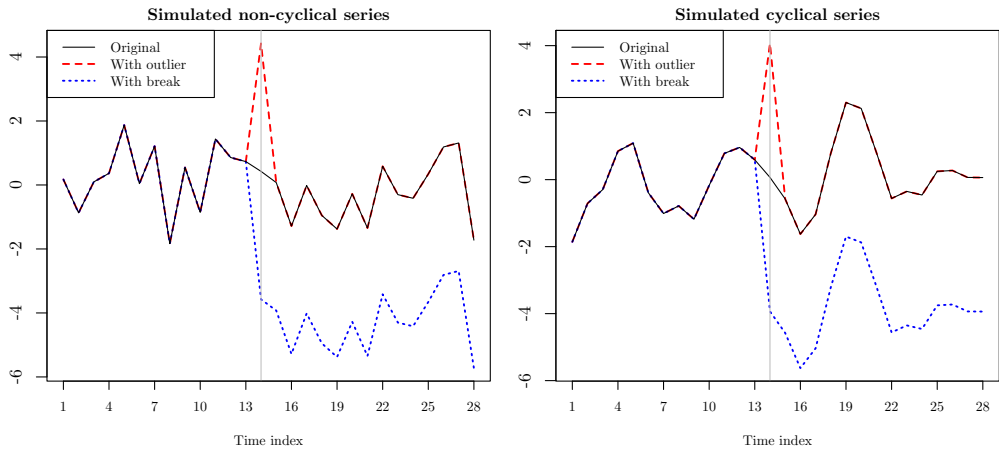


Figure 3. Cyclical and non-cyclical series with and without an outlier and a break. *Notes:* This figure displays two simulated paths from \mathcal{S}_1 , where one is non-cyclical and the other is cyclical. The original simulations are given by the straight black lines. The dashed red lines correspond to their transformation where an outlier with size $\delta = 4$ occurs at time $s = 14$. The dotted blue lines correspond to the transformation where a structural break with size $\delta = 4$ occurs at time $s = 14$.

mix of the original time series and the time series with either outliers or structural breaks at different times s and different values of $\delta \in \{0, 1, 2, 3\}$. Note that the data sets \mathcal{S}_i^* and \mathcal{S}_j^* for $i \neq j$ still differ in the length of the cycle period.

Figs. 4, 5, 6, and 7 display the performance measures with accuracy rates on the left columns, true positive rates on the middle columns, and true negative rates on the right columns. Figs. 4 and 5 focus on outliers, where the results in Fig. 4 are such that the NNs are trained on data without outliers, whereas in Fig. 5 the NNs are trained with outliers. Analogously, Figs. 6 and 7 focus on structural breaks, where the results in Fig. 6 are such that the NNs are trained on data without breaks, whereas in Fig. 7 the NNs are trained with breaks. In particular, the test data sets are \mathcal{S}_j^* , $j = 1, \dots, 5$, and in Figs. 4 and 6 the NN models are trained on $\cup_{1 \leq i \leq 4} \mathcal{S}_i \setminus \mathcal{S}_j$ for $j = 1, \dots, 4$ and $\cup_{1 \leq i \leq 4} \mathcal{S}_i$, respectively, whereas in Figs. 5 and 7 the NN models are trained on $\cup_{1 \leq i \leq 4} \mathcal{S}_i^* \setminus \mathcal{S}_j^*$ for $j = 1, \dots, 4$ and $\cup_{1 \leq i \leq 4} \mathcal{S}_i^*$, respectively. The results are broken down by length of cycle periods in each row of the figures, i.e., the test data sets \mathcal{S}_j^* for $j = 1, \dots, 5$.

The left columns of Fig. 4 show that the performance of all methodologies deteriorates with the size of the outliers in all test data sets. The accuracy rate of the AR(p) models falls quickly at $\delta = 2$ below that of the uncorrected g -test. Interestingly, the g -test without Bonferroni correction is the least impacted by the size of the outlier except on the test data \mathcal{S}_3^* . In particular, in most cases, the g -test gives the highest accuracy rate from $\delta = 4$. Among the NN models, the architecture that is most robust to outliers is the convolutional NN, which has an accuracy rate consistently above that of the other architectures. However, as with all other methods except the uncorrected g -test, its accuracy rate drops near 60% for large outliers. From the middle and right columns of Fig. 4, the deterioration of the performance of all models appears to be due to a high rejection rate of cyclical-ity leading to poor true positive rates, while true negative rates are close to 100%. Shifting now the focus towards Fig. 5, it turns out that training the NN models on data with outliers significantly improves their performance, except for $\delta = 5$ on the test data \mathcal{S}_3^* . Discounting that exception, the accuracy rates of the convolutional, the recurrent, and the hybrid architectures retain levels of accuracy close to their levels without outliers (i.e. $\delta = 0$). However, the accuracy rates of the fully connected NN architecture fall towards those of the uncorrected g -test despite being trained on data with outliers. Overall, the figures suggest that once trained on data containing series with

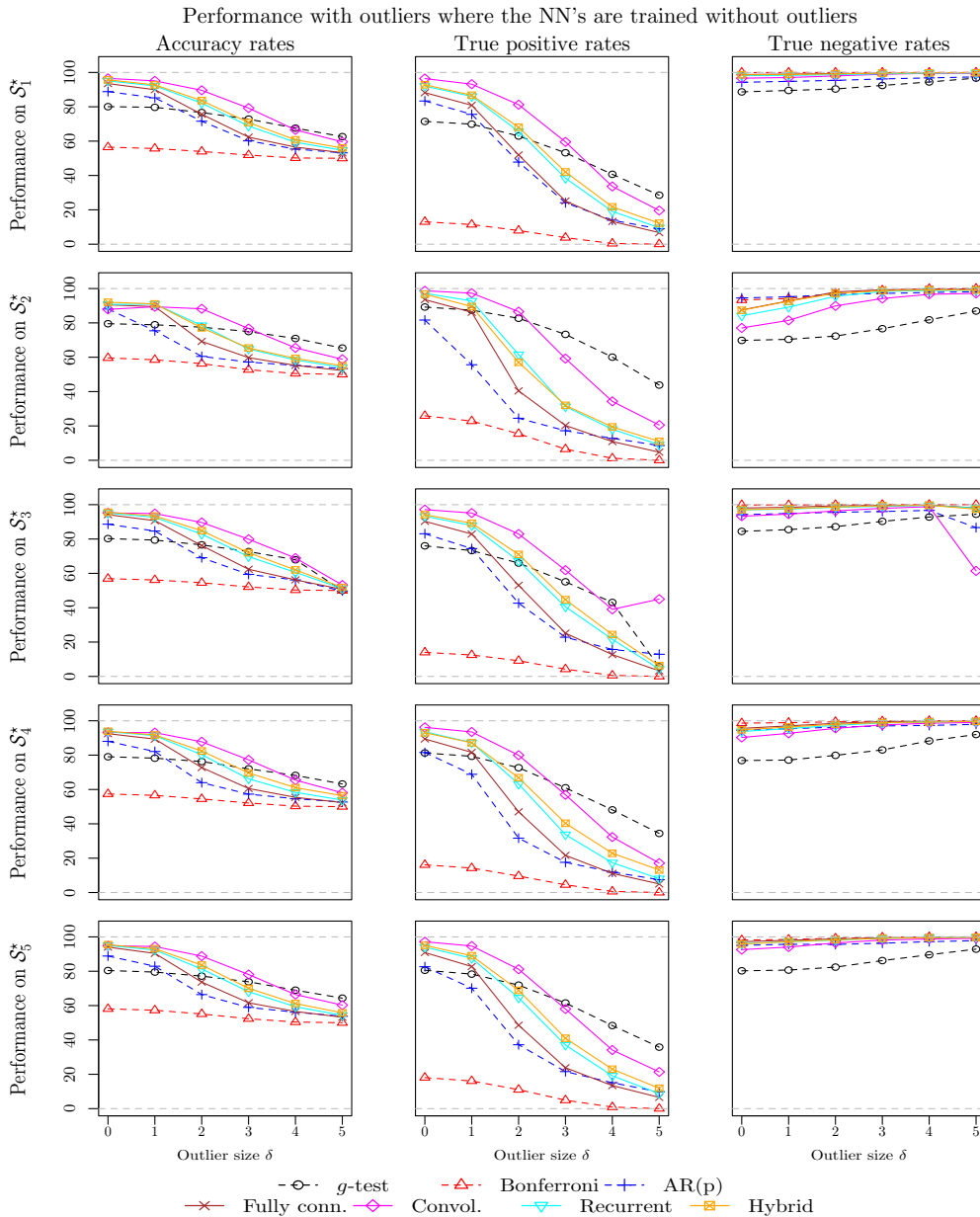


Figure 4. Accuracy, sensitivity, and specificity of all methodologies in the presence of outliers where the NN models are trained on data without outliers. *Notes:* This figure displays the accuracy rates (left columns), the true positive rates (middle columns) and the true negative rates (right columns) in function of the size of the outliers $\delta = 0, 1, 2, 3, 4$ and 5 for all methodologies on the test data S_j^* when the training data set is $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, for $j = 1, 2, 3, 4$, as well as the performance on the test data S_5^* when the training data set is $\cup_{1 \leq i \leq 4} S_i$, i.e. the training data sets do not contain outliers. The results from the autoregressive models and the g -test with p -value threshold 0.05 with and without Bonferroni correction are given by the dashed lines. The results from the neural networks are given by the straight lines.

outliers, the top-performing model is the hybrid NN, closely followed by the convolutional or the recurrent models. The methodologies based on the FNN, AR(p), and the g -test are too sensitive to the presence of outliers compared to the CNN, RNN, and the hybrid NN.

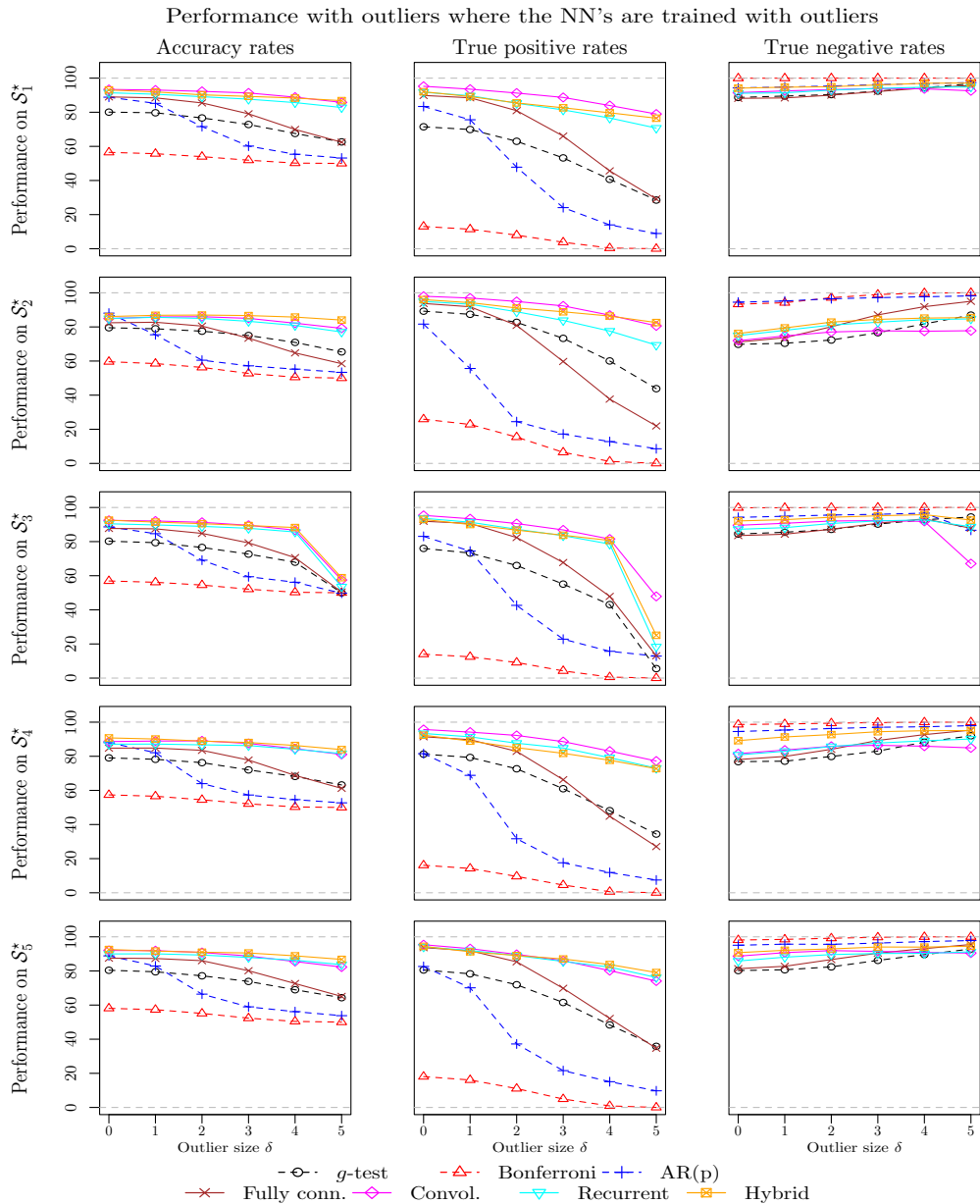


Figure 5. Accuracy, sensitivity, and specificity of all methodologies in the presence of outliers where the NN models are trained on data with outliers. *Notes:* This figure displays the accuracy rates (left columns), the true positive rates (middle columns), and the true negative rates (right columns) in function of the size of the outliers $\delta = 0, 1, 2, 3, 4$ and 5 for all methodologies on the test data S_j^* when the training data set is $\cup_{1 \leq i \leq 4} S_i^* \setminus S_j^*$, for $j = 1, 2, 3, 4$, as well as the performance on the test data S_5^* when the training data set is $\cup_{1 \leq i \leq 4} S_i^*$, i.e. the training data sets contain outliers. The results from the autoregressive models and the g -test with p -value threshold 0.05 with and without Bonferroni correction are given by the dashed lines. The results from the neural networks are given by the straight lines.

The left columns on Fig. 6 show that the accuracy rates of all models are also decreasing with the size of the structural break. Nevertheless, the figure suggests that the accuracy rates of NN models are less sensitive to the presence of structural breaks than outliers, with the exception

Performance with breaks where the NN's are trained without breaks

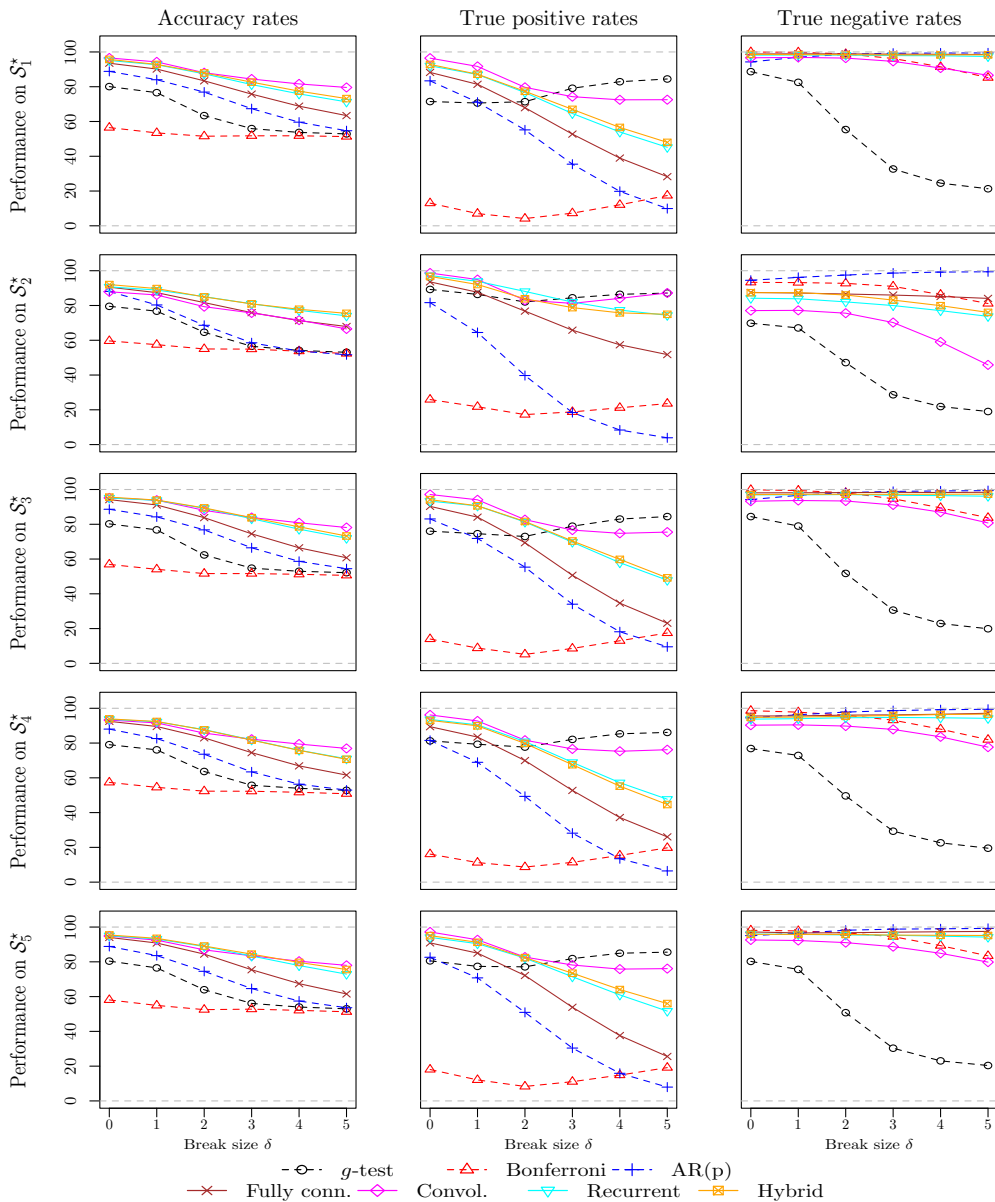


Figure 6. Accuracy, sensitivity, and specificity of all methodologies in the presence of structural breaks in the mean where the NN models are trained on data without breaks. *Notes:* This figure displays the accuracy rates (left columns), the true positive rates (middle columns) and the true negative rates (right columns) in function of the size of the break $\delta = 0, 1, 2, 3, 4$ and 5 for all methodologies on the test data S_j^* when the training data set is $\cup_{1 \leq i \leq 4} S_i \setminus S_j$, for $j = 1, 2, 3, 4$, as well as the performance on the test data S_5^* when the training data set is $\cup_{1 \leq i \leq 4} S_i$, i.e. the training data sets do not contain breaks. The results from the autoregressive models and the g -test with p -value threshold 0.05 with and without Bonferroni correction are given by the dashed lines. The results from the neural networks are given by the straight lines.

of the fully connected NN whose accuracy rates fall near those of the AR(p) and the g -test. The reasonable performance of the convolutional, the recurrent, and the hybrid architectures seems to be associated with a mild decrease in the true positive rate compared to the case with

Performance with breaks where the NN's are trained with breaks

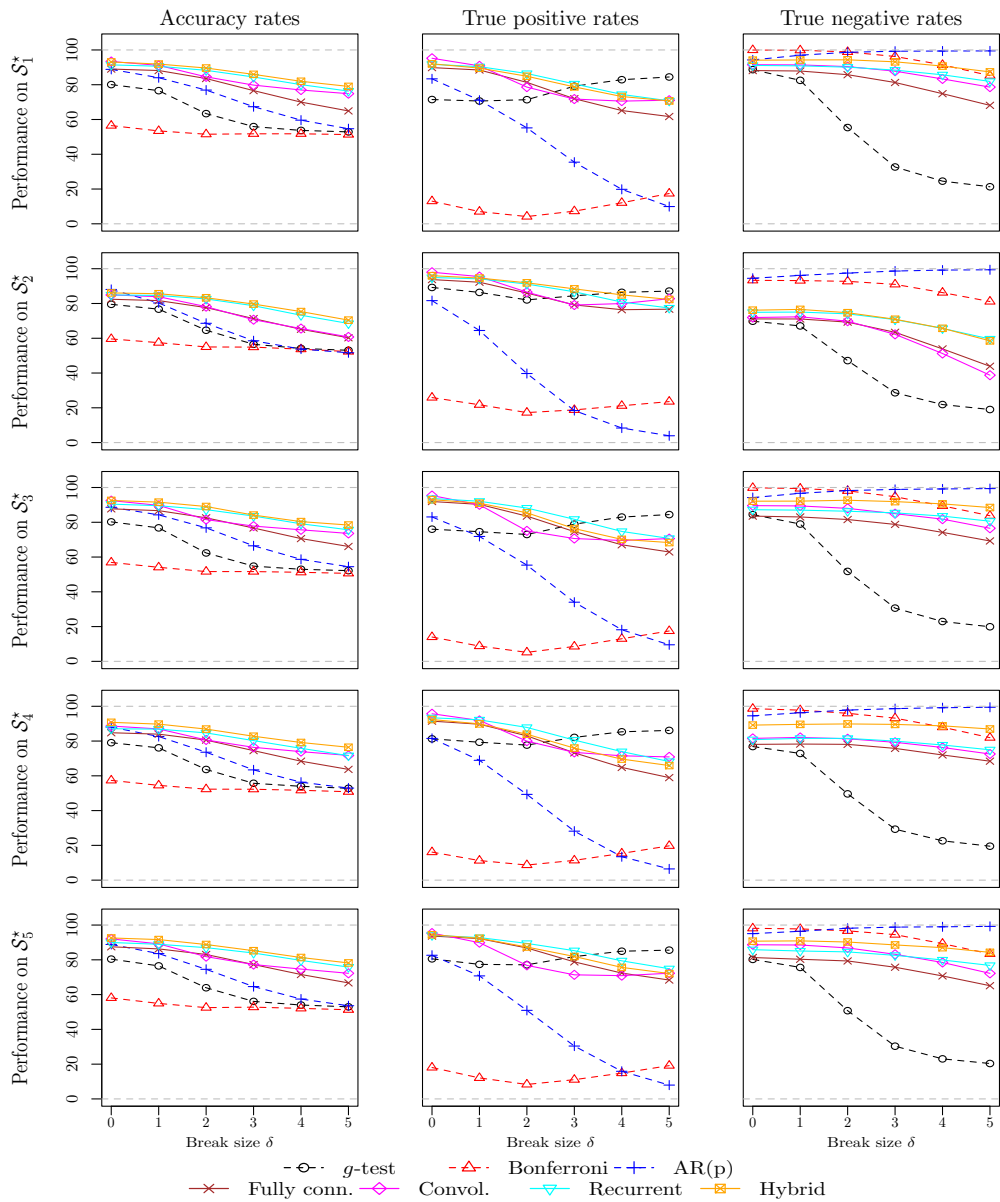


Figure 7. Accuracy, sensitivity, and specificity of all methodologies in the presence of structural breaks in the mean where the NN models are trained on data with breaks. *Notes:* This figure displays the accuracy rates (left columns), the true positive rates (middle columns), and the true negative rates (right columns) in function of the size of the break $\delta = 0, 1, 2, 3, 4$ and 5 for all methodologies on the test data S_j^* when the training data set is $\cup_{1 \leq i \leq 4} S_i^* \setminus S_j^*$, for $j = 1, 2, 3, 4$, as well as the performance on the test data S_5^* when the training data set is $\cup_{1 \leq i \leq 4} S_i^*$, i.e. the training data sets contain breaks. The results from the autoregressive models and the g -test with p -value threshold 0.05 with and without Bonferroni correction are given by the dashed lines. The results from the neural networks are given by the straight lines.

outliers. Furthermore, Fig. 7 shows that training the NN models on data containing structural breaks improves their accuracy rate and the true positive rates, but the improvement is less noticeable than in the case of data with outliers. Where the NN models are trained with data that include structural breaks (i.e. Fig. 7), the top-performing model is again the hybrid architecture.

Table 3. Analysis of price cycles

	Granular data (% of profiles)	Aggregate data
<i>g</i> -test	70.00%	Non-cyclical
Bonferroni	22.68%	Non-cyclical
<i>Neural network trained without outliers and structural breaks</i>		
FNN	85.12%	Cyclical
RNN	93.41%	Cyclical
CNN	92.44%	Cyclical
Hybrid NN	92.20%	Cyclical
<i>Neural network trained with outliers and structural breaks</i>		
FNN	96.34%	Cyclical
RNN	99.02%	Cyclical
CNN	98.29%	Cyclical
Hybrid NN	98.78%	Cyclical

Notes: The middle column of this table reports the proportion of cyclical time series of prices from the classification of all models. The right column reports the outcome of the classification of all models for the data aggregated at the market level, i.e. the average market price where the average is weighted by the exposure. The neural network models were trained using either all simulated data $\cup_{1 \leq i \leq 5} \mathcal{S}_i$ without outliers and structural breaks or all simulated data $\cup_{1 \leq i \leq 5} \mathcal{S}_i^*$ with outliers and structural breaks.

The superior performance of NN models in the presence of outliers and structural breaks can be attributed to their ability to learn from diverse training data and extract underlying patterns rather than relying on parametric assumptions. Traditional AR models estimate coefficients that are directly impacted by sudden changes in the mean or extreme observations, leading to instability in cycle classification. In contrast, NN architectures learn representations that emphasize recurrent patterns while de-emphasizing transient distortions. By training NN models on both “clean” and “perturbed” series, they develop robustness to irregularities that would otherwise bias AR-based detection methods.

Overall, the analysis suggests that for a training data set without outliers, the hybrid architecture outperforms all other models when there are no outliers or structural breaks, but that performance is sensitive to the presence of such features, and the least sensitive architecture is the convolutional NN. However, once trained on data that contain outliers or structural breaks, the hybrid architecture is to be preferred. The methodologies based on AR(p) or the *g*-test are too sensitive to the presence of outliers and structural breaks, and their performance deteriorates with the size of the outlier and the break, especially for the AR(p). Of course, it is worth noting that the performance of the AR(p) and *g*-test could possibly be improved using an appropriate method to correct the time series for the outliers and structural breaks. Such corrections were not used here as they are not common in the literature on the underwriting cycle.

5. Empirical study

This section reports the results of the classification of the AUTOSEG data of prices. Recall that based on the simulation study in the previous section, the NN models stand out as the most accurate on the test data for cycle detection, especially the hybrid architecture trained on data with outliers or structural breaks in the means. Nevertheless, for the sake of completeness, this section reports the classification outcome of all methods. Note that in this section, the NN models were trained using all simulated data, namely either on $\cup_{1 \leq i \leq 5} \mathcal{S}_i$ for the models trained on data without outliers and breaks, or on $\cup_{1 \leq i \leq 5} \mathcal{S}_i^*$ for the models trained on data with outliers and breaks.

The classification outcome from all methods is reported in Table 3, containing the proportions of price time series classified as cyclical in the middle column, where the classification is performed per risk profile, as well as the classification outcome at the aggregate level in the right column, using average market price across all risk profiles weighted by the exposure.

The findings provide strong evidence that insurance prices for almost all risk profiles exhibit cyclical behavior. The AR(p) methodology typically used in the literature detects cyclicity in 79.51% of risk profiles, while the g-test without correction detects it in 70%. The g-test with Bonferroni correction detects cyclicity in 20.68%, but as demonstrated in the simulation study, this correction leads to over-rejection of cyclicity. The NN models identify cyclicity in proportions exceeding 90%, with the hybrid architecture trained on data with outliers and breaks (i.e. the top-performing model in the simulation study) detecting cyclicity in 98.78% of the time series, or 405 out of 410 risk profiles. The classification of aggregate prices is consistent with that of the granular data, where all methodologies classify the average market price as cyclical, except for the g-tests. To the best of my knowledge, this is the first paper that provides evidence of cyclicity in insurance prices, and at this level of granularity.

The interpretation of the above results can be subject to two important criticisms.

The first one is that, strictly speaking, the NN models employed in this paper are trained to distinguish between cyclical AR(2) and cosine-based series from AR(1)/AR(0), and do not necessarily identify all types of cyclicity. The simulation study, although comprehensive in the treatment of the standard processes considered in the literature, does not investigate how NN models would classify series with complex but not cyclical temporal structure. In particular, the simulation study in Section 4 does not address the performance of NNs trained on noisy sinusoids and AR(p) with $p \leq 2$ when the series to be classified are e.g. non-cyclical AR(3) models, heteroscedastic, and/or exhibiting both outliers and structural breaks. Nevertheless, the preliminary tests performed on the data suggest absence of heteroscedasticity, and i.i.d. residuals when the series are modeled using AR(1) or AR(2) models. In a separate analysis, an AR(3) was estimated for all series, but it outperforms the AR(1) and AR(2) in only a few instances, namely for 20 series of price. For all these series, the condition of AR(3) cyclicity was investigated numerically by verifying whether the characteristic equation of the process has a pair of complex roots, and both AR(3) and AR(2) suggest cyclicity.

The second criticism of this study is that the data set encompasses only 14 years (approximately 3 cycles based on the estimation of the AR(2) on aggregate premiums), making it challenging to identify cyclicity from an econometric perspective. However, the simulation study demonstrates that NNs were able to detect cyclical AR(2) patterns and noisy sinusoids across various cycle lengths with the same number of observations as the empirical study. Therefore, based on the simulation study results, the number of observations in the empirical analysis is not a significant concern.

6. Conclusion

The present paper addresses the methodological issue of cycle detection in the insurance industry using NNs. The efficacy of NNs is demonstrated on simulated data and compared to that of traditional methods. The simulation study highlights the performance of NN models in cycle detection under various settings, including when the data contain outliers and structural breaks. The NN-based methodology is then applied to a novel granular data set from the Brazilian insurance industry of prices per risk profile. The application shows that, despite the skepticism expressed in the literature, insurance cycles exist, and they manifest themselves in prices of almost all risk profiles.

While this study focuses on NNs, other machine learning techniques, such as gradient boosting, could also be considered for cycle detection. Tree-based methods have shown strong classification performance in various applications and could, therefore, also perform well in contexts where temporal dependencies in time series have to be taken into account. Nevertheless, NN architectures, particularly convolutional and recurrent models, are designed to capture sequential patterns, making them well-suited for this task. Future research could explore the performance of

alternative machine learning methods, but the findings presented here highlight the advantages of NNs in cycle detection.

The results of this study raise several avenues for future research. First, the high classification accuracy of NNs suggests that they could be adapted to detect other time-series characteristics, such as stationarity, autocorrelation, or seasonality. Extending the current methodology to a broader set of time-series properties could provide further insights into insurance market dynamics, and it would also be relevant more generally in the context of time-series analysis. Second, a natural extension is to develop methods for identifying cycle peaks and troughs, which could assist actuaries in designing more responsive pricing strategies. This could involve integrating NN-based cycle detection with change-point detection techniques or state-space models. Third, Fig. 1 suggests that insurance prices for different risk profiles exhibit a degree of synchronization. Future research could investigate whether this synchronicity is driven by common macroeconomic factors, regulatory changes, or market competition. This could be explored using multivariate models or causal inference techniques to better understand the interdependencies between risk profiles.

Data availability statement. The code that supports the findings of this study is available on request from the author. The data source is provided in the main text.

Funding statement. This work was supported by Fundación MAPFRE, grant Ignacio H. de Larramendi for insurance and social protection.

Competing interests. The author declares none.

References

- Aggrawal, C. (2018). *Neural networks and deep learning: A textbook*.
- Boyer, M., Jacquier, E., & Van Norden, S. (2012). Are underwriting cycles real and forecastable? *Journal of Risk and Insurance*, 79(4), 995–1015.
- Boyer, M., & Owadally, I. (2015). Underwriting apophenia and cryptids: Are cycles statistical figments of our imagination? *The Geneva Papers*, 40(2), 232–255.
- Brockwell, P., & Davis, R. (2002). *Introduction to time series and forecasting*. Springer.
- Bruneau, C., & Sghaier, N. (2015). Cyclicity in the french property-liability insurance industry: New findings over the recent period. *Journal of Risk and Insurance*, 82(2), 433–462.
- Chen, R., Wong, K. A., & Lee, H. C. (1999). Underwriting cycles in Asia. *Journal of Risk and Insurance*, 66(1), 29–47.
- Choi, S., Hardigree, D., & Thistle, P. (2002). The property/liability insurance cycle: A comparison of alternative models. *Southern Economic Journal*, 68, 530–548.
- Chollet, F., & Allaire, J. J. (2018). *Deep learning with R*. Manning Publications.
- Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on Loess. *Journal of Official Statistics*, 6, 3–73.
- Cummins, D., & Outreville, F. (1987). An international analysis of underwriting cycles in property-liability insurance. *Journal of Risk and Insurance*, 54(2), 246–262.
- Doherty, N., & Kang, H. B. (1988). Interest rates and insurance price cycles. *Journal of Banking and Finance*, 12(2), 199–214.
- Emms, P. (2012). Equilibrium pricing of general insurance policies. *North American Actuarial Journal*, 16(3), 323–349.
- Feldblum, S. (2001). Underwriting cycles and business strategies. In *Proceedings of the casualty actuarial society* (pp. 175–235), vol. 88.
- Fisher, R. A. (1929). Tests of significance in harmonic analysis. *Proceedings of the Royal Society of London*, 125, 54–59.
- Frazão, A. (2020). Control of price related terms in standard form contracts in Brazil. In *Control of price related terms in standard form contracts*, IUS Comparatum - Global Studies in Comparative edn. Springer.
- Fun, H.-G., Lai, G. C., Patterson, G. A., & Witt, R. C. (1998). Underwriting cycles in property and liability insurance: An empirical analysis of industry and by-line data. *Journal of Risk and Insurance*, 65(4), 539–561.
- Gabrielli, A. (2020). A neural network boosted double overdispersed poisson claims reserving model. *ASTIN Bulletin*, 50(1), 25–60.
- Grace, M., & Hotchkiss, J. (1995). External impacts on the property-liability insurance cycle. *Journal of Risk and Insurance*, 62(4), 738–754.
- Gron, A. (1994). Capacity constraints and cycles in property-casualty insurance markets. *The RAND Journal of Economics*, 25(1), 110–127.

- Hainaut, D.** (2018). A neural-network analyzer for mortality forecast. *ASTIN Bulletin*, **48**(02), 481–508.
- Hartung, J.** (1999). A note on combining dependent tests of significance. *Biometrical Journal*, **41**(7), 849–855.
- Higgins, M., & Thistle, P.** (2000). Capacity constraints and the dynamics of underwriting profits. *Economic Enquiry*, **38**(3), 442–457.
- IMF (2012). Brazil: Detailed assessment of observance of insurance core principles of the International Association of Insurance Supervisors. IMF report No. 12/334.
- Lamm-Tennant, J., & Weiss, M.** (1997). International insurance cycles: Rational expectations/institutional intervention. *Journal of Risk and Insurance*, **64**(3), 415–439.
- Lazar, D., & Denuit, M.** (2012). New evidence for underwriting cycles in US property-liability insurance. *Journal of Risk Finance*, **13**(1), 4–12.
- Li, J.** (2022). A model stacking approach for forecasting mortality. *North American Actuarial Journal*, Available online.
- Lindholm, M., Richman, R., Tsanakas, A., & Wüthrich, M.** (2022). Discrimination-free insurance pricing. *ASTIN Bulletin*, **52**(1), 55–89.
- Malinovskii, V.** (2010). Competition-originated cycles and insurance strategies. *Astin Bulletin*, **40**, 797–843.
- Malinovskii, V.** (2013a). Rationale of underwriters pricing conduct on competitive insurance market. *Insurance: Mathematics and Economics*, **53**(2), 325–333.
- Malinovskii, V.** (2013b). Reflexivity in competition-originated underwriting cycles. *Journal of Risk and Insurance*, **81**(4), 883–905.
- Meier, U., & Outreville, F.** (2006). Business cycles in insurance and reinsurance: the case of France, Germany and Switzerland. *Journal of Risk Finance*, **7**(2), 160–176.
- Niehaus, G., & Terry, A.** (1993). Evidence on the time series properties of insurance premiums and causes of the underwriting cycle: New support for the capital market imperfection hypothesis. *Journal of Risk and Insurance*, **60**(3), 466–479.
- Pentikäinen, T., & Rantala, J.** (1982). *Solvency of insurers and equalization reserves Vol. I and Vol II*. The Insurance Publishing Company.
- Rumelhart, D., Hinto, G., & Williams, R.** (1986). Learning internal representations by backpropagating errors. In *Parallel distributed processing: Explorations in the microstructure of cognition* (pp. 318–362).
- Shi, P., & Shi, K.** (2022). Non-life insurance risk classification using categorical embedding. *North American Actuarial Journal*, Available online.
- Stewart, B. D., Stewart, R. E. & Roddis, R. S. L.** (1991). A brief history of the underwriting cycle. In *Appendix to “cycles and crises in property-casualty insurance: Causes and implications for public policy”*. National Association of Insurance Commissioners.
- Taylor, G. C.** (1986). Underwriting strategy in a competitive insurance environment. *Insurance: Mathematics and Economics*, **5**(1), 59–77.
- Trufin, J., Albrecher, H., & Denuit, M.** (2009). Impact of underwriting cycles on the solvency of an insurance company. *North American Actuarial Journal*, **13**(3), 385–403.
- Venezian, E.** (1985). Ratemaking methods and profit cycles in property and liability insurance. *Journal of Risk and Insurance*, **52**(3), 477–500.
- Venezian, E. C.** (2006). The use of spectral analysis in insurance cycle research. *Journal of Risk Finance*, **7**(2), 177–188.
- Venezian, E., & Leng, C.-C.** (2006). Application of spectral and ARIMA analysis to combined-ratio patterns. *Journal of Risk Finance*, **7**(2), 189–214.
- Wang, N., & Murdock, M.** (2019). A dynamic model of an insurer loss shocks, capacity constraints and underwriting cycles. *Journal of Risk Finance*, **20**(1), 82–93.
- Winter, R.** (1988). The liability crisis and the dynamics of competitive insurance markets. *Yale Journal on Regulation*, **5**.