Roots from Trees – A Machine Learning Approach to Unit Root Detection

A Preprint

Gary Cornwall *
Office of the Chief Economist
Bureau of Economic Analysis
Suitland, MD USA
Gary.Cornwall@bea.gov

Jeffrey Chen
Bennett Institute for Public Policy
University of Cambridge
Stockholm, Sweden
contact@jeffchen.org

Beau Sauley
Department of Economics and Finance
Murray State University
Murray, KY USA
bsauley@murraystate.edu

January 3, 2022

Abstract

In this paper we draw inspiration from the ensemble forecasting and model averaging literature and use a gradient boosting algorithm to exploit variation between test statistics used to determine if a series contains a unit root. The result is a *pseudo*-composite ML-based test for unit roots which is four to six percentage points more accurate than the next best traditional test. Through a train-validation framework this method allows for control over Type I error rates and the gains in power come with little variation in specificity (empirical size). Additionally, the proposed method is agnostic towards deterministic elements traditionally needed in the established testing environment and thus closes off an additional error path for unit root testing; that of model misspecification. We illustrate this new testing procedure by

*The authors would like to acknowledge Peter C.B. Phillips, Tara Sinclair, Marina Gindelsky, Scott Wentland, Jeff Mills, Jeremy Moulton, and Olivier Parent for their helpful comments. The views expressed here are those of the authors and do not represent those of the U.S. Bureau of Economic Analysis or the U.S. Department of Commerce.

applying it to an established benchmark data set and examining the state-level hypothesis of unemployment hysteresis.

Keywords Integrated Processes · Forecast or Model Averaging · Time Series · Machine Learning

1 Introduction

The identification of time-series which contain a unit root has important implications for data users and researchers. Granger and Newbold (1974) demonstrated this through the use of Monte Carlo simulation and showed that failure to identify the presence of a unit root can lead to 'nonsense regressions'. Over the next five decades there was a great deal of research into test statistics which, under various conditions, improved the ability for a researcher to identify time-series containing a unit root (see Dickey and Fuller (1981), Phillips and Perron (1988), Schmidt and Phillips (1992), Elliot et al. (1996), Zivot and Andrews (2002), and Jansson and Nielsen (2012) among many others). Despite the bevy of options available to researchers interested in time-series analysis, there remains a great deal of skepticism regarding available tests stemming primarily from evidence of low power when the root of an autoregressive polynomial is near, but not equal to, one (Ng and Perron, 2001).[2]

Our primary interest in this paper lies not in the development of a new test, but rather exploiting existing tests in a new framework designed to maximize a practitioner's chance of identifying a unit root. Moreover, we would like this framework be agnostic to the choice of deterministic elements assumed as part of the data generating process (*e.g.,* drift or trend terms) so as to limit the role of model misspecification in the testing procedure. To do this we draw inspiration from the ensemble forecasting literature, recognizing that each test for a unit root (*e.g.,* the ADF test (Said and Dickey, 1984) or the DF-GLS (Elliot et al., 1996)) can bring unique information to the table.

To wit, we will produce this *pseudo*-composite test by first showing that a hypothesis test such as the ADF is, in the context of machine learning, a weak learner known as a decision stump. Second, we will show that in a group of tests which include the ADF (Dickey and Fuller, 1981), PP (Phillips and Perron, 1988), KPSS (Kwiatkowski et al., 1992), PGFF (Pantula et al., 1994), Breit (Breitung, 2002), (Breitung and Taylor, 2003), ERS-d and ERS-p (Elliot et al., 1996), URSP (Schmidt and Phillips, 1992), and URZA (Zivot and Andrews, 2002) that there is disagreement regarding the presence of a unit root, and that there exists interesting variation between the statistics which can be exploited to gain accuracy. Finally, we will use a well known gradient boosting algorithm, known as XGboost (Chen and Guestrin, 2016), to exploit this variation and

[2]The detection of a unit root is a critical function in the modeling of time-series data. As a result, more accurate tests for this data feature are important to statistical agencies such as the U.S. Bureau of Economic Analysis, U.S. Census Bureau, and Bureau of Labor Statistics among many others.

more accurately identify time series containing a unit root while being agnostic to the choice of deterministic elements in the data generating process.

More recently, research in this area has focused primarily on tests which are considered "nearly efficient", or those with asymptotic power functions which are at, or near, the Gaussian power envelope (Elliot et al. (1996), Ng and Perron (2001), Jansson (2008), Jansson and Nielsen (2012)). For example, the DF-GLS test (Elliot et al., 1996), which applied a Dickey-Fuller t test to locally demeaned or detrended time series, has good performance with respect to finite sample size and power. Moreover, they show that this modified Dickey-Fuller t test lies near the power envelope and thus little power gains could be expected under similar assumptions with larger samples. Jansson (2008) derived similar asymptotic power envelopes for test in cases where one is faced with a zero-mean autoregressive model. Jansson and Nielsen (2012) showed that likelihood ratio tests, point optimal tests, and DF-GLS tests share these types of near-optimality properties. We agree that pushing the power envelope for any single test may or may not be feasible. However, in the case of our proposed method we are jointly evaluating multiple tests, some of which may be close to the power envelope, and exploiting variation between them to exceed that limiting factor on power.

In our case, the joint evaluation of nine test statistics – all of which are common in the literature – produces power which exceeds that of any single test by at least four percentage points while having comparable empirical size. It is important to note that this power increase is a pessimistic improvement as it assumes a practitioner knows precisely what deterministic trends to include in the test structure, and that they use the most powerful test. If we are unconcerned about maintaining a fixed size and prefer to maximize accuracy we find that the power is nearly sixteen percentage points higher at the cost of only nine percentage points in size. Common classification measures such as the F-score and Matthew's Correlation Coefficient (Matthews, 1975; Hastie et al., 2009; Kuhn et al., 2013; Tharwat, 2018) show the proposed method to be better than any standard test alternative regardless of the fixed size preference.[3] Moreover, the framework we establish here allows for future test statistics, additional current test statistics, or alternative data generating processes to be added as necessary.

The remainder of this paper is structured as follows. In Section 2 we describe the unit root problem, illustrate that there is variation even under the null in how tests classify a series, and show how a single test such as the Augmented Dickey Fuller test can be viewed as a decision stump, the most basic of weak learners. Section 3 lays out an ML-based unit root test which delivers marked improvement in test accuracy. Section 4 provides a look at two empirical applications. First, we apply the proposed test to macroeconomic time-series which have long been used as a benchmark dataset in the unit root literature (Nelson and Plosser, 1982). Second,

[3]Earlier versions of this paper had additional power improvements based on different data generating processes that mirrored the regressions used in the Augmented Dickey-Fuller test rather than those contained herein.

using state-level, not-seasonally-adjusted unemployment rate data we examine the hysteresis hypothesis in the continental United States. Finally, Section 5 concludes and offers avenues for additional research.

2 Unit Root Tests and Weak Learners

In this section we define the unit root problem and outline several of the challenges practitioners face when employing statistical tests. Additionally, we show how a single test statistic (*e.g.,* the Augmented Dickey-Fuller Test) is equivalent to a weak learner. This equivalence facilitates our use of gradient boosting to reconcile the differences between tests.

2.1 Challenges to Consistent Unit Root Testing

To begin, let y_t be an autoregressive time-series generated from,

$$y_t = \beta' d_t + \mu_t, \tag{1}$$
$$\mu_t = \rho \mu_{t-1} + \epsilon_t$$

where d_t is a set of deterministic elements such that $d_t = 1$ or $d_t = (1, t)'$, β is an unknown parameter or vector of parameters, and ϵ_t is an unobserved zero-mean white noise process. The primary interest here lies in a null hypothesis that $\rho = 1$ versus an alternative of $\rho < 1$.

While amongst the most studied statistical phenomena in time series, the ability to detect a unit root is impacted not only by which test is applied but whether the practitioner applies it correctly. The most common test statistic recognized by practitioners, but not necessarily the most powerful, is the DF test (Dickey and Fuller, 1979; Said and Dickey, 1984) which is a t-type test for $\phi_0 = 0$ in the following regression:

$$\Delta y_t = \beta' d_t + \phi_0 y_{t-1} + \sum_{p=1}^{k} \phi_p \Delta y_{t-p} + \epsilon_{tk} \tag{2}$$

where Δ denotes a first difference operator. New tests or modifications to existing tests have been developed over subsequent years to deal with size distortions in the presence of a moving average term (Perron and Ng, 1996), the existence of structural breaks (Zivot and Andrews, 2002), and low power (Elliot et al., 1996), among many other features and/or assumptions.

In addition to the variety of tests designed for different conditions, the manner in which tests are applied also introduces variability into their effectiveness. One important aspect to keep in mind for many of these tests is the choice of elements to include in d_t. For example, including both a drift and trend term in the regression equation when it is not necessary results in lost power, a problem already of concern for tests of a unit root (Ng and Perron, 2001; Kennedy, 2008); while not including enough of them leads to a bias towards the null. A practitioner needs to evaluate the plausibility of each possible case (*e.g.,* no drift or trend, trend but no drift,

etc.) and choose the specification accordingly. Moreover, the plausibility of any particular specification being the appropriate one to use can be argued reasonably between practitioners when evaluating the same series. Since even the same test may not reject the null under all possible cases this is incredibly important to get right. The choice of including a drift or trend term necessitates a testing strategy be laid out for practitioners since the results may not be consistent across scenarios with different assumptions, see Elder and Kennedy (2001) for one such discussion. We view this as an additional error path beyond the traditional Type I and II errors common in hypothesis testing and will be an important part of our discussion going forward.

Finally, even if a practitioner chooses the correct deterministic elements it is possible that one test may disagree with another. For example, suppose we generate data from,

$$\begin{aligned} y_t &= \mu_t, \\ \mu_t &= \rho\mu_t + \epsilon_t \end{aligned} \tag{3}$$

where $y_0 = 0$, $\rho = 1$, and ϵ_t is identically and independently distributed from a standard normal distribution. We then apply the DF test under the assumption of no deterministic elements, and the DF-GLS test put forth by Elliot et al. (1996). In this scenario, the null is true and as a result we would expect to reject the null α proportion of the time. While this is the case, both tests reject approximately five out of every one-hundred tests, it is not the case that they reject the same five series. In fact, when we run this repeatedly we find that on average the two tests combined will reject about eight out of every one-hundred series (using a 5% critical value for each); however, of those eight, the two tests only agree on four. Even more variation exists when the alternative is true, a point we will return to in Section 3.

The disagreement and variation amongst individual tests can be problematic, only if their differences are left unreconciled. We exploit this variation by employing a classification algorithm as a mapping function in place of the traditional null distribution. We simulate data from Equation 1 under a variety of conditions with respect to d_t and train the algorithm to differentiate series containing a unit root from those that do not. Included in the features we use to train this algorithm are nine different tests for a unit root (or stationarity) including the ADF (Dickey and Fuller, 1981), PP (Phillips and Perron, 1988), KPSS (Kwiatkowski et al., 1992), PGFF (Pantula et al., 1994), Breit (Breitung, 2002), (Breitung and Taylor, 2003), ERS-d and ERS-p (Elliot et al., 1996), URSP (Schmidt and Phillips, 1992), and URZA (Zivot and Andrews, 2002). In this way we are exploiting the between statistic variation and mapping the joint distribution of the hypothesis test space.[4]

[4]The simulation environment can be arbitrarily expanded to include other DGPs that are pertinent to the problem at hand. This environment reflects many DGPs that are relied upon in the literature when evaluating unit roots.

2.2 Weak Learners for Unified Testing

The variability between tests can be reconciled by framing the problem in terms of a binary classification. The typical classification model is developed and applied in three steps: *training, validation, and testing.* In the training step, a model learns from a sample of input features (i.e. right-hand side variables) in order to classify instances by one of two classes of a binary target (i.e. dependent variable). For example, let $\mathcal{D} = \{(x_1, y_1), \ldots, (x_M, y_M)\}$ be a set of training data, indexed by $m = (1, \ldots, M)$. It is assumed these observations are independent and identically distributed, with each x_m a realization of random variable $\mathcal{X}$ having support $\mathbb{R}$.[5] Following convention we will denote the collection of observations as a vector with the capital such that $\mathcal{D} = (X, Y)$. A classifier, h, is a mapping, $h : \mathcal{X} \to \{-1, +1\}$, which returns the predicted class, y, for each m, conditional upon x. For example, we can consider the following classifier,

$$y_m = \begin{cases} +1 \text{ if } x_m \in \zeta \\ -1 \text{ if } x_m \in \zeta^c \end{cases}, \tag{4}$$

where $\zeta \subset \mathcal{X}$ such that $\zeta \cup \zeta^c = \mathcal{X}$.[6] Both simple and complex classifiers can partition $\mathcal{D}$ into classifications $y = +1$ and $y = -1$ conditional upon $\mathcal{X}$. Interestingly, this is nearly identical to the definition of a "statistical test" as outlined by Neyman and Pearson (1933).

The objective of the validation step is to evaluate the performance of candidate models on an unbiased data set that is independent of the training sample. Model validation designs can take on any number of forms. A simple design involves splitting a sample into three randomly assigned partitions – a training set for fitting a model, a validation set to identify the best candidate model, and a test set to evaluate the best model's performance. In each step, the model validation design ensures identical and independently distributed samples, thereby eliminating bias in performance estimates. The design can also be evolved to meet the specific requirements of the application at hand. For example, the training sample could serve as its own validation sample through k-folds cross validation, allowing the validation set to be used to calibrate a decision threshold which reflects preferred relative costs between Type I and II errors, or one based upon a desired long run specificity (Type I error rate). The test sample remains as the final evaluation.

Only once a model achieves a desired level of accuracy when applied to the out-of-sample test can it advance to the second stage of the classification process– *scoring.* A model is said to score new, previously unseen

[5] Note that for simplicity we are assuming that there is but one observed x for each observation. It is trivial to assume that x_m is a K-dimensional vector of observed features such that $X \in \mathcal{X} \subset \mathbb{R}^K$.

[6] The use of a binary classification system is merely for its simplicity and readability. In practice $y \in Y$ where Y has some finite cardinality, k.

instances by mapping them to the hypothesis space, thereby resulting in a probability of belonging to the positive class ($Pr(y_m = +1|X)$). To convert a probability into a prediction of class membership requires one to make an explicit choice of a classification threshold. In a balanced two-class classifier, a threshold can be selected to explicitly reflect the Type I and Type II costs. With this in mind, we can consider traditional hypothesis tests, such as the ADF and KPSS, as effectively simple binary classification scoring models. Given striking resemblance, we map the similarities between hypothesis tests and classification problems in Table 1.

[Table 1 about here.]

It is clear that both classification problems, as outlined in the machine learning literature, and hypothesis tests have a great deal in common. The primary differences lie in the mapping function and the resulting choice of threshold. Let us begin with perhaps the simplest learning algorithm, the decision stump. A decision stump can be thought of as a severely pruned decision tree (Oliver and Hand, 1994), with its purpose to minimize the risk function $R(h) = \mathbb{P}(Y = -1)R_1(h) + \mathbb{P}(Y = +1)R_2(h)$ where $R_1(h) = \mathbb{P}(h(X) \neq Y|Y = -1)$ and $R_2(h) = \mathbb{P}(h(X) \neq Y|Y = 1)$ denote a Type I (false positive) and Type II (false negative) error respectively (Tong et al., 2016). One way to minimize this risk function is to evaluate the conditional probability density functions $f(x|y = +1)$ and $f(x|y = -1)$ from the training set. Assuming that the training sample is representative of the process, then – in a binary classification system – it can be shown that the minimized combination of Type I and II errors occurs where these two densities intersect (see Schapire and Freund (2013) pp. 28 for a discussion).[7]

[Figure 1 about here.]

Let us examine the Augmented Dickey-Fuller test statistic from Equation 2 as an illustrative example in Figure 1. In the top panel, we have plotted the ADF statistics, marking the decision threshold with a typical critical value of $\alpha = 0.05$ (-1.95). The density of all collected ADF statistics is represented by the dashed line, $f(x)$. Since we have simulated this data we know what the true value of ϕ_0 is, and thus we know the true disposition of the series and can outline the resulting conditional distributions $f(x|y = +1)$ when the null is false (*i.e.* the series is stationary), and $f(x|y = -1)$ when the null is true (*i.e.* the series contains a unit root). Here, we expect a long run Type I error rate of 5% as indicated by our choice of α (shaded area labeled e_1), and an indeterminate number of Type II errors (shaded area labeled e_2).

On the other hand, the decision stump has chosen the intersection point between the two conditional densities such that $\zeta \in (-\infty, x_0 \approx -1.03]$ and $\zeta^c \in (x_0 \approx -1.03, \infty)$ which leads to a larger Type I error rate but a

[7] In practice the intersection of the conditional densities is cumbersome computationally and the decision threshold is made via Gini Impurity or some other information based metric. We find this particular explanation useful to provide intuition and allow for consistency with common graphics shown in most texts covering hypothesis testing.

smaller Type II error rate. As mentioned earlier, this is the point in which the combination of Type I and Type II errors, conditional upon a $0-1$ loss function, is minimized. This means that the overall accuracy of the decision stump is strictly greater than that of the hypothesis test at the chosen $\alpha = 0.05$ for the hypothesis test. Note that both methods rely upon the support of $\mathcal{X}$, which in this case is $\mathbb{R}$. Since x_α and x_0 are both chosen over the same support, then it must be the case that for some $\alpha = \alpha'$ the cutoffs chosen are equivalent, $x_\alpha = x_0$. In our example above the decision stump corresponds to a hypothesis test in which $\alpha \approx 0.273$, meaning the hypothesis test $g(x)$, and the decision stump, $h(x)$, are equivalent conditional upon $\alpha = \alpha'$ (Neyman and Pearson, 1933). More plainly, a hypothesis test such as the Augmented Dickey Fuller test is equivalent to a weak learner.

Before we discuss how the Augmented Dickey-Fuller test – and indeed any of the unit root tests – can benefit from modern machine learning techniques, we would like to take a moment and show not only an equivalence between the test statistic and decision stump at $\alpha = \alpha'$, but also that, for any choice of α, a boosted statistic can recover the full size-power trade off of a hypothesis test.

Suppose one would like to apply a supervised learning algorithm to a hypothesis testing problem, but restrict the long-run Type I error rate to 5%. This task can be accomplished through boosting algorithms (Schapire and Freund, 2013). Boosting extends our single decision stump by combining two or more stumps to improve classification of the phenomenon of interest – giving the technique the latitude to map the hypothesis space. Let us consider the case of Adaptive Boosting (AdaBoost), a supervised learning algorithm where the learning takes place through an iterative process. In a nutshell, the Adaboost algorithm trains a weak learner (almost always a decision stump), re-weighting the sample based on the mistakes of that learner, then training another stump on a re-weighted sample. [8] This has been shown to be surprisingly accurate and relatively robust to overfitting, see Friedman et al. (2001),Schapire and Freund (2013), and Kuhn et al. (2013) for a more robust discussion regarding the diversity in algorithms, strengths, weaknesses, and implementation.[9]

In Figure 2 we present how boosting converges towards the performance of the ADF statistic with increased number of iterations. The first plot depicts the performance of a single decision stump, which neatly intersects the ADF. When AdaBoost is trained with two iterations, the ROC curve intersects the ADF at two points, that is we have recovered two values of α' for which $\alpha = \alpha'$. In each successive figure, we increase the number

[8] Note that because the weak learner used in the AdaBoost algorithm is the decision stump does not prohibit the boosting of multiple features. At each iteration a different feature may provide a better classifying decision stump based on the re-weighted sample.

[9] For our purposes, in R version 4.0.2 – "Taking Off Again" – we rely on the packages "Rweka" and "ada" to implement our decision stumps and boosting algorithms respectively.

of iterations the boosting algorithm is allowed to evaluate. By 50 iterations, the ROC curve for the boosted test statistic has converged to that of the null distribution. It should be clear that any weighted average of cutoffs produced by the boosting algorithm will produce some level of $\alpha' = \alpha$ and thus a boosted test statistic is equivalent to a standard hypothesis test for any choice of long-run Type I errors that may be made. Thus, if we would like to restrict our Type I errors to $\alpha = 0.05$ we can then simply choose a probability threshold such that Specificity over the validation set is $1 - \alpha = 0.95$.

[Figure 2 about here.]

It is important to note that any model or test – no matter how simple or sophisticated – is only as good as the data on which it is trained and validated. For an algorithm to retain its predictive qualities "in the wild", the training data must reflect the conditions that will be encountered when applied in real world use. In many cases the qualities of a phenomenon are overly expansive and impossible to capture in a neat empirical definition, thus the data and – in turn the test – run the risk of biasing from the true class distribution. However, if the definition of a phenomenon is well-specified, one can simulate from the definition (*i.e.* the DGP) so an algorithm can have ample opportunity to thoroughly inspect and distinguish between the null and the alternative. Unit roots fit into this latter case; we can be certain that our training set accurately reflects the hypothesis test under consideration. Increasing the size of this training set will only improve the accuracy of our threshold choice.

3 A Composite Test for Unit Roots Using Gradient Boosting

In this section, we lay out the blueprint of a composite test and illustrate the marked performance improvements made possible through this novel testing strategy. We do this in four steps: first, we outline the simulated training environment which mirrors that used in the literature. Second, we outline the features we use in the training of our composite test, including nine test statistics developed in the unit root literature. Third, we provide an overview of the general Gradient Boosting Algorithm we employ as a mapping function. Finally, we outline the performance of this novel approach relative to the more traditional statistical tests.

3.1 Simulating a Robust Training Environment

Simulation studies have become rather ubiquitous in the unit root literature, in part because the null distribution for many of the tests has no analytic form. As a result, critical values are often obtained through simulation (see MacKinnon (1996, 2010) for example) and are thus dependent on the assumptions applied to the simulation environments under which they were constructed. In order to assess the performance of current unit root tests, we simulate a large number of time-series, apply tests to these simulations, then compare their ability to detect unit roots.

To that end we will generate $M = 500,000$ series, of which $350,000$ and $75,000$ are reserved for training and validating ML algorithms, respectively. To begin let $\pi_u \sim U(0,1)$ such that,

$$U = \begin{cases} +1: \ \pi_u \geq 0.50 \\ -1: \ \pi_u < 0.50 \end{cases}, \tag{5}$$

where $U = +1$ denotes a series with an unit root and $U = -1$ a near unit root.

Conditional upon the series containing a unit root we generate data from Equation 1 where $\rho = 1$. If $U = -1$ then we draw ρ uniformly over the interval $(0.9000, 0.9999)$.[10] For all simulated series we draw the error from a Gaussian white noise process such that $\epsilon_t \sim N(0,1)$.[11] For our purposes we assume all series are monthly though when developing our input features we estimate the frequency of the series based on the spectral density. Each series is variable in length with the number of years drawn uniformly over the interval $(5, 50)$, which implies that the smallest data set is 60 observed periods and the largest is 600.[12] The inclusion of deterministic elements is determined via a random draw as well. Let π_d be a draw from $U(0,1)$ such that,

$$d_t = \begin{cases} (0,0)': \ \pi_d \leq 0.34 \\ (1,0)': \ 0.34 <= \pi_d < 0.68 \\ (1,t)': \ 0.68 <= \pi_d < 1.00 \end{cases}, \tag{6}$$

where $\beta = (1, 0.005)$ for all simulations.[13]

Finally, we have included a fourth potential generating process such that,

$$y_t = (1-\rho)\gamma + \rho y_{t-1} + \epsilon_t, \tag{7}$$

[10] We have examined other distributions which govern the draw of ρ in the case of a near unit root including $U(0.00, .99)$, $B(\alpha = 8, \beta = 1)$, and $B(\alpha = 2, \beta = 2)$. We have also varied relative weights given to unit roots and near unit roots by change the cutoff for ϕ_u. All additional results and code will be available on our website.

[11] We have also varied the distribution of the error term by using $U(-1, 1)$ with similar results.

[12] This is a practical limitation based on our input features. We have omitted some features to avoid this limitation which allowed the use of shorter series, as small as $T = 10$ with similar qualitative results.

[13] Again, we have varied these parameters and found similar qualitative results even on cross-testing (e.g., using an out-of-sample test set created with $\beta = (1, 0.005)$ against a model trained on different values).

where ρ and ϵ_t are derived in the aforementioned way. Here we assume $\gamma = 1$ and thus any series in the near unit root interface has a drift term which is close to one. While the results contained herein are based on a training set derived uniformly over these four data generating processes we have also examined other processes such as those with a moving average term and data generated directly from the regressions outlined in the Dickey Fuller test paradigm. Each of these exercises produced similar results qualitatively but are omitted for the sake of brevity.

3.2 Input Features

The input features are the drivers of predictive accuracy in any modeling problem, and are dependent on how much unique information each input feature offers. Previous research has shown that there is significant differences in the power of these test statistics. That necessarily means for some series, when the null is false, tests will disagree about the whether to reject, or fail-to-reject, the null. From another perspective, the disagreement is an opportunity to reconcile seemingly uncorrelated information that can make a test more robust. The question is *how much unique information does each input feature provide a composite ML test?*

By drawing on our bank of simulated time-series we can examine the quality of information through two-way comparisons between test statistics, something we illustrate in Figure 3. Perhaps what is most striking about these bivariate comparisons is that none exhibit a continuous, linear relationship. For example, the ERS-d and the ADF exhibit non-linear and disjoint relationships that could not be captured through linear decision boundaries (*e.g.,* those produced by a logit model). The complexity of these patterns is unsurprising as the hypothesis space combines multiple different unit root processes, requiring one to diagnose each time-series for its "true" unit root process in order to identify the appropriate critical value. As we mentioned previously, this necessitates a testing strategy for unit roots and misspecified deterministic elements in the process – a subjective decision open to reasonable discussion – can add to the lack of agreement between tests.

[Figure 3 about here.]

We can further investigate these relationships between test statistics by calculating their mutual information. While a Pearson's correlation could quantify the strength of their *linear* relationship, we can evaluate the uniqueness of each test's information through *mutual information*, which is based in information theory. By calculating how much each test encodes information as *bits* we can estimate mutual information, given as:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)}. \tag{8}$$

Mutual Information establishes how much information – linear or non-linear – is shared between two random variables, which in turn gives a sense of how unique the signal is among our input features. Further, one can

look at the *Information Quality ratio* defined as,

$$\text{IQR}(X,Y) = \mathbb{E}\big[I(X;Y)\big] = \frac{I(X;Y)}{H(X,Y)}, \tag{9}$$

where $H(X,Y)$ is the joint entropy of the two random variables. The Information Quality Ratio measures the amount of information in X based on Y against complete uncertainty (Wijaya et al., 2017). In Table 2 we present the Information Quality Ratio for each pair of tests where a value of 1.0 indicates observing random variable X provides perfect information about random variable Y, or more plainly; there is no unique information in Y relative to X. A value of 0.0 indicates that observing X provides no information about Y, each random variable is unique in its information content. In each pairwise comparison, at least 70% of the information is unique.

[Table 2 about here.]

Given what we know about the interface between near unit roots and unit roots with respect to the test statistics, we can imagine that mapping the hypothesis space may require more information than is available in current tests. Because ML techniques can evaluate and integrate an arbitrarily large number of input features, we believe that saturating the model with additional input features can more richly represent the contours of the unit root surface. While traditional statistical and econometric methods run the risk of unstable results when over-parameterized, ML techniques are engineered to mitigate the effects of overfitting and capture interaction amongst input features. As outlined in Table 3, we include a set of meta-characteristics calculated from each time series as outlined by Wang et al. (2006) and Wang et al. (2009) . Moreover we include the series length and variance ratio between the first difference and level data.[14]

[Table 3 about here.]

When the current generation tests are applied to real world problems, we may assume that any pair of tests can arrive at seemingly contradictory verdicts about the presence of a unit root. By drawing upon a full spectrum of unit root statistics and features, as many ML-based approaches are effective in handling large feature sets and non-linearities, we believe that even a *standard* set of algorithms can effectively map the contours of the NUR-UR interface and reconcile seemingly divergent information about unit root cases.

3.3 Mapping Function: Gradient Boosting Machine

While the center of gravity in the field of machine learning has shifted to deep learning, we believe that traditional machine learning can be employed as a reliable, no frills mapping function in the unit root context.

[14]The variance ratio is a heuristic whereby evidence of non-stationarity in the level series is present if the ratio between $\text{var}(\Delta y)/\text{var}(y)$ is less than one half.

We thus focus on gradient boosting (Friedman, 2001; Chen and Guestrin, 2016), which is a tree-based ensemble learner which is well-suited for structured data problems and can accommodate an arbitrarily large number of variable interactions to map non-linear and disjoint hypothesis spaces.

Gradient boosting machines iteratively grow decision trees to *boost* overall model accuracy. Each iteration is seen as an opportunity to target, and correct, for the residuals of previous iterations. All trees are grown to the same, pre-specified terminal depth, but can also be de-correlated from other trees via bootstrap and random feature sampling. In general, the algorithm continues to grow additional trees until either a pre-determined number have been formed or until no additional improvement is realized. However, the number of trees is a parameter that is balanced with the learning parameter, η, which is a shrinkage parameter that is designed to inhibit overfitting by reducing the contribution of each additional tree. The final prediction is a weighted combination of these iteratively built trees, weighted by the learning parameter.

One practical quality of gradient boosting algorithms is the number of transparent, well-vetted open source implementations. For our purposes we rely on Extreme Gradient Boosting (or XGBoost) – an open source framework for scalable gradient boosting. In Algorithm 1 we have outlined the underlying logic of a gradient boosting machine (Friedman, 2001).[15]

Algorithm 1 Gradient Boosting Machine for Classification

Input: Data, $\mathcal{D} = (X, Y)$, and a differentiable loss function, $L(y{-}i, F(x))$.
Initialize model with a $F_0(x) = \underset{\gamma}{\text{argmin}} \sum_m^M L(y_m, \gamma)$
while $b \leq B$ **do**
    Compute $r_{mb} = -\left[\frac{\delta L(y_i, F(x_i))}{\delta F(x_i)}\right]_{f(x)=F_{b-1}(x)}$ for $m = 1, \ldots, M$
    Fit a tree to the r_{mb} values and create terminal regions R_{jb} for $j = 1, \ldots, J_b$.
    For $j = 1, \ldots, J_b$, compute $\gamma_{jb} = \underset{\gamma}{\text{argmin}} \sum_{x_m \in R_{mj}} L(y_i, F_{b-1}(x_m) + \gamma)$
    Update $F_b(x) = F_{b-1}(x) + \nu \sum_{j=1}^{J_b} \gamma_b I(x \in R_{jb})$.
end while
Output $F_B(x)$.

To calibrate the mapping function and identify the optimal set of hyperparameters we use a grid search over the parameters listed in Table 4. We conducted this grid search following a five-fold cross-validation design using the $M = 350,000$ series training set. The parameter space spans a total of forty combinations, requiring a total of 240 model runs. The optimal hyperparameters have been bolded in Table 4. Upon identifying the optimal parameters we train a "final" version of each algorithm ont he full training set, which can then be applied to any new time series to obtain a predicted probability of containing a unit root.

[15]The primary difference between the algorithm presented here and XGBoost specifically is adjustments made for speed and scalability. We encourage readers to see both Friedman (2001) and Chen and Guestrin (2016) for more information.

[Table 4 about here.]

3.4 Performance of the Composite Unit Root Test

One feature of this ML-based test is that it can easily be applied to any scenario without modification. That is, unlike existing test statistics there is no *a priori* choice of deterministic elements. The mapping function has already learned how to differentiate the processes based on unique combinations of test statistics and time-series features provided during the training phase. Since our primary interest is in understanding the Type I and Type II error rates, and not the alternative error paths that can be opened by misspecification, we assume that for traditional tests the true deterministic elements are known and correctly specified by the practitioner. This means that, in this simulated environment, we are computing the lower-bound of performance improvement for the ML-based test. In practice the accuracy of the traditional test statistics would be lower due to the additional error paths not present in the proposed method.

Additionally, since we are often interested in choosing some fixed Type I error rate through α, we not only report the optimal accuracy classification, but also alternatives based on a threshold which approximates the standard choices of α. These thresholds were calculated in the validation step by examining the Receiver Operating Characteristic Curve and choosing a specificity corresponding to the desired α. This choice provides the appropriate decision threshold. Alternatively, one could choose a threshold based on a weight of ratio which outlines the cost of Type II relative to Type I errors. In some fields where the cost function around the error types is more explicit this may be preferable for determining the appropriate threshold.[16]

[Table 5 about here.]

In either scenario, the threshold is applied to the scored out-of-sample test set in order to evaluate its accuracy. As seen in Table 5, the out-of-sample performance gains from ML-based tests are approximately five percentage points greater than the next highest alternative (ERSp). By exploiting the variation between tests and ancillary information, the *pseudo*-composite test increases accuracy primarily through an increase in sensitivity (empirical power).

In contrast the average empirical size of the traditional unit root tests in this simulated environment is approximately 6.14% using the standard 5% critical values. The most comparable result from the proposed ML-based composite test comes from using a threshold corresponding to $\alpha = 0.05$ which translates empirically to a size of 5.6%. In this case the power gains are more modest but still out perform the next best alternative by approximately four percentage points. Note that these results are over a variety of sample sizes, deterministic

[16] For example, one could identify $c(\boldsymbol{e}) = c(e_2)/c(e_1)$ where $c(e_2)$ and $c(e_1)$ are the costs of Type II and Type I errors respectively. While these may not be known explicitly their ratio, $c(\boldsymbol{e})$, maybe known and thus used to determine the appropriate threshold.

elements included in the generating process (trend or drift), as well as values of ρ. Additionally, as mentioned previously, for traditional tests such as the ADF these results include no possible model misspecification as it is assumed the practitioner knows perfectly which deterministic elements to include.

[Figure 4 about here.]

In Figure 4 we have plotted more traditional power curves over the associated values of ϕ.[17] In Figure 4a we have plotted the corresponding power curves under an assumption of perfect knowledge regarding the deterministic trends in the process. Within the interval $[0.975, 1.00)$ the proposed ML test produces identical power to that of the DF-GLS test proposed by Elliot et al. (1996) and slightly greater power than other alternatives. As one moves farther away from the null, $\phi = 1$, the proposed ML test is more powerful than any of the alternatives. In Figure 4b we assume complete ignorance on the deterministic trends and choose at random which to include in the test. Our proposed ML test is the same since we teach the algorithm to recognize the presence of these elements and thus no choice is necessary on the part of a practitioner. However, the traditional tests see a decrease in power due to model misspecification, matching comments in Hamilton (1994) and Kennedy (2008). Neither case, full knowledge or random choice, is representative of practical application however we can think of these as bounds on power conditional upon the included deterministic elements. Again, note that these are finite sample power curves where the length of the series is contained within the interval $(60, 600)$, values which seem feasible for most practical applications. Our power curves say nothing about the asymptotic power of the tests in question, including the proposed, machine-learning based test.

4 Empirical Examples

This section is presented in two subsections. The first examines a well-known collection of macroeconomic indicators from the United States. First examined in Nelson and Plosser (1982), these indicators have been repeatedly revisited by subsequent works. While this data is quite old and has for all purposes lost its practical applications, it still serves as a useful benchmark in the literature. The second example is a brief look at state level unemployment hysteresis in the United States using data from the Local Area Unemployment Statistics program governed by the U.S. Bureau of Labor Statistics.

[17] Note that our test sample has a variety of data generating processes and sample sizes. As a result these are average power curves over the simulated test environment rather than a power curve specific to a single sample size and data generating process.

4.1 Nelson & Plosser Data: A Benchmark Data Set

Following the seminal works of Dickey and Fuller (1979) and Dickey and Fuller (1981) a conversation began about the presence of unit roots in macroeconomic indicators. Nelson and Plosser (1982) examined fourteen such indicators for the presence of a unit root; these included four measures of Gross National Product (real, nominal, per capita, and the deflator), both national employment and unemployment, real and nominal wages, money stock and velocity, bond yields, stock prices, and indices of industrial production and consumer prices. These series were annual in nature and ranged from 62 to 111 years in length. Table 6 provides the summary statistics for the data set as put forth by the authors. This data set, though quite old in relative terms, has become the benchmarking data set for unit root test development and was revisited in Perron (1989); Stock (1991); Kwiatkowski et al. (1992); Andrews and Chen (1994); Zivot and Andrews (2002), and Charles and Darné (2012) among others.

[Table 6 about here.]

In Figure 5a we have plotted the resulting probability of a unit root for each of the fourteen series examined in Nelson and Plosser (1982) and subsequent works. Following other authors, we examined the log of these series for the presence of a unit root using our composite testing mechanism. For each series we calculated the full set of features used for training in our mapping function. The dashed lines represent decision thresholds corresponding to $\alpha \in (0.10, 0.05, 0.01)$. For any series to the right of a threshold one would fail-to-reject the null of a unit root, and those to the left of any threshold would reject the null. As the level of desired Type I errors decrease one needs less evidence to claim the series is a unit root. Note what is changing is the threshold choice, not the probability of unit root, this remains static based on the pre-trained model and these values are constant even if we rerun the prediction.

[Figure 5 about here.]

Using $\alpha = 0.10$ requires the least amount of evidence to reject the null; when we choose this threshold we see that four series – Common Stock Prices, Money Velocity, Money Stock, and Industrial Production Index – all fall under fail-to-reject status and thus contain a unit root. Any $\alpha < 0.10$ necessarily means that more evidence is needed to reject the null and thus these four series will always fail-to-reject the null. Conversely, an α of 0.01 choice will require more evidence to reject the null and we see that only Real per capita GNP continues to reject the null at this level.

In Figure 5b we have contextualized our results at all three levels of α with those of prior literature. Broadly speaking there seems to be consensus that, at a choice of $\alpha = 0.05$, three of the fourteen indicators contain a unit root – Velocity, Bond Yields, and Consumer Prices. Using our proposed method (with $\alpha = 0.05$) we see that Consumer Prices switches from failure-to-reject the null to reject. This is consistent with our method having more power as shown in the simulation study and, on the margin, rejecting the null of a

unit root more often when that null is false.[18] For this particular choice of α our results most closely align with those of Perron (1989) with differences in Industrial Production (reject to fail-to-reject), Consumer Prices (fail-to-reject to reject), Money Stock (reject to fail-to-reject), and Common Stock Prices (reject to fail-to-reject).

4.2 Unemployment Hysteresis in the United States: A Brief Look

Starting with Blanchard and Summers (1986) there has been an interest in the unemployment hysteresis hypothesis; defined as the strong dependence of current unemployment on past values. This is often examined using tests for a unit root, see Song and Yangru (1997); Papell et al. (2000); León-Ledesma (2002); Camarero et al. (2006); Romero-Avila and Usabiaga (2007); Khraief et al. (2020); Yaya et al. (2021) for example.

Following Song and Yangru (1997) we will examine the hysteresis hypothesis in the United States using our composite unit root test at the state level. We obtain state level unemployment data from the U.S. Bureau of Labor Statistics Local Area Unemployment Statistics and following previous research we examine the natural log of this indicator. This monthly, not seasonally adjusted data covers a time period from January 1976 through March of 2019, and while it covers all fifty states plus the District of Columbia we will limit our focus only to the conterminous United States. Furthermore, we will divide our analysis into four distinct time periods: the full sample from 1976-2019, a period from 1992-2003 which includes the dot-com recession from March 2001 to November 2001, a period from 2002-2019 which includes the Great Recession from December 2007 to June of 2009, and finally a period post Great Recession from July 2010 to March 2019 which represents a period of slow but steady growth. For comparison we will use the DF-GLS test outlined by Elliot et al. (1996) which is, according to our simulation results, the closest in power. For now we will omit the use of panel unit root tests as it falls outside the scope of our analysis in this particular work. Many of the panel unit root tests do strongly reject the null of a unit root however, see Song and Yangru (1997). It is important to note that, while a panel unit root test may reject the null for all a collection of states (in most cases the entire continental United States), this may obscure an individual state's status which may be relevant to both state and local policy makers more so than an examination of all states.

[Figure 6 about here.]

In Figure 6 we have plotted the probability of each state level unemployment rate time series over the entire sample containing a unit root. The states that are least likely to reject the null of unit root based on the results of our *pseudo*-composite test are New Hampshire and Connecticut. However, it is important to note two things; first, none of the probabilities are greater than 0.50 and an accuracy maximizing threshold would

[18] For further context, the ADF test rejects the null at $\alpha = 0.10$, assuming a trend is included, while the DF-GLS test rejects at $\alpha = 0.05$.

reject the null for all states. Second, the choice of long-run Type I error rates is very important as a shift from $\alpha = 0.05$ to $\alpha = 0.01$ results in an additional eighteen states which would fail-to-reject the null. By comparison, over the full sample using the DF-GLS test, we find that forty-three of the forty-nine states fail-to-reject the null hypothesis under $\alpha = 0.05$, nearly the opposite finding which is consistent with previous literature. For example, in Song and Yangru (1997), which used the ADF, Phillips-Perron, and Zivot-Andrews tests, it was shown that no more than five of the states reject the null over the examined time period. Finally, we should note that if one is concerned with maximizing accuracy without greater weight being placed on the different error types then the *pseudo*-composite test outlined herein would reject the null for all states, a finding consistent with many panel unit root tests.

[Figure 7 about here.]

In Figure 7 we have plotted a map showing the individual states for which we reject the null at various levels of α. Using the DF-GLS test there is no change between a choice of $\alpha = 0.05$ and $\alpha = 0.01$. A total of six states reject the null at both significance levels with five of those states clustered in the upper mid-west. Using the proposed composite test we see that the choice of α is much more salient. At $\alpha = 0.05$ only two states (New Hampshire and Connecticut) fail-to-reject the null while an additional eighteen gain that status when $\alpha = 0.01$ is chosen. Many of these new states are in the north east (the lone rejection being Vermont) and south. At the accuracy optimizing choice of α all states would reject the null of a unit root and thus reject hysteresis under the proposed test.

[Table 7 about here.]

In Table 7 we have provided the results for all four time spans for both the proposed test and the DF-GLS test. Shorter time periods, as expected due to the limited sample size, tend to reject the null for fewer states across both testing regimes. However, even in the shortest time horizon the proposed test rejects the null for more than half of the states. These results are consistent with the power curves shown earlier which showed a clear advantage in power over the near unit root domain. Additionally, recall that a feature in the composite test we have proposed is the DF-GLS test itself and that additional tests can be added to the feature set if needed. Overall these results point to state level unemployment being a stationary process in the long run, but the hysteresis hypothesis being true over shorter time horizons. One possible mechanism for such a result is that migratory frictions are more salient over the short or medium term and thus the persistence is felt much more than over the longer time horizons where perhaps these frictions are less relevant.

Finally, we would like to point out that while there is evidence to suggest that panel unit root tests strongly reject the null of a unit root (Song and Yangru, 1997; Romero-Avila and Usabiaga, 2007), there is value in examining this at an individual level. States often face the effects of both national shocks to the labor market, where Federal policy may have broader impact, and localized effects which may be addressed more efficiently by state or local level policy. Identifying persistent unemployment at a more granular level than jointly over

all states may assist these local policy makers in identifying time spans of persistent unemployment and act accordingly.[19]

5 Conclusion

Testing for a unit root is one of the foundational pieces in all of time-series econometrics. Its importance is made apparent by the thousands of hours of human capital which have been spent developing and refining test statistics under a variety of different contexts. In this paper we have proposed a *pseudo*-composite test for unit roots which leverages modern computational power and classification algorithms to exploit variation between existing tests and other pertinent information. The proposed testing method is more powerful at differentiating unit roots from near unit roots, an area in which current tests are known to struggle. Moreover, by leveraging the train-validation framework common to the Machine Learning literature we are able to retain control over long-run Type I error rates by allowing users to specify a desired value of α.

Using a simulated environment which is common to the unit root literature we find that our proposed ensemble based method is approximately four percentage points more accurate than the next best alternative controlling for size. Accuracy can be improved at the cost of a fixed size such that the overall improvement is nearly seven percentage points. Note that this is a pessimistic view on the improvements as our results are conditional upon perfect knowledge of which deterministic elements to include in the testing procedure. Any misspecification on the part of a practitioner would only increase the accuracy gains. Since the proposed method removes the necessity of element choice from the equation we see it as a strict improvement over any single test statistic. Moreover, as progress is made in the testing of unit roots via new test statistics, or improvements to existing tests, these can be easily incorporated into our proposed framework. Additionally, we provide an R package which contains a pre-trained algorithm using our simulated environment so that practitioners can test any individual or group of time series in the same manner in which they conduct current tests.[20] Finally, this package allows for custom training sets to be created by a practitioner concerned about moving average or autoregressive conditional heteroskedastic data generation.

To illustrate this process we revisited the fourteen macroeconomic time series found in Nelson and Plosser (1982), a commonly used benchmarking data set in the unit root literature. We find that our results are broadly consistent with those found in the literature with only four indicators – Industrial Production, Money Stock, Money Velocity, and Common Stock Prices – failing to reject the null at all standard levels of

[19]The LAUS has data at a county level for example and the proposed test can be easily extended to this finer geographic structure.

[20]The package can be found at `https://github.com/DataScienceForPublicPolicy/unitrootML`.

significance. In an effort to contextualize our results in a more current data set we examine the hypothesis of unemployment hysteresis and find that our results are broadly consistent with panel unit root tests rather than univariate tests. We find that the choice of α is more salient for the proposed test than the DF-GLS test which has implications for state and local policy makers.

As we have demonstrated with the unit roots case, the unit root hypothesis space's underlying multivariate distribution is not well-behaved in practice and the abilities of current hypothesis tests are quite variable. Thus, the ability for ML-based hypothesis tests to reconcile conflicting diagnostics and realize large performance gains has far-reaching implications – the quality of inferences can be vastly improved and the disagreement among tests are resolved. Much like replicating a sample of DNA for further study, ML-based testing can be extended to any phenomenon that is well-defined and can be simulated (*e.g.* equality of distributions, normality).

A State Probabilities

[Figure 8 about here.]

[Figure 9 about here.]

[Figure 10 about here.]

B State Level Hysteresis Maps

[Figure 11 about here.]

[Figure 12 about here.]

[Figure 13 about here.]

C Individual Data Generating Processes: Main Results

[Table 8 about here.]

[Table 9 about here.]

[Table 10 about here.]

[Table 11 about here.]

D Alternative Data Generating Processes

Let y_t be an autoregressive time-series generated such that one of the following is true,

$$y_t = \gamma + \phi y_{t-1} + \delta t + \epsilon_t, \tag{10}$$

$$y_t = \gamma + \phi y_{t-1} + \epsilon_t, \tag{11}$$

$$y_t = \phi y_{t-1} + \epsilon_t, \tag{12}$$

indexed by $t = 1, \ldots, T$. In Equation 10 we have included a drift term, γ, as well as a time trend, δt, and in all cases $\epsilon_t \sim WN$. Equations 11 and 12 omit the time trend and both the time trend and drift term respectively. These data generating processes where used in previous versions of the paper. The following tables outline results of these processes. The structure of the simulated environment is identical in terms of the randomized elements (e.g., number of unit root versus near unit roots, values for ϕ, etc).

[Table 12 about here.]

[Table 13 about here.]

[Table 14 about here.]

References

Andrews, D. W. and H.-Y. Chen (1994). Approximately median-unbiased estimation of autoregressive models. *Journal of Business & Economic Statistics 12*(2), 187–204.

Blanchard, O. J. and L. H. Summers (1986). Hysteresis and the European unemployment problem. *NBER Macroeconomics Annual 1*, 15–78.

Breitung, J. (2002). Nonparametric tests for unit roots and cointegration. *Journal of Econometrics 108*(2), 343–363.

Breitung, J. and R. Taylor (2003). Corrigendum to" nonparametric tests for unit roots and cointegration"[j. econom. 108 (2002) 343-363]. *Journal of Econometrics 117*(2), 401–404.

Camarero, M., J. L. Carrion-i Silvestre, and C. Tamarit (2006). Testing for hysteresis in unemployment in OECD countries: new evidence using stationarity panel tests with breaks. *Oxford Bulletin of Economics and Statistics 68*(2), 167–182.

Charles, A. and O. Darné (2012). Trends and random walks in macroeconomic time series: A reappraisal. *Journal of Macroeconomics 34*(1), 167–180.

Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Dickey, D. A. and W. A. Fuller (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association 74*(366a), 427–431.

Dickey, D. A. and W. A. Fuller (1981). Likelihood ratio statistics for autoregressive time series with a unit root. *Econometrica: Journal of the Econometric Society*, 1057–1072.

Elder, J. and P. E. Kennedy (2001). Testing for unit roots: what should students be taught? *The Journal of Economic Education 32*(2), 137–146.

Elliot, B., T. Rothenberg, and J. Stock (1996). Efficient tests of the unit root hypothesis. *Econometrica 64*(8), 13–36.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The Elements of Statistical Learning*, Volume 1. Springer series in statistics New York.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 1189–1232.

Granger, C. W. and P. Newbold (1974). Spurious regressions in econometrics. *Baltagi, Badi H. A Companion of Theoretical Econometrics*, 557–61.

Hamilton, J. D. (1994, January). *Time Series Analysis* (1 ed.). Princeton University Press.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Jansson, M. (2008). Semiparametric power envelopes for tests of the unit root hypothesis. *Econometrica 76*(5), 1103–1142.

Jansson, M. and M. Ø. Nielsen (2012). Nearly efficient likelihood ratio tests of the unit root hypothesis. *Econometrica 80*(5), 2321–2332.

Kennedy, P. (2008). *A Guide to Econometrics*. John Wiley & Sons.

Khraief, N., M. Shahbaz, A. Heshmati, and M. Azam (2020). Are unemployment rates in OECD countries stationary? evidence from univariate and panel unit root tests. *The North American Journal of Economics and Finance 51*, 100838.

Kuhn, M., K. Johnson, et al. (2013). *Applied Predictive Modeling*, Volume 26. Springer.

Kwiatkowski, D., P. C. Phillips, P. Schmidt, Y. Shin, et al. (1992). Testing the null hypothesis of stationarity against the alternative of a unit root. *Journal of Econometrics 54*(1-3), 159–178.

León-Ledesma, M. A. (2002). Unemployment hysteresis in the US states and the EU: a panel approach. *Bulletin of Economic Research 54*(2), 95–103.

MacKinnon, J. G. (1996). Numerical distribution functions for unit root and cointegration tests. *Journal of Applied Econometrics 11*(6), 601–618.

MacKinnon, J. G. (2010). Critical values for cointegration tests. Technical report, Queen's Economics Department Working Paper.

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure 405*(2), 442–451.

Nelson, C. R. and C. R. Plosser (1982). Trends and random walks in macroeconomic time series: some evidence and implications. *Journal of Monetary Economics 10*(2), 139–162.

Neyman, J. and E. S. Pearson (1933). The testing of statistical hypotheses in relation to probabilities a priori. In *Mathematical Proceedings of the Cambridge Philosophical Society*, Volume 29, pp. 492–510. Cambridge University Press.

Ng, S. and P. Perron (2001). Lag length selection and the construction of unit root tests with good size and power. *Econometrica 69*(6), 1519–1554.

Oliver, J. J. and D. Hand (1994). Averaging over decision stumps. In *European Conference on Machine Learning*, pp. 231–241. Springer.

Pantula, S. G., G. Gonzalez-Farias, and W. A. Fuller (1994). A comparison of unit-root test criteria. *Journal of Business & Economic Statistics 12*(4), 449–459.

Papell, D. H., C. J. Murray, and H. Ghiblawi (2000). The structure of unemployment. *Review of Economics and Statistics 82*(2), 309–315.

Perron, P. (1989). The great crash, the oil price shock, and the unit root hypothesis. *Econometrica: Journal of the Econometric Society*, 1361–1401.

Perron, P. and S. Ng (1996). Useful modifications to some unit root tests with dependent errors and their local asymptotic properties. *The Review of Economic Studies 63*(3), 435–463.

Phillips, P. C. and P. Perron (1988). Testing for a unit root in time series regression. *Biometrika 75*(2), 335–346.

Romero-Avila, D. and C. Usabiaga (2007). Unit root tests, persistence, and the unemployment rate of the us states. *Southern Economic Journal*, 698–716.

Said, S. E. and D. A. Dickey (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika 71*(3), 599–607.

Schapire, R. E. and Y. Freund (2013). Boosting: Foundations and algorithms. *Kybernetes*.

Schmidt, P. and P. C. Phillips (1992). Lm tests for a unit root in the presence of deterministic trends. *Oxford Bulletin of Economics and Statistics 54*(3), 257–287.

Song, F. M. and W. Yangru (1997). Hysteresis in unemployment evidence from 48 us states. *Economic Inquiry 35*(2), 235–243.

Stock, J. H. (1991). Confidence intervals for the largest autoregressive root in us macroeconomic time series. *Journal of Monetary Economics 28*(3), 435–459.

Tharwat, A. (2018). Classification assessment methods. *Applied Computing and Informatics*.

Tong, X., Y. Feng, and A. Zhao (2016). A survey on neyman-pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics 8*(2), 64–81.

Wang, X., K. Smith, and R. Hyndman (2006). Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery 13*(3), 335–364.

Wang, X., K. Smith-Miles, and R. Hyndman (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing 72*(10-12), 2581–2594.

Wijaya, D. R., R. Sarno, and E. Zulaika (2017). Information quality ratio as a novel metric for mother wavelet selection. *Chemometrics and Intelligent Laboratory Systems 160*, 59–71.

Yaya, O. S., A. E. Ogbonna, F. Furuoka, and L. A. Gil-Alana (2021). A new unit root test for unemployment hysteresis based on the autoregressive neural network. *Oxford Bulletin of Economics and Statistics*.

Zivot, E. and D. W. K. Andrews (2002). Further evidence on the great crash, the oil-price shock, and the unit-root hypothesis. *Journal of Business & Economic Statistics 20*(1), 25–44.

Table 1: Drawing parallels between hypothesis tests and classification models

Qualities	Hypothesis Tests	Classification Problem
Outcomes	Complementary hypotheses	Binary target
Mapping Function	Test statistic and null distribution	Classification algorithm
Model Validation	Type I and Type II errors	Type I and Type II errors
Threshold	Rejection Threshold	Decision Threshold

Table 2: Mutual Information of Test Statistics

	ADF	KPSS	PP	PGFF	Breit	ERS-d	ERS-p	URZA	URSP
ADF	1.000								
KPSS	0.158	1.000							
PP	0.084	0.028	1.000						
PGFF	0.142	0.142	0.064	1.000					
Breit	0.218	0.214	0.052	0.131	1.000				
ERS-d	0.273	0.162	0.101	0.144	0.259	1.000			
ERS-p	0.085	0.068	0.084	0.027	0.037	0.064	1.000		
URZA	0.069	0.026	0.206	0.041	0.038	0.073	0.092	1.000	
URSP	0.061	0.024	0.257	0.056	0.044	0.081	0.043	0.096	1.000

Note: In this table we have provided the Information Quality Ratio (IQR) for each two-by-two test comparison. The IQR measures the amount of mutual information, $I(X;Y)$, relative to the joint entropy $H(X,Y)$ and is a representation of total correlation.

Table 3: Features for Classification

UR Tests	Level and First Difference	STL Decomposed Series	Miscellaneous
ADF	Skewness	TNN Test	Length
PP	Kurtosis	Skewness	$\text{var}(\Delta y)/\text{var}(y)$
PGFF	Box Statistic	Kurtosis	
KPSS	Lyapunov Exponent	Box Statistic	
ERS (d & p)	TNN Test		
URSP	Hurst Exponent		
URZA	Strength of Trend		
Breit	Strength of Seasonality		

Note: The statistics calculated on the STL decomposed series are done both on the level and the first difference. Going forward we use Δ to denote a statistic on the first difference and "Decomposed" to denote those on the adjusted series. TNN Test refers to the Teraesvirta Neural Network test. See Wang et al. (2006), Wang et al. (2009), and `https://robjhyndman.com/hyndsight/tscharacteristics/` for more information.

Table 4: Grid Search Parameter Space

Hyperparameters	Gradient Boosting
Eta (Shrinkage Parameter)	$\{0.01, 0.03, \mathbf{0.1}, 0.3, 0.5\}$
Column Sample by Tree (% of Columns)	$\{\mathbf{0.8}.1\}$
Subsampling of Training Instances	$\{\mathbf{0.8}, 1\}$
Max Tree Depth (Number of Levels)	$\{4, \mathbf{6}\}$

Note: As identified through five-fold cross validation, the winning value for each hyperparameter is bolded.

Table 5: Main Results

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
GB|Optimal	0.846	0.827	0.864	0.859	0.834	0.843	0.692
GB|$\alpha = 0.100$	0.842	0.786	0.898	0.885	0.807	0.832	0.688
GB|$\alpha = 0.050$	0.821	0.698	0.944	0.926	0.758	0.796	0.662
GB|$\alpha = 0.010$	0.736	0.483	0.990	0.980	0.657	0.647	0.548
ADF	0.761	0.572	0.948	0.917	0.690	0.705	0.562
PP	0.730	0.518	0.941	0.897	0.662	0.657	0.506
KPSS	0.611	0.270	0.952	0.849	0.566	0.409	0.303
PGFF	0.748	0.525	0.971	0.948	0.672	0.676	0.554
BREIT	0.665	0.379	0.951	0.885	0.605	0.531	0.402
ERSd	0.788	0.632	0.943	0.917	0.720	0.748	0.605
ERSp	0.800	0.661	0.939	0.916	0.735	0.768	0.625
URZA	0.628	0.313	0.942	0.843	0.579	0.456	0.328
URSP	0.716	0.573	0.860	0.803	0.668	0.669	0.452

Note: For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct choice of deterministic elements. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error. All results for existing test statistics are presented assuming the choice $\alpha = 0.05$.

Table 6: Summary Statistics for Nelson and Plosser (1982) Data

Variable	Start	End	T	Min	Max	SD
Real GNP	1909	1970	62	116.80	724.70	180.32
Nominal GNP	1909	1970	62	33,400	974,126	252,334.20
GNP Per Capita	1909	1970	62	1,126	3,577	726.59
Industrial Production	1860	1970	111	0.90	110.70	27.65
Employment	1890	1970	81	21,102	81,815	16,755.98
Unemployment Rate	1890	1970	81	1.20	24.90	5.56
GNP Deflator	1889	1970	82	22.10	135.30	31.38
CPI	1860	1970	111	25.00	116.30	23.41
Nominal Wages	1900	1970	71	487	8,150	2,134.63
Real Wages	1900	1970	71	19.48	70.81	16.46
Money Stock	1889	1970	82	3.60	401.30	102.67
Velocity of Money	1869	1970	102	1.16	5.61	1.14
Bond Yields	1871	1970	100	3.14	98.70	24.05
Stock Prices	1900	1970	71	2.43	7.60	0.96

Table 7: Results Over Different Time Spans

	XG		DF-GLS	
Period	Reject	Fail-to-Reject	Reject	Fail-to-Reject
1976-2019	47	2	6	43
1992-2003	45	4	4	45
2002-2019	43	6	1	48
2010-2019	34	15	1	48

Note: All decisions in this table were made using $\alpha = 0.05$ critical values. In Appendix B we provide maps of the appropriate time spans and indicate which states fail-to-reject versus reject the null.

Table 8: Results for DGP 1

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.848	0.833	0.861	0.856	0.840	0.844	0.695
XG|$\alpha = 0.10$	0.837	0.890	0.786	0.804	0.879	0.845	0.679
XG|$\alpha = 0.05$	0.830	0.723	0.935	0.916	0.774	0.808	0.674
XG|$\alpha = 0.01$	0.749	0.505	0.989	0.978	0.669	0.666	0.565
ADF	0.813	0.678	0.947	0.926	0.749	0.783	0.649
PP	0.743	0.541	0.942	0.902	0.675	0.676	0.528
KPSS	0.634	0.312	0.951	0.863	0.584	0.458	0.343
PGFF	0.771	0.570	0.969	0.948	0.696	0.712	0.589
BREIT	0.681	0.404	0.954	0.896	0.619	0.557	0.429
ERSd	0.815	0.688	0.941	0.919	0.753	0.787	0.650
ERSp	0.822	0.709	0.935	0.914	0.765	0.799	0.661
URZA	0.633	0.318	0.943	0.846	0.584	0.463	0.335
URSP	0.717	0.569	0.862	0.803	0.670	0.666	0.452

Note: The results in this table are for series which originate from Equation 1 where $d_t = (0, 0)'$ such that there is no deterministic elements. For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct choice of deterministic elements. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error. All results for existing test statistics are presented assuming the choice $\alpha = 0.05$.

Table 9: Results for DGP 2

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.851	0.832	0.871	0.866	0.838	0.849	0.703
XG|$\alpha = 0.10$	0.837	0.888	0.786	0.806	0.875	0.845	0.678
XG|$\alpha = 0.05$	0.833	0.723	0.942	0.926	0.772	0.812	0.682
XG|$\alpha = 0.01$	0.746	0.502	0.991	0.982	0.665	0.664	0.565
ADF	0.731	0.510	0.952	0.914	0.660	0.655	0.515
PP	0.744	0.546	0.943	0.905	0.674	0.681	0.532
KPSS	0.626	0.302	0.950	0.858	0.576	0.447	0.331
PGFF	0.771	0.571	0.972	0.953	0.693	0.714	0.592
BREIT	0.676	0.401	0.951	0.891	0.613	0.553	0.421
ERSd	0.812	0.683	0.942	0.922	0.748	0.784	0.647
ERSp	0.820	0.699	0.942	0.923	0.757	0.796	0.660
URZA	0.630	0.322	0.940	0.842	0.580	0.465	0.332
URSP	0.718	0.577	0.860	0.805	0.670	0.672	0.455

Note: The results in this table are for series which originate from Equation 1 where $d_t = (1, 0)'$ such that there is a drift term but no trend. For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct choice of deterministic elements. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error. All results for existing test statistics are presented assuming the choice $\alpha = 0.05$.

Table 10: Results for DGP 3

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
xgoptimal	0.832	0.803	0.861	0.853	0.814	0.827	0.665
xgp10	0.828	0.878	0.779	0.799	0.864	0.837	0.660
xgp05	0.801	0.665	0.938	0.915	0.737	0.770	0.627
xgp01	0.697	0.406	0.989	0.973	0.624	0.573	0.486
ADF	0.678	0.409	0.949	0.889	0.616	0.560	0.425
PP	0.695	0.454	0.937	0.879	0.631	0.599	0.447
KPSS	0.560	0.160	0.960	0.802	0.533	0.267	0.201
PGFF	0.685	0.393	0.977	0.944	0.616	0.555	0.455
BREIT	0.633	0.315	0.952	0.869	0.581	0.463	0.347
ERSd	0.708	0.464	0.953	0.908	0.639	0.614	0.477
ERSp	0.733	0.516	0.950	0.911	0.662	0.659	0.517
URZA	0.619	0.299	0.940	0.833	0.572	0.440	0.311
URSP	0.716	0.574	0.859	0.803	0.668	0.669	0.451

Note: The results in this table are for series which originate from Equation 1 where $d_t = (1, t)'$ such that there is both a drift and trend term. For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct choice of deterministic elements. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error. All results for existing test statistics are presented assuming the choice $\alpha = 0.05$.

Table 11: Results for DGP 4: Equation 7

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.848	0.833	0.861	0.856	0.840	0.844	0.695
XG|$\alpha = 0.10$	0.837	0.890	0.786	0.804	0.879	0.845	0.679
XG|$\alpha = 0.05$	0.830	0.723	0.935	0.916	0.774	0.808	0.674
XG|$\alpha = 0.01$	0.749	0.505	0.989	0.978	0.669	0.666	0.565
ADF	0.813	0.678	0.947	0.926	0.749	0.783	0.649
PP	0.743	0.541	0.942	0.902	0.675	0.676	0.528
KPSS	0.634	0.312	0.951	0.863	0.584	0.458	0.343
PGFF	0.771	0.570	0.969	0.948	0.696	0.712	0.589
BREIT	0.681	0.404	0.954	0.896	0.619	0.557	0.429
ERSd	0.815	0.688	0.941	0.919	0.753	0.787	0.650
ERSp	0.822	0.709	0.935	0.914	0.765	0.799	0.661
URZA	0.633	0.318	0.943	0.846	0.584	0.463	0.335
URSP	0.717	0.569	0.862	0.803	0.670	0.666	0.452

Note: The results in this table are for series which originate from Equation 7. For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct choice of deterministic elements. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error. All results for existing test statistics are presented assuming the choice $\alpha = 0.05$.

Table 12: Results for Alternative DGP 1

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.982	0.964	1.000	1.000	0.965	0.982	0.964
XG|$\alpha = 0.100$	0.980	0.960	1.000	1.000	0.962	0.980	0.961
XG|$\alpha = 0.050$	0.978	0.955	1.000	1.000	0.957	0.977	0.956
XG|$\alpha = 0.010$	0.966	0.933	1.000	1.000	0.937	0.965	0.935
ADF	0.708	0.423	0.992	0.981	0.632	0.591	0.505
PP	0.722	0.453	0.991	0.980	0.644	0.620	0.526
KPSS	0.571	0.159	0.983	0.903	0.539	0.271	0.250
PGFF	0.692	0.385	0.998	0.994	0.619	0.556	0.485
BREIT	0.652	0.309	0.995	0.986	0.590	0.471	0.419
ERSd	0.703	0.410	0.997	0.992	0.628	0.580	0.502
ERSp	0.706	0.421	0.990	0.977	0.631	0.589	0.500
URZA	0.640	0.295	0.985	0.953	0.583	0.450	0.387
URSP	0.763	0.540	0.987	0.976	0.682	0.695	0.589

Note: For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct originating data generating process. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error.

Table 13: Results for Alternative DGP 2

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.992	0.991	0.992	0.992	0.992	0.992	0.984
XG|$\alpha = 0.100$	0.990	0.992	0.987	0.987	0.992	0.990	0.979
XG|$\alpha = 0.050$	0.993	0.989	0.997	0.997	0.990	0.993	0.987
XG|$\alpha = 0.010$	0.986	0.972	1.000	1.000	0.973	0.986	0.973
ADF	0.760	0.522	0.996	0.992	0.678	0.684	0.589
PP	0.772	0.550	0.992	0.985	0.690	0.706	0.605
KPSS	0.647	0.291	1.000	1.000	0.588	0.451	0.413
PGFF	0.773	0.543	1.000	1.000	0.689	0.704	0.612
BREIT	0.694	0.384	1.000	1.000	0.621	0.555	0.488
ERSd	0.763	0.524	1.000	1.000	0.680	0.688	0.597
ERSp	0.777	0.551	1.000	1.000	0.692	0.710	0.617
URZA	0.633	0.315	0.947	0.854	0.583	0.460	0.338
URSP	0.705	0.544	0.865	0.799	0.657	0.648	0.432

Note: For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct originating data generating process. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error.

Table 14: Results for Alternative DGP 3

	ACC	SEN	SPE	PPV	NPV	F^1	MCC
XG|Optimal	0.852	0.852	0.852	0.852	0.852	0.852	0.704
XG|$\alpha = 0.100$	0.839	0.897	0.782	0.804	0.883	0.848	0.683
XG|$\alpha = 0.050$	0.844	0.758	0.930	0.915	0.793	0.829	0.698
XG|$\alpha = 0.010$	0.724	0.455	0.995	0.988	0.645	0.623	0.534
ADF	0.821	0.691	0.951	0.934	0.755	0.794	0.665
PP	0.737	0.532	0.943	0.903	0.668	0.669	0.520
KPSS	0.624	0.301	0.948	0.853	0.575	0.445	0.326
PGFF	0.768	0.568	0.969	0.949	0.691	0.711	0.586
BREIT	0.670	0.391	0.949	0.884	0.609	0.542	0.409
ERSd	0.820	0.700	0.940	0.921	0.758	0.796	0.659
ERSp	0.827	0.718	0.937	0.919	0.768	0.806	0.670
URZA	0.632	0.318	0.946	0.855	0.581	0.464	0.340
URSP	0.714	0.570	0.858	0.801	0.666	0.666	0.447

Note: For each of the alternative tests (*e.g.,* ADF) we have provided the test with the correct originating data generating process. This means that our comparison group is the "best" these tests can do in the environment provided and contains no model specification error.

Figure 1: Hypothesis Tests as Classification Problems

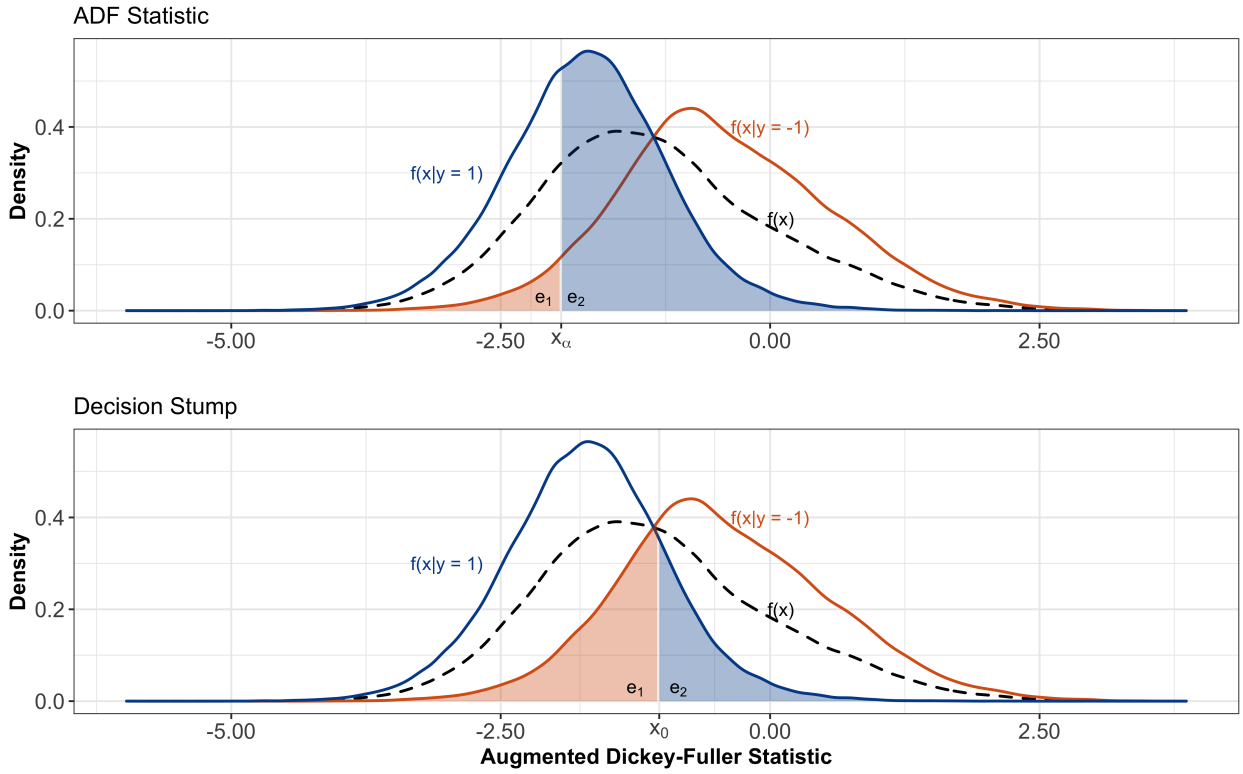
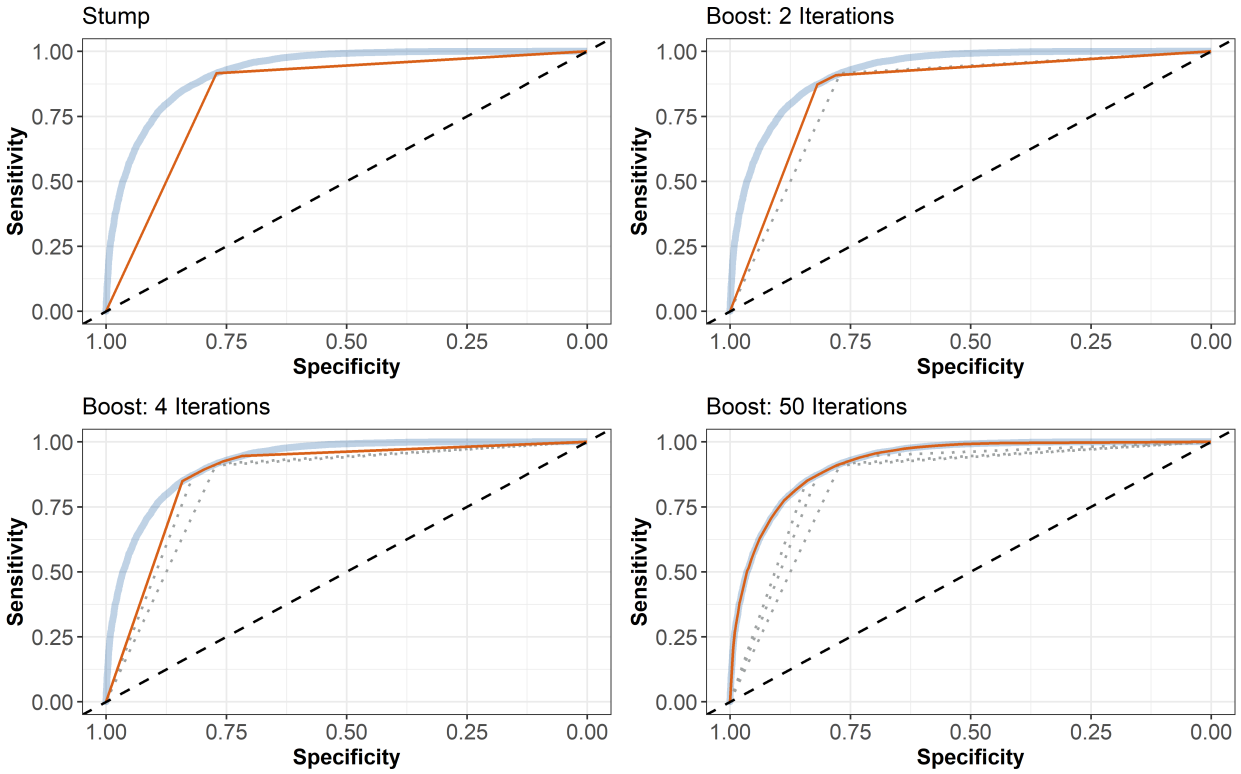
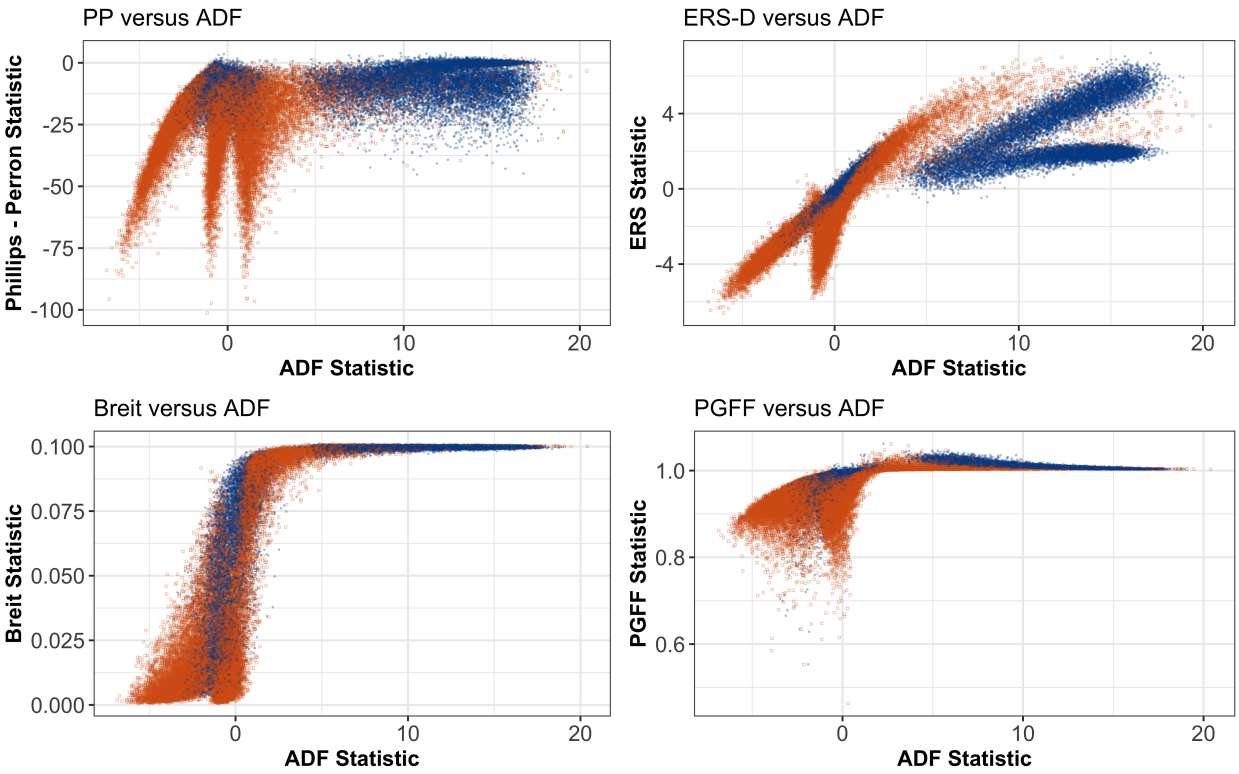


Figure 2: Receiver Operating Characteristic Curves: Convergence from Boosting



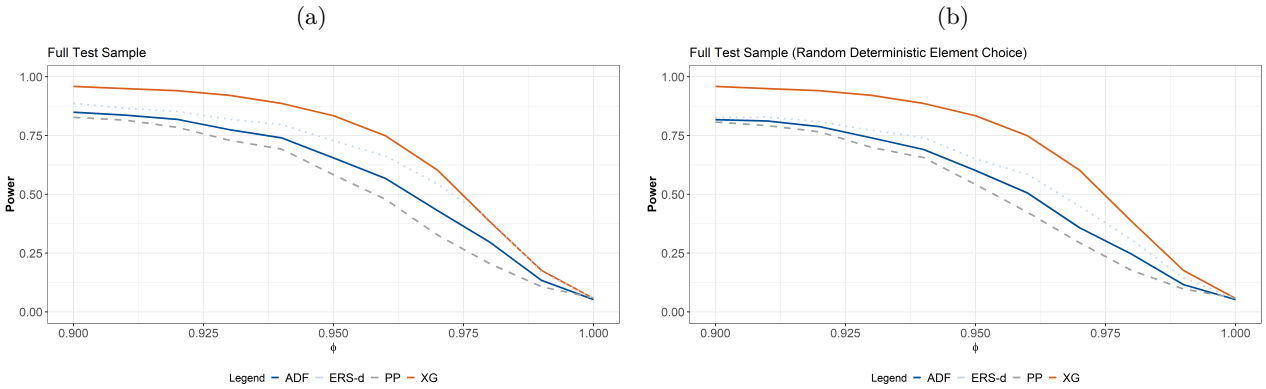
Note: Here we are plotting the Receiver Operating Characteristic (ROC) curve under the null (thick light-blue line) versus the boosted alternatives. To calculate the ROC curve under the null we calculated the corresponding probability for the ADF statistic under the cumulative probability distribution of the [simulated] null. This allows for a direct comparison to the predictions of the boosting algorithm.

Figure 3: Variation in Test Statistics.



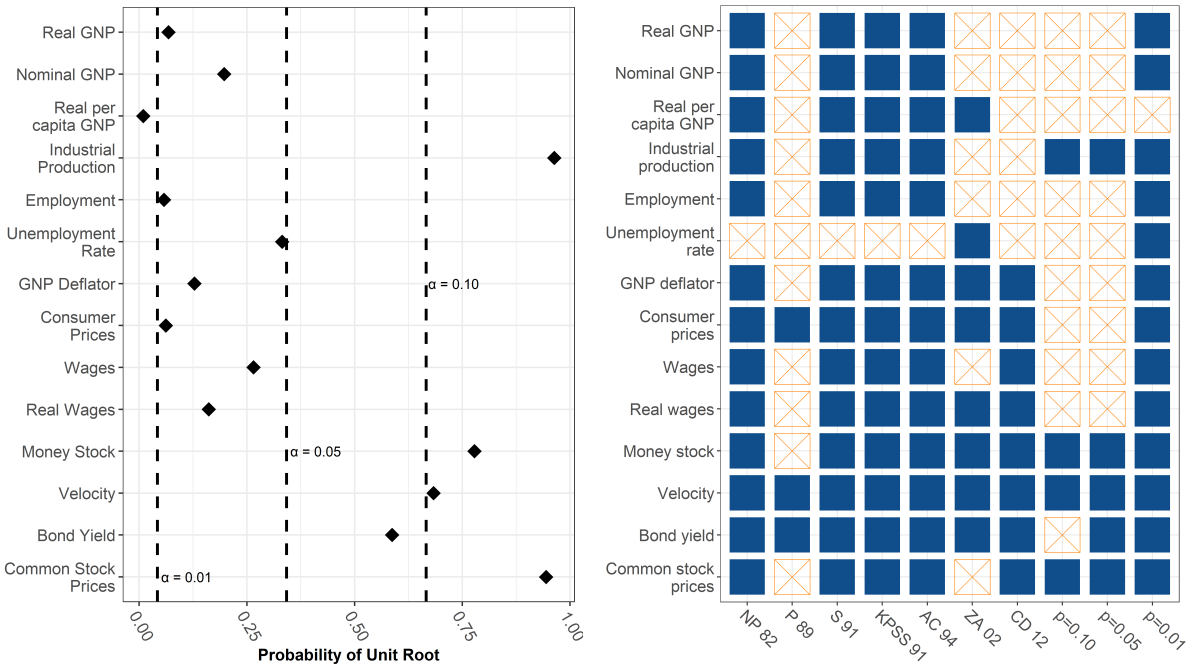
Note: Blue points are series with a unit root while orange points are stationary series.

Figure 4: Power Curves Comparing Test Statistics with ML-Based Test



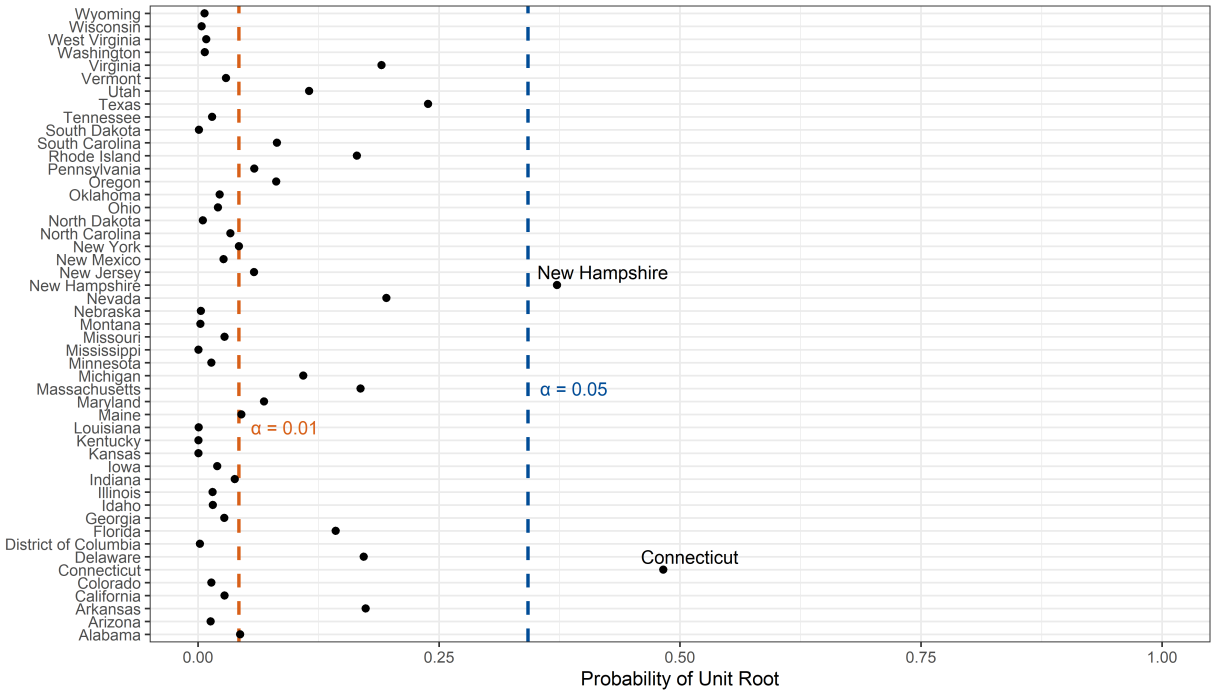
Note: In Figure 4a we plot the power curves for the proposed ML based test against several traditional testing options. These tests assume the practitioner desires a 5% Type I error rate and that she has perfect knowledge of the deterministic elements needed to best represent the data generating process. In Figure 4b we keep the 5% Type I error rate but now assume the practitioner has no *a priori* knowledge about the deterministic elements and includes them in tests at random.

Figure 5: Comparison of Findings on Data from Nelson and Plosser (1982)



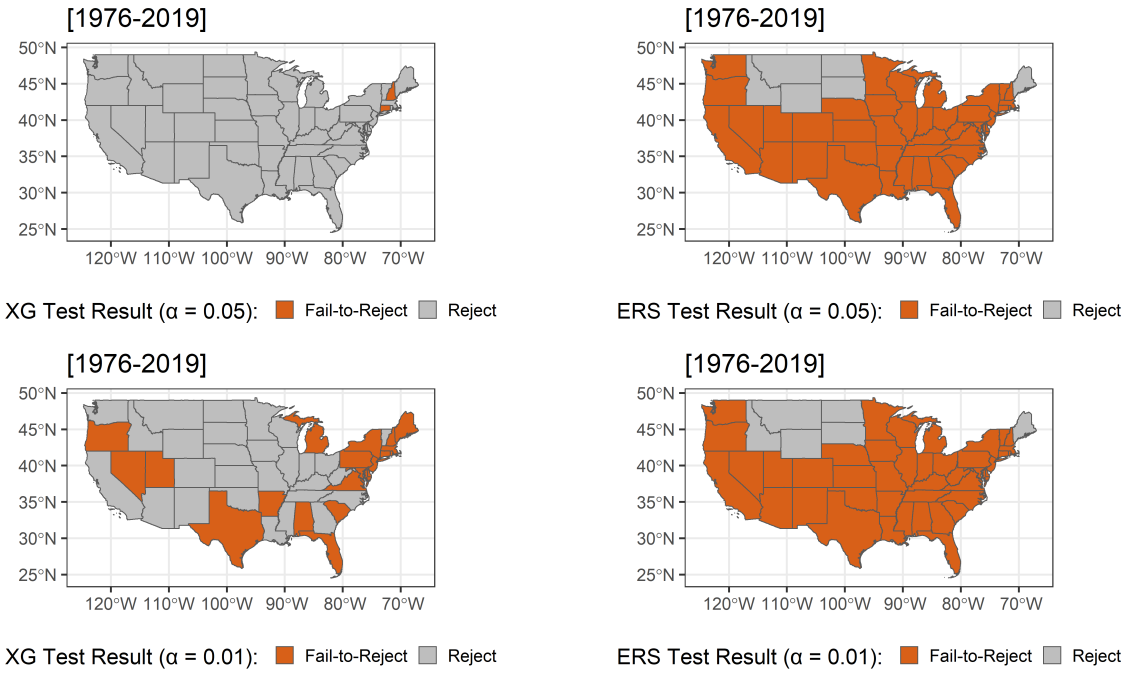
Note: Here we have plotted, on the left, the probability each series is a unit root (filled diamonds) in concert with thresholds based on $\alpha \in (0.10, 0.05, 0.01)$. On the right we have contextualized our results in the literature. If the related paper indicates the series is a unit root it is represented here by a shaded blue square. A decision indicating stationarity is shown here by an orange square with an X inside.

Figure 6: Is the Hysteresis Real?



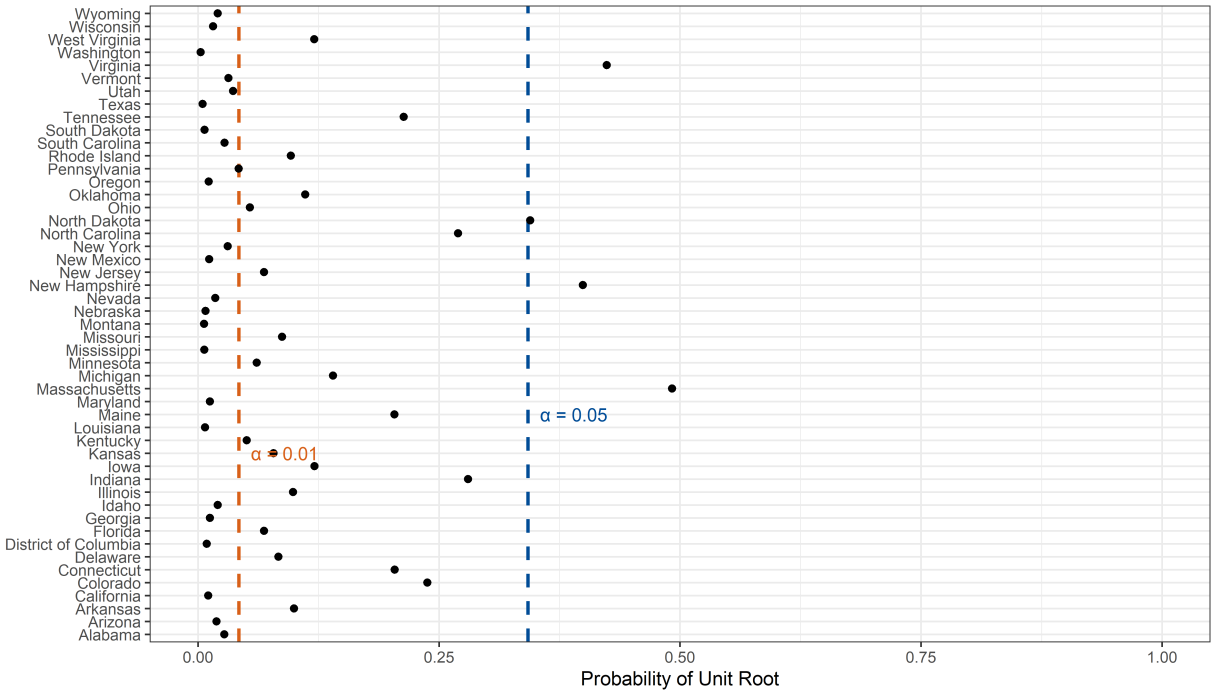
Note: Here we have plotted the probability that the time series, for each state, over the entire sample, contains a unit root. The vertical dashed lines represent decision thresholds which correspond to the standard $\alpha \in \{0.05, 0.01\}$. Any series to the right of the desired threshold should be considered a unit root and thus fail-to-reject the null hypothesis of hysteresis. The accuracy optimizing threshold would reject the null for all states. We provide similar plots for other time horizons in Appendix A.

Figure 7: Full Sample Geographic Comparison: Choice of α matters.



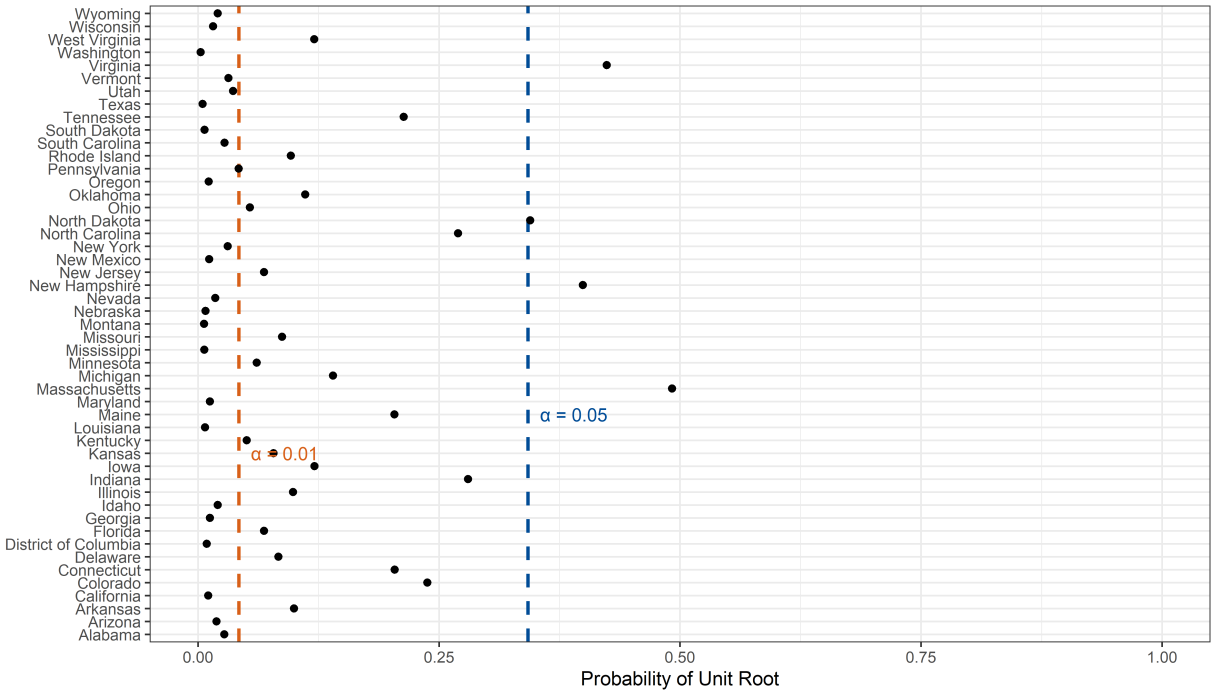
Note: Here we have plotted a map of the continental United States showing the test results on a full sample from $1976 - 2019$. Grey states are those in which the test rejects the null of a unit root and thus unemployment is mean reverting with a [relatively] short memory. States in orange are those which the test fails-to-reject the null of a unit root. The choice of α is highly relevant in the case of the proposed test as moving from $\alpha = 0.05$ to $\alpha = 0.01$ increases the number of states that fail-to-reject the null by eighteen.

Figure 8: Is the Hysteresis Real?[1992-2003]



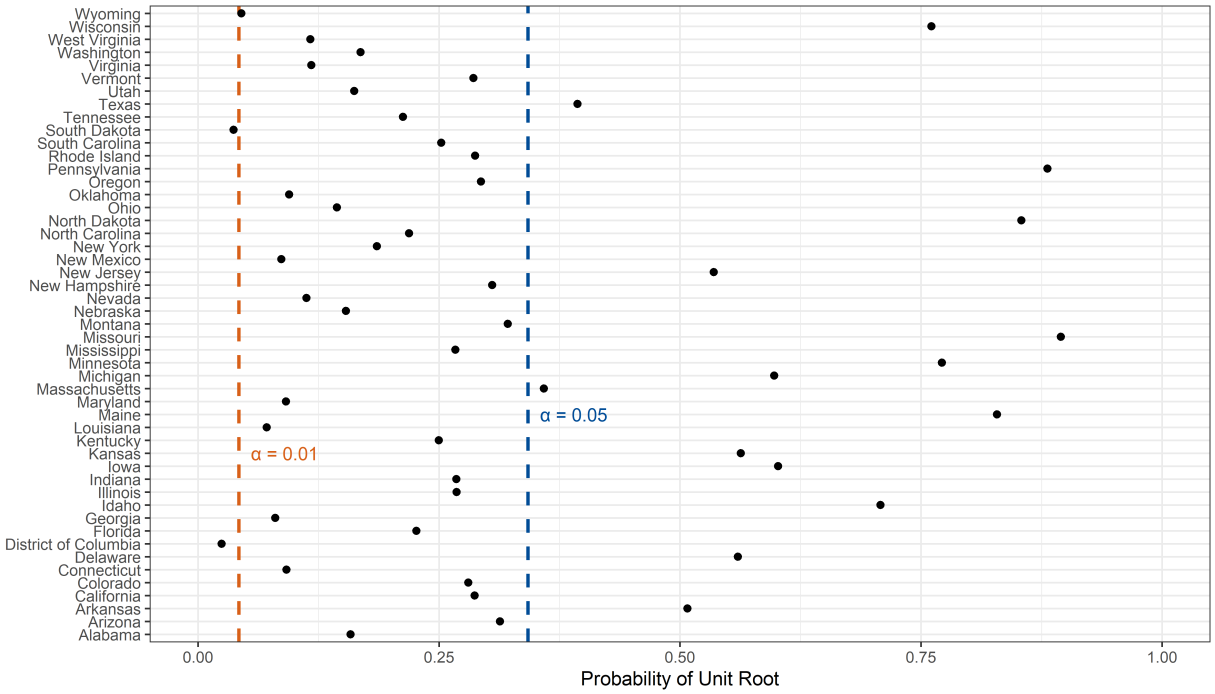
Note: Here we have plotted the probability that the time series, for each state, over the period of January 1992 to December 2003, contains a unit root. The vertical dashed lines represent decision thresholds which correspond to the standard $\alpha \in \{0.05, 0.01\}$. Any series to the right of the desired threshold should be considered a unit root and thus fail-to-reject the null hypothesis of hysteresis.

Figure 9: Is the Hysteresis Real?[2002-2019]



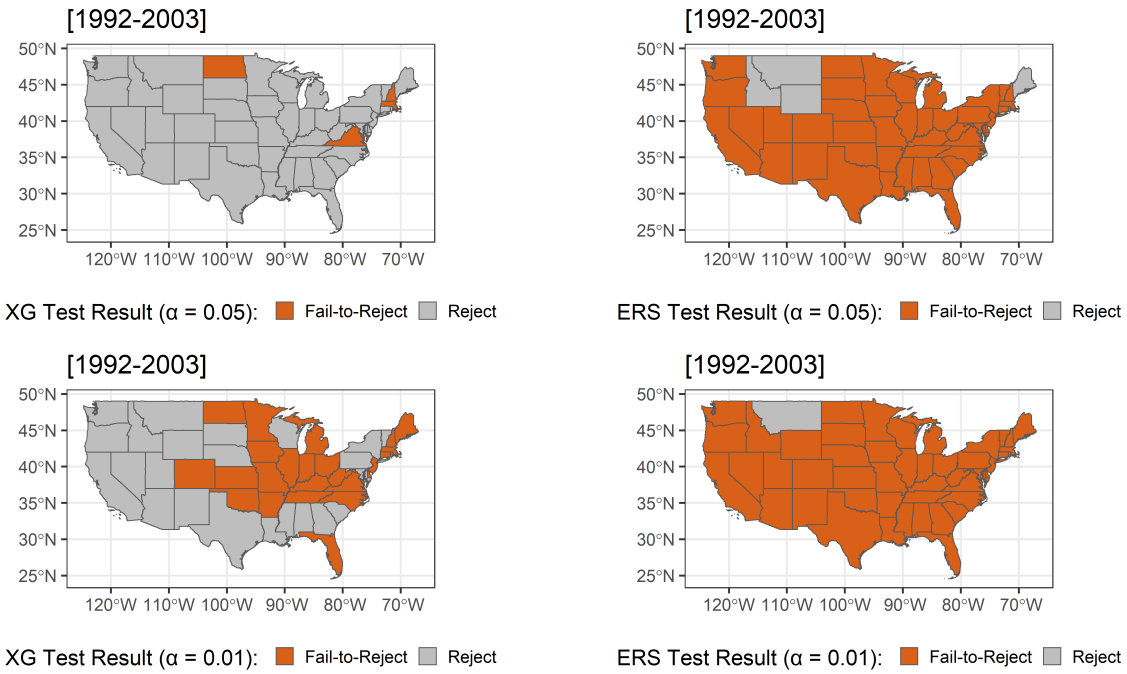
Note: Here we have plotted the probability that the time series, for each state, over the period of January 2002 to December 2019, contains a unit root. The vertical dashed lines represent decision thresholds which correspond to the standard $\alpha \in \{0.05, 0.01\}$. Any series to the right of the desired threshold should be considered a unit root and thus fail-to-reject the null hypothesis of hysteresis.

Figure 10: Is the Hysteresis Real?[2010-2019]



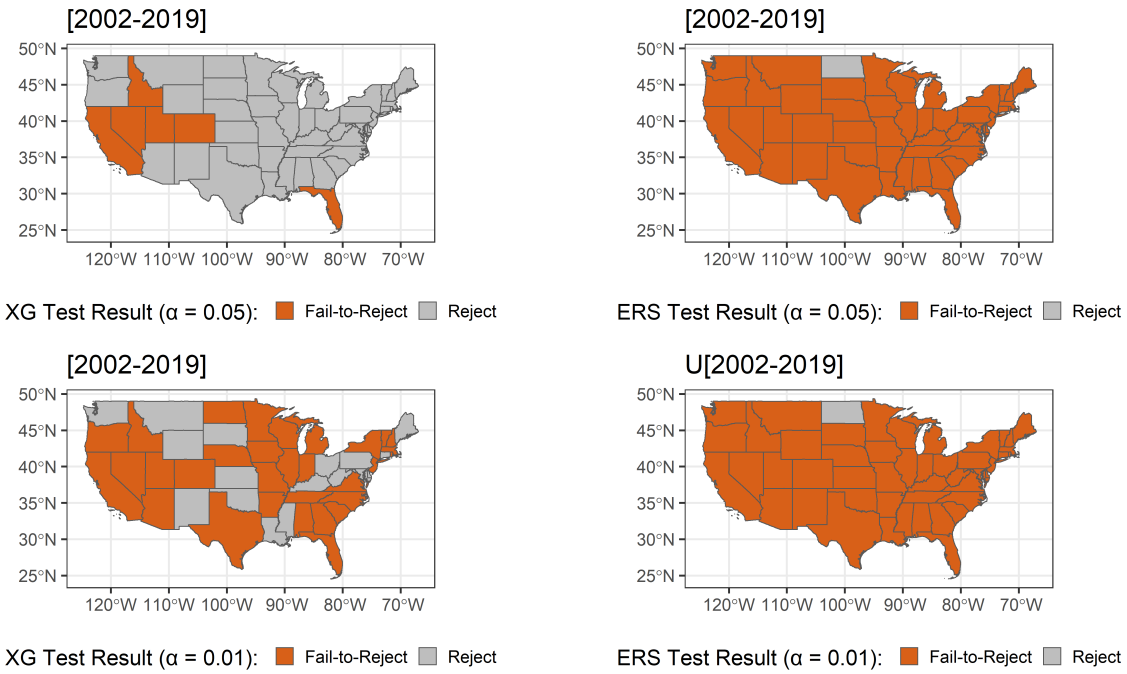
Note: Here we have plotted the probability that the time series, for each state, over the period of July 2010 to March 2019, contains a unit root. The vertical dashed lines represent decision thresholds which correspond to the standard $\alpha \in \{0.05, 0.01\}$. Any series to the right of the desired threshold should be considered a unit root and thus fail-to-reject the null hypothesis of hysteresis.

Figure 11: Sub-Sample Test Results: Choice of α matters.



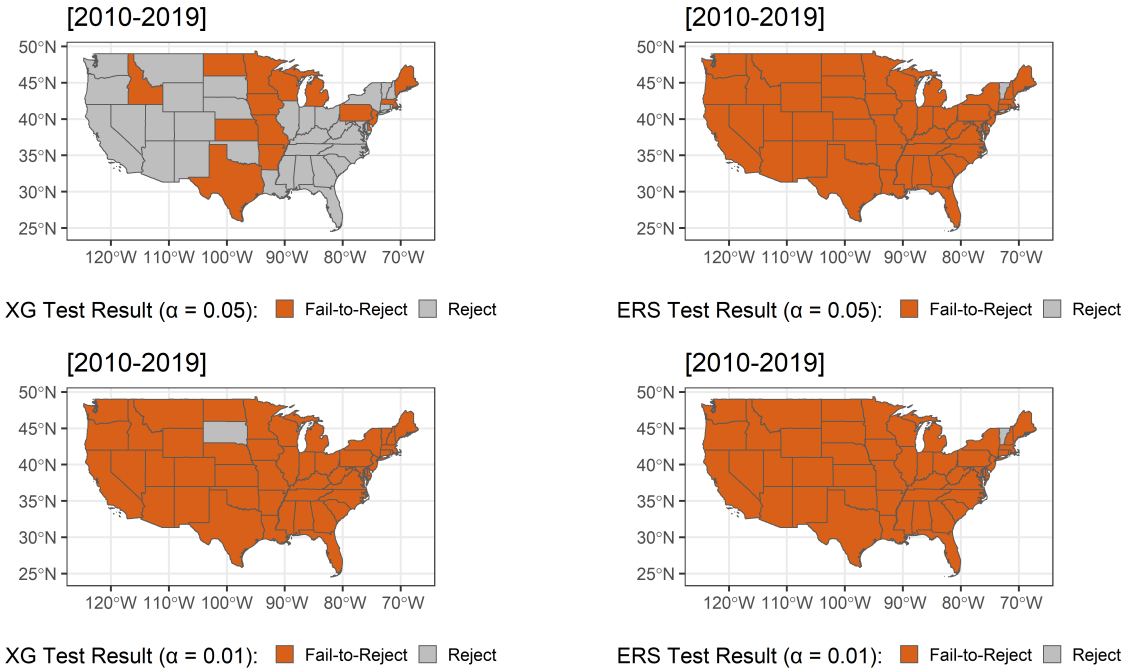
Note: Like Figure 10, here we have plotted the sub-sample from 1992-2003. This period includes the "dot-com" recession near the end of the period but otherwise was a period of remarkable economic growth. Using a 5% critical value we find that four state-level unemployment rates can be considered a unit root, three of which appear on the east coast. Using a standard DF-GLS test results in nearly all states failure-to-reject the null of a unit root. Using a 1% critical value results in our failure-to-reject the null for the majority of the Midwest region as well as additional portions of the east coast.

Figure 12: Sub-Sample Test Results: Choice of α matters.



Note: Like Figure 10, here we have plotted the sub-sample from 2002-2019. This period includes the great recession and subsequent recovery. Using a 5% critical value we find that six state-level unemployment rates can be considered a unit root, five of which appear in the western half of the country. Using a standard DF-GLS test results a failure-to-reject the null of a unit root for nearly all states.

Figure 13: Sub-Sample Test Results: Choice of α matters.



Note: Like Figure 10, here we have plotted the sub-sample from 2010-2019. This period includes the recovery from the great recession. Using a 5% critical value we find that six state-level unemployment rates fail-to-reject the null of a unit root, many of which appear in the middle of the country. Using a standard DF-GLS test results a failure-to-reject the null of a unit root for nearly all states.