WP/20/262

IMF Working Paper

unFEAR: Unsupervised Feature Extraction Clustering with an Application to Crisis Regimes Classification

by Jorge A. Chan-Lau and Ran Wang

IMF Working Papers **describe research in progress by the author(s) and are published to elicit comments and to encourage debate.** The views expressed in IMF Working Papers are those of the author(s) and do not necessarily represent the views of the IMF, its Executive Board, or IMF management.

INTERNATIONAL MONETARY FUND

WP/20/262

IMF Working Paper

Strategy, Policy and Review Department

UnFEAR: Unsupervised Feature Extraction Clustering with an Application to Crisis Regimes Classification

Prepared by Jorge A. Chan-Lau and Ran Wang

Authorized for distribution by Daria Zakharova

November 2020

***IMF Working Papers* describe research in progress by the author(s) and are published to elicit comments and to encourage debate.** The views expressed in IMF Working Papers are those of the author(s) and do not necessarily represent the views of the IMF, its Executive Board, or IMF management.

Abstract

We introduce **unFEAR**, Unsupervised Feature Extraction Clustering, to identify economic crisis regimes. Given labeled crisis and non-crisis episodes and the corresponding features values, unFEAR uses unsupervised representation learning and a novel mode contrastive autoencoder to group episodes into time-invariant non-overlapping clusters, each of which could be identified with a different regime. The likelihood that a country may experience an econmic crisis could be set equal to its cluster crisis frequency. Moreover, **unFEAR** could serve as a first step towards developing cluster-specific crisis prediction models tailored to each crisis regime.

JEL Classification Numbers: C1, C45, G1

Keywords: clustering, unsupervised feature extraction, autoencoder, machine learning, deep learning, biased label problem, crisis prediction

Authors' E-Mail Addresses: jchanlau@imf.org, ran.wang@email.ucr.edu

Content

Figures

Tables

1. Introduction[1]

Economic crises inflict substantial damage to the economy. Long-term economic costs, measured in terms of output foregone, are on average 5 percent for balance of payments crises; 10 percent for banking crises, and 15 percent for twin crises (Cerra and Saxena, 2008). Following a financial crisis, a country needs on average eight years to return to its pre-crisis level of income (Reinhard and Rogoff, 2014). Societal costs are also staggering, as average life expectancy declines, primary school enrollment drops, and infant mortality increases (Alexander et al. 2009; van Dijk, 2013).

Macroprudential policy has an important role in crisis prevention and crisis mitigation. The policy effectiveness, however, hinges on whether the macroprudential tools can target the root causes of economic crises. Crisis prediction models, hence, need to support macroprudential policy. By flagging in advance economic and financial conditions leading to an economic crises, the models can guide policy actions aimed at reducing the crisis likelihood.

This paper proposes **unFEAR**, an unsupervised feature extraction clustering method aimed at facilitating economic crisis prediction. The approach in unFEAR is quite different from that in other machine learning-based crisis prediction models. The latter adopt a supervised learning framework: at any time period, the models assign a crisis or no crisis label to a country's observed economic and financial predictor data based on whether the observation was followed or not by a crisis n periods ahead.

The reliance on labeled data gives rise to the biased label problem. Briefly, two countries characterized by similar economic and financial data may receive different labels as only one of them experienced a crisis in the near term. A supervised learner would try to separate both countries even though from a vulnerability perspective both countries belong to the same class. We explain the biased label problem in detail below.

unFEAR avoids the biased label problem using unsupervised learning to find clusters using information in the distribution of the economic and financial data. Rather than working with the raw data unFEAR leverages on the use of autoencoders to reduce the dimensionality of the original data set and generates time-invariant clusters using a novel mode contrastive autoencoder. The crisis and non-crisis observations in a cluster do not correspond to a specific time period, a finding that suggests that a time-invariant economic regime and crisis generating mechanism characterizes each cluster.

Once the clusters are identified it is possible to assess a country's crisis vulnerability at a given point in time. The simplest approach is to assign a country to its closest neighboring cluster. The crisis vulnerability is then calculated as the empirical crisis frequency in the cluster. A second approach, which is normally used in applied machine learning work, is to fit separate supervised learning classifiers to each cluster.

Both approaches for measuring crisis vulnerability could help guide macroprudential policy. Analysts could project the effect of policies on economic and financial fundamentals to determine whether a country may migrate to a safer or riskier cluster. Even if a country's cluster assignment does not change a supervised classifier estimated for the cluster could help to assess whether policies may contribute to reduce or to increase crisis risk.

The remainder of this paper offers first a brief overview of the literature on early warning and crisis prediction models and provides the needed background to understand the differences between previous machine learning-based crisis prediction models and the unFEAR method proposed here. The next section describes unFEAR in detail both at the conceptual and technical level. To illustrate unFEAR capability we apply it to a group of advanced economies using a data set of economic and financial variables covering the period 1980 - 2018. Crisis risk and crisis prediction is examined next, and the concluding section examines possible extensions of unFEAR.

[1] The authors thanks seminar participants at the IMF and AMRO-3 for useful discussions; Ziya Gorpe, Chengyu Huang, Tae-Hwy Lee, James Mongardini, Chris Redl, Ruoyao Shi, Aman Ullah, and Harry Peng Zhao for their comments; and Chuqiao Bi and Lamya Kejji for preparing the data sets. All omissions and errors are the authors' sole responsibility. Please address correspondence to both authors.

2. Machine Learning-Based Crisis Prediction Models

Work on crisis prediction models have largely side-stepped the use of the standard macroeconomic workhorse, the dynamic stochastic general equilibrium model (DSGE). While useful for conducting policy experiments the models do not perform well for forecasting crisis events partly due to the fact that these events are out-of-equilibrium states.[2] Unsurprisingly, most crisis prediction models are formulated as econometric and/or statistical models where economic theory serves to narrow the selection of predictive variables, or features.

The wave of speculative currency attacks on countries with fixed or pegged exchange rates experienced in the 1990s prompted the development of a first generation of crisis prediction models, also known as early warning models. Examples of such models include Frankel and Rose (1996), Kaminsky et al. (1998), and Berg and Patillo (1999) among others. Research on crisis prediction tapered off in the early 2000s as the Great Moderation brought a large decline in macroeconomic volatility (Bernanke, 2004).

Research resumed in the aftermath of the Great Recession in 2008, an event not foreseen by central banks, policy makers, and a majority of market participants. The ensuing studies focus on the reassessment of existing models and on the development of more accurate early warning systems. Example of such work include, among others, Babecky et al. (2012), Chamon and Crowe (2013), Christofides et al. (2016), and Ahuja et al. (2017).

More recently there has been much interest in developing machine learning based models for crisis prediction. The interest sparks from the success of machine learning models in prediction tasks in a vast range of knowledge domains outside economics. Recent examples include Alessi et al. (2014), Holpainen and Sarlin (2017), Beutel et al. (2018), Lang et al. (2018), and a number of studies conducted at the International Monetary Fund, with models specialized to predict external crises, financial crises, and fiscal crises.

The machine learning models cited above are supervised learning models. First, the set of explanatory variables (covariates or attributes) include observable economic and financial variables. In some cases, economic theory guides the selection of variables. In other cases, a large number of variables is included with the expectation that the machine learning algorithm will sort out what variables matter the most for crisis prediction. A data point is simply the set of attributes of a country at a given point in time.

Second, since the goal of the models is to predict crisis events, each data point is labeled as a crisis (or non-crisis) point when a crisis occurs (or does not occur) n periods ahead, that is, data points at time t serve to predict crisis and non-crisis events at time $t + n$. In models developed for policy making purposes, n typically ranges from one to two years. If the model flags a future crisis such relatively long prediction horizon leaves time for the authorities to implement preventive or mitigating measures. Finally, crisis definitions and the timing of the crisis are determined outside the model using expert domain knowledge.

Challenges in supervised learning crisis prediction models

Model estimation presents analysts with several challenges. First, despite the widespread perception in the popular press that economic crises recur frequently, crises are still rare events. Compared to non-crisis episodes, the number of non-crisis events largely exceeds that of crisis events, raising the issue of *imbalanced data* (Kotsiantis et al. 2006).

Second, the data sample includes as many countries as possible so that the learner algorithm can observe a non-negligible number of crisis events. Many countries, however, may lack observations for several of the attributes which raises the issue of *missing data.* One solution is to include only attributes with complete observations at the cost of discarding attributes containing useful information. Another solution is to eliminate observations for which the set of attributes is incomplete, which may drastically reduce the number of crisis observations and further worsen the imbalanced data problem. The third option, is to use *data imputation* methods to complete the data set by assigning values to any missing data observation, which raises the question on whether the imputed values truly represent the missing data. A final option is to allow the classifier to learn a set of functions, each one specialized to classify the data points using a subset of covariates.

[2] See Stiglitz (2017) for a critique, and Christiano et al. (2018) for a rebuttal.

Third, the set of attributes may include information extraneous for crisis prediction, a likely situation when the number of covariates is large. Extraneous information represents noise and makes model estimation more difficult. Adding to the model estimation complications is that two or more covariates may be strongly dependent. While covariate dependence may not harm the predictive ability of the model, it makes difficult to evaluate a particular covariate importance to predict an economic crisis.

The biased label problem and its unsupervised learning solution

Last but equally important, the biased label problem, may impair the predictive ability of a supervised learning model. Figure 1 illustrates this problem.

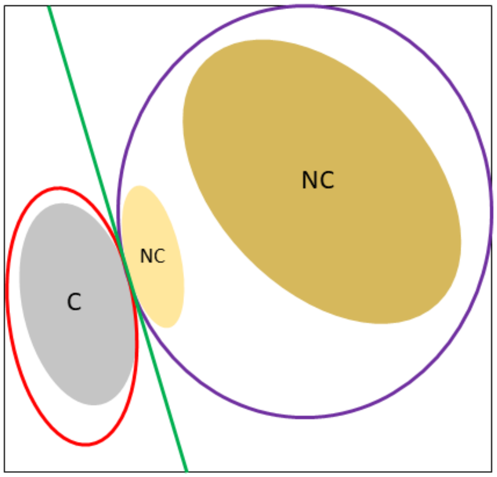

Figure 1: The biased label problem.

The figure is a simple two-dimensional representation of crisis and non-crisis points where there are only two features. Each point is represented by its features coordinates. The large circular purple cluster contains all the non-crisis points and the red elliptical cluster all the crisis points. A supervised learning classifier generates a separating hyperplane depicted as the the green line. The hyperplane imposes a hard separation between the crisis and non-crisis points. Ignoring the data points labels would yield a different separating hyperplane, one that separates the large non-crisis cluster from the crisis cluster and the small non-crisis cluster. The latter two clusters belong to a same class different from the non-crisis class. This situation reflects the fact that in this example the information the features convey cannot be used to discriminate properly between crisis and non-crisis labeled points.

The biased label problem is prevalent in policy crisis prediction models due to their long forecasting horizon. Two data points sharing the same characteristics, i.e. two different countries with the same economic fundamentals possibly measured at different times, may suffered a different fate two years ahead as only one of them would experience a crisis. There might be several explanations on why the countries' fates were so different, none of which the features are able to capture. For instance, the lucky country may experience a favorable commodity price movement that strengthens its fundamentals. Or economic policies may have been put in place that prevented the crisis.[3]

This is a situation unsupervised learning could handle adequately. An unsupervised learner, rather than forcing a hard separation between crisis and non-crisis points, assigns countries with similar features or economic fundamentals to different clusters. The crisis risk in the cluster corresponds roughly to the frequency of its crisis observations. Hence, it becomes possible to rank clusters in terms of crisis risk and to assign a crisis frequency to a data point even if the point label was not used to identify the cluster structure. One

[3] The biased label problem in crisis prediction is somewhat similar to the problem of label bias and fairness: data points are falsely attributed to a certain class even if the features may not justify it. See for instance, Jiang and Nachum (2019).

natural interpretation is that clusters represent different economic regimes, each with a different propensity to generate an economic crisis.

Mathematically a supervised classifier tries to estimate the conditional probability distribution of the crisis/non-crisis label, y, conditional on the features, x, i.e. $P(y|x)$. In contrast, the unsupervised classifier attempts to learn the unconditional probability distribution of the features, $P(x)$. From a computational and estimation perspective, an added advantage of the unsupervised classifier is that its estimation requires fewer data points (or examples) than a supervised learner to produce a reliable cluster structure. Also, there is no need to separate the available data into training, validation, and test sets.

It is also worth noting that adopting an unsupervised approach is consistent with economic intuition since we expect, given current knowledge of crisis dynamics and the partial information of economic and financial data, that in a group of countries with similar economic fundamentals some may experience a crisis and others may not. Hence, on a first pass, it makes sense to identify the clusters first using unsupervised learning and then to fit a cluster-specific supervised classifiers.

3. unFEAR: Unsupervised Feature Extraction Clustering for Crisis Regimes Classification

Conceptually, **unFEAR** is a simple method:

- First, it performs feature engineering (also known as feature learning or representation learning) to extract relevant information from the raw data set useful for clustering analysis.
- Second, once an appropriate representation is built, **unFEAR** identifies separate clusters and the corresponding data point assignments.

At first it may seem odd to perform feature engineering on the raw data since this is not yet usual practice in econometrics. Figures 2 and 3 illustrate why this step is necessary to obtain a suitable data representation. Namely:

- *Raw data attributes do not generate separable clusters.* The left panel in Figure 2 shows clusters obtained with K-means using annual data on data points comprising 75 attributes collected for 34 countries during the period 1970-2018. The raw data was used without any pre-processing prior to the application of the clustering algorithm, and the number of clusters was selected with a scree plot. The two-dimensional representation, generated using the t-SNE algorithm (van der Maaten and Hinton, 2008), shows that raw data attributes do not have enough discriminatory power.
- *Raw data capture different time periods rather than different economic fundamentals, i.e. the data exhibits time trends.* The right panel in Figure 2 simply places time labels, corresponding to different periods, over the data points without any attempt to assign them to clusters. The time labeled data points overlap substantially with the raw data-based clusters in the left panel of the figure (i.e. cluster 8 corresponds to the most recent data points). Absent feature engineering, an unsupervised learner may only pick data points in different time periods.

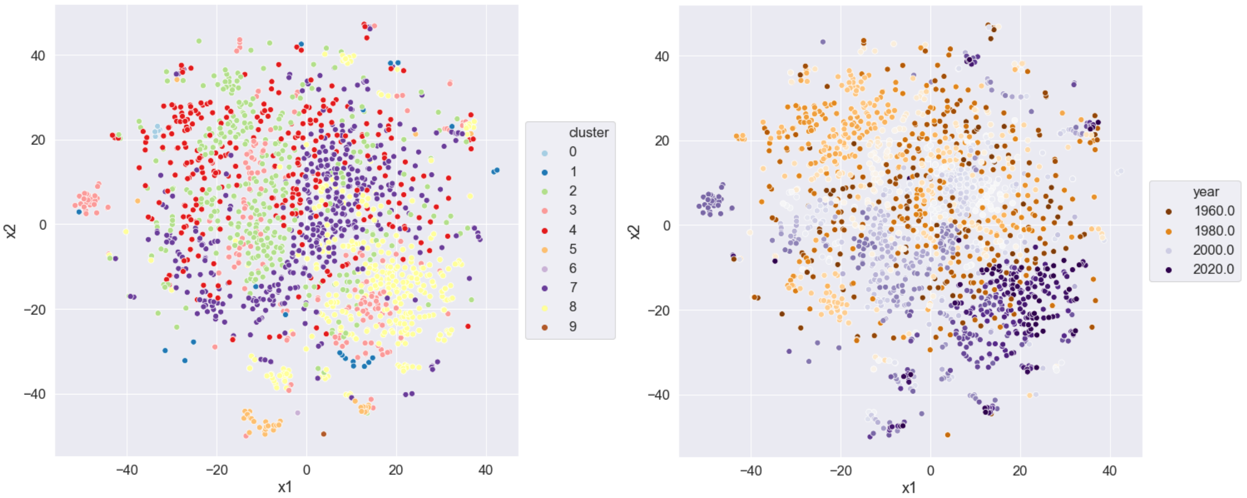

Figure 2: Raw data clusters: non-separability and time clustering

- *Raw data tends to group data points corresponding to the same country.* The left panel in Figure 3 shows the data points colored by countries. While the cluster structure remains badly defined, near neighbor points tend to correspond to the same country. Reliance on the raw attributes may yield clusters with a majority of data points corresponding to the same country.

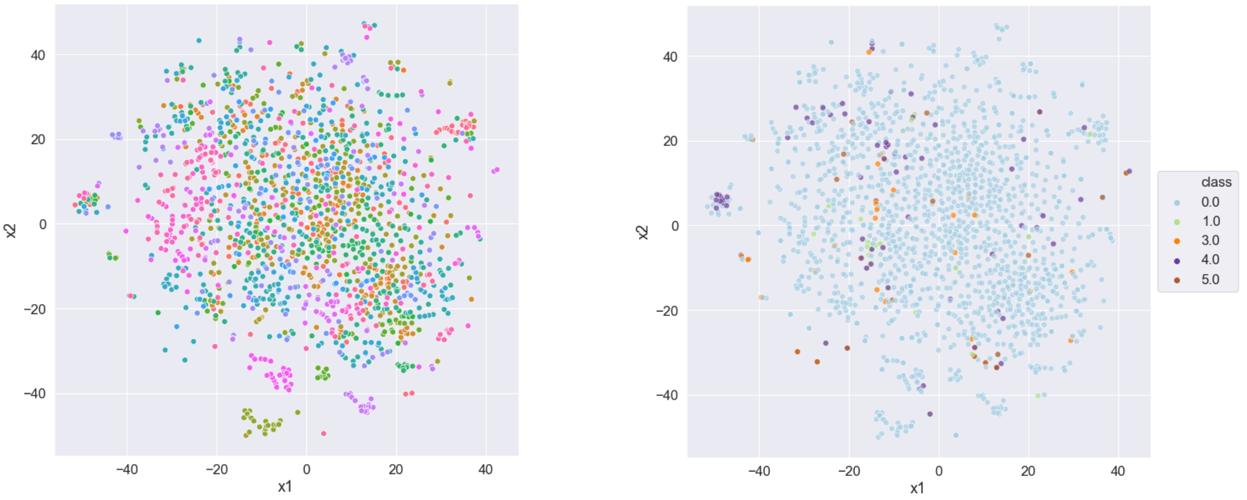

Figure 3: Raw data clusters: country bias and imbalanced data

- *Raw data is imbalanced, i.e. few crisis observations.* The right panel in Figure 3 shows the non-crisis points (in light blue) and the crisis points, which comprise 90 percent and 10 percent of the observations. An algorithm may find clusters biased to reflect the distribution of the non-crisis observations.
- *Missing data is a big issue.* It is not uncommon to find several missing and incomplete data points when putting together a common data set of economic and financial data for a large panel of countries. In our data set, for all observed data points only five variables do not have any missing value and for about two thirds of the variables (58), missing values could be found as in as much as fifty percent of the data points.

To perform feature engineering step we use **autoencoders**, which are commonly used in machine learning and deep learning, with a suitable loss function designed with the purpose to to address the first three issues described above, i.e. lack of separability in the data points, time clustering, and country clustering. After some experimentation, we fall back on the Synthetic Minority Over-sampling Technique (SMOTE) to address the data imbalance issue (Chawla et al. 2002). The next section describes in detail the technical details of the method.

Technical details

The main tool in the feature engineering task is the Autoencoder. To understand the logic behind it we first present its foundation, the multilayered network. Next we examine how Autoencoders work using as an analogy principal components analysis. Once the intuition is established it becomes straightforward to understand why autoencoders serve to input missing data, to remove time clusters, and to identify the different data point clusters.

Multilayer neural networks The multi-layered neural network is the basic workhorse of deep learning methods (Goodfellow, Bengio, and Courville, 2016). Figure 4 illustrates two neural networks.

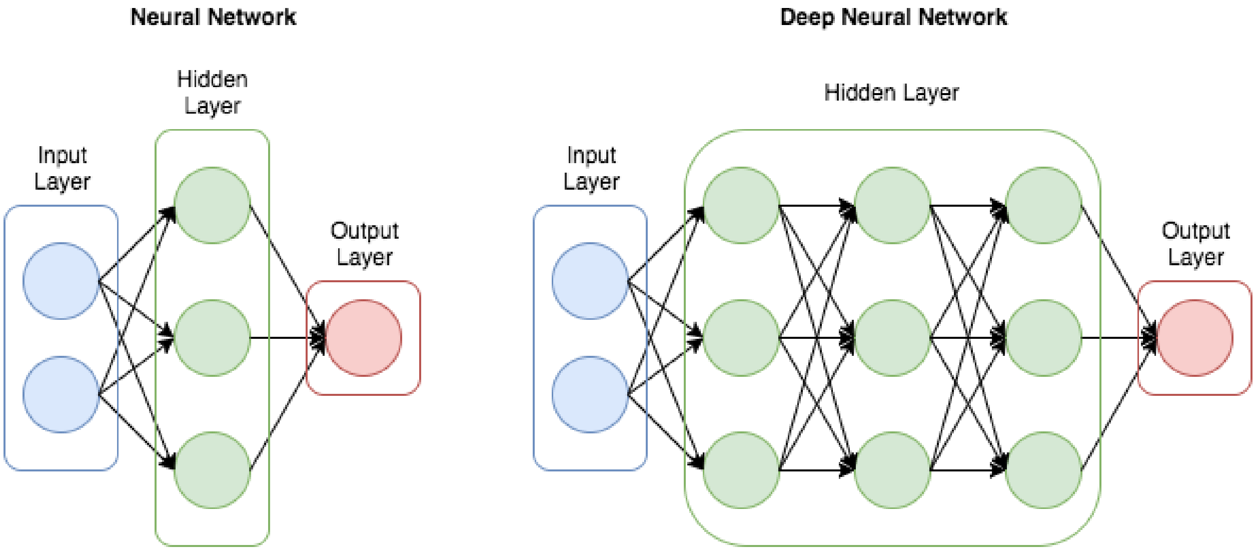

Figure 4: Two neural networks

The neural network in the left panel consists of three layers: the input layer, the hidden layer, and the output layer. From the outside, only the input, x, and the final output, y, are observed. The input layer collects the input, x, and feeds it to the hidden layer, whose units transform the input into an intermediate output, h, via a function g_1, i.e. $h = g_1(x)$. The intermediate output is then fed to the output layer, which processes it and produces the final output, y, using the function g_2, i.e. $y = g_2(h)$. Allowing the functions g_1 and g_2 to be non-linear allows the neural network to capture nonlinearities present in the data. The number of units in the hidden layer is a hyperparameter which is tuned (calibrated) using the data points.

It is possible to use multilayer networks, which contain several hidden layers instead of a single hidden layer. For instance, the right panel shows a three layer network. In this case, the output of the first hidden layer is the input of the second hidden layer. In turn, its output serves as an input to the third layer, whose output is then fed to the output layer. More generally, the transforming functions of the hidden layers can be specified recursively:

$$h_l = g_l(h_{l-1}),$$

where l is the l-th hidden layer, and g_l is a nonlinear transformation. Including several layers enables a deep learning network to captures the dependence between the input data and the output data in complex cases

(Pascanu et al., 2014; Arora et al., 2018). Autoencoders exploit this property to find data patterns, such as the joint probability distribution of the attributes, which then serves to input missing data; and the time clustering information in the data, which then allows removing time effects as explained later.

Understanding autoencoders To understand autoencoders it is useful to examine their conceptual similarity with principal components analysis (PCA), a standard method for dimensionality reduction widely applied in statistics as well as in finance and economics. PCA transforms the data input from its original space into an orthogonal space via a projection matrix, or in other words, it decomposes the data attributes along vectors (directions) orthogonal to each other (upper panel, Figure 5). It is possible to go from the orthogonal space to the original space, or reconstruct the inputs, if the projection matrix is known. To reduce the dimensionality of the original data input, we only keep a few components of the orthogonal space provided the retained components explain a substantial amount of the data total variance.

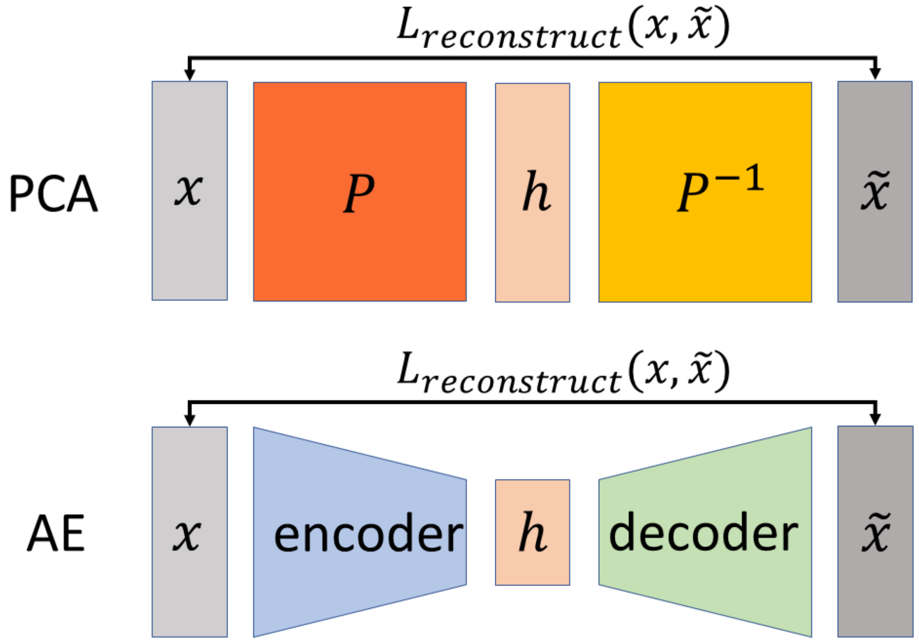

Figure 5: The analogy between principal components analysis and the autoencoder

Functionally, the matrix P is an encoder, i.e, it *encodes* the data using a few components and yields a low-dimensional representation, h. or the code in the transformed space. The inverse of the matrix P, P^{-1}, is a decoder, i.e. it *decodes* the coded, h, and returns an approximation of x, $\tilde{x}$. Ideally, what we want is to encode the data to obtain a good but simpler data representation while at the same time retain enough information such that the data approximation in the original space is adequate. The process of encoding and decoding the data yields the matrix P, which captures the relevant characteristics of the data input.

The autoencoder generalizes the PCA coding and decoding function beyond linear transformations and it comprises an encoder and a decoder, which are typically specified as multilayer neural networks (bottom panel, Figure 5). The encoder learns a function, $g_{encoder}$ by projecting the original input x onto h, with h contained in a lower dimensional space:

$$h = g_{encoder}(x).$$

We require the encoder to reduce the dimensionality of the data input in order to simplify any subsequent classification or learning process applied to the encoded data. In turn, the decoder learns the $g_{decoder}$ function that enables the autoencoder to reconstruct an approximation $\tilde{x}$ of the the original input x from the encoded representation h:

$$\tilde{x} = g_{decoder}(h).$$

To find the best data representation, it is necessary to specify a loss function associated with the reconstruction error, $L_{reconstruct}(x, \tilde{x})$:

$$L_{reconstruct}(x, \tilde{x}) = L(x, g_{decoder}(h)) = L(x, g_{decoder}(g_{encoder}(x))).$$

Optimization of the loss function yields the optimal encoder function $g_{encoder}(x)$ for the nonlinear transformation. By using multilayer networks, the autoencoder easily performs PCA when nonlinearities are present in the data. Stacking more layers and introducing noise in the encoder and decoder functions enable autoencoders to deliver complex but robust data transformations (Vincent et al., 2008 and 2010). We exploit the autoencoder properties to perform missing data imputation.

Missing data imputation with autoencoders Missing data imputation is often performed using one of the three following methods: replacing the missing value by a constant value, typically the median (median imputation) or the mean (mean imputation); resampling from the empirical distribution of the non-missing values; or exploiting the dependence among variables by regressing observed values on other variables and replacing the missing data by the predictions of the regression equations, such as done in the multivariate imputation by chained equations (MICE) method (Raghunathan et al., 2001, Van Buuren, 2007).

unFEAR introduces an autoencoder-based missing data imputation strategy using the Mean Squared Error loss function (MSE) to measure the reconstruction error (Figure 6).

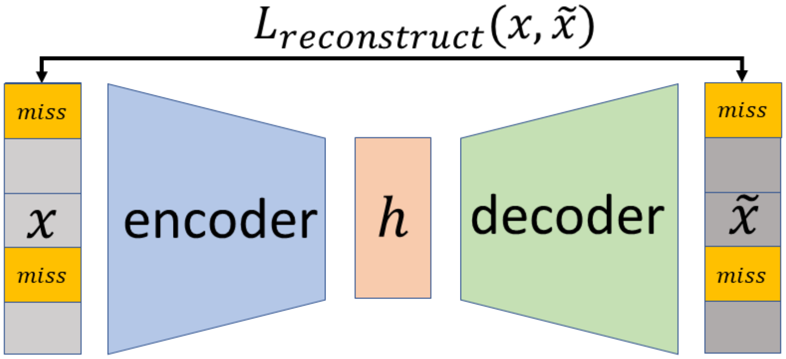

Figure 6: Missing data imputation using an autoencoder

The use of the autoencoder builds on the assumption that all the attributes (variables) in a high dimensional data exhibit dependence. The dependence assumption typically holds in reality especially for economic data. This fact enables the autoencoder strategy to recover missing attribute values using the observed values of other attributes. It is worth noting that the MICE also exploits the dependence assumption to justify regressing an attribute on other attributes. Contrary to MICE, the autoenconder does not need to specify whether an attribute with missing data can or cannot be used as a regressor. The autoencoder automatically assigns higher weights to attributes with more observations. Hence, the autoencoder method is equivalent to a data driven MICE method, combining resampling and dependence exploitation.[4]

The autoencoder missing data imputation method then reduces to:

- First, draw samples randomly from the non-missing data points.
- Second, train an autoencoder on the randomly drawn data sample.

[4] A related method is the Markov Chain Monte Carlo variational autoencoder-based of Rezende et al. (2015).

- Third, use the estimated autoencoder to fill the missing data values.[5]

Removing time trends using autoencoders Data exploration shows that the raw data exhibits time clusters or time trends, i.e. data points in certain time periods tend to be close to each other. For low dimensional data sets it is feasible to remove trends using univariate methods but they become burdensome as the number of attributes increases.

unFEAR uses a *Boosted Autoencoder* to remove time trends. The procedure is performed in several rounds. Each round starts with a trained autoencoder which allows us to reconstruct the approximated data input, $\tilde{x}$. The resulting reconstruction error, $r_i = x_i - \tilde{x_i}$ is then fed as an input for training a new autoencoder in the next round (Figure 7).

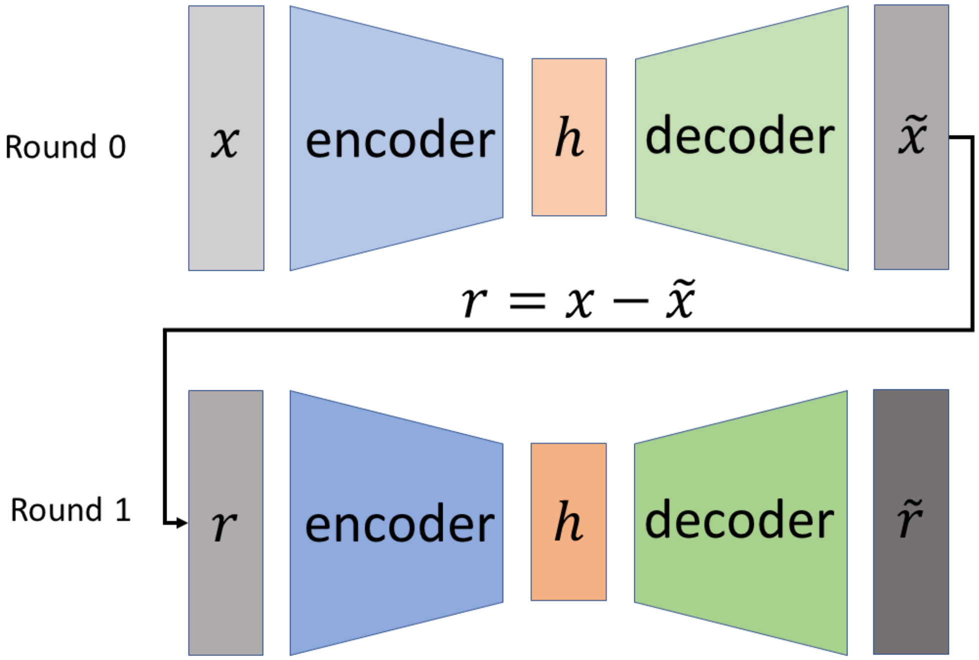

Figure 7: A Boosted Autoencoder

Time trends, either linear or non-linear, characterize the variables in the data set. The boosted autoencoder, in a first pass, learns to project the data input x onto a space containing the time trends. Hence, the reconstruction error $r_i = x_i - \tilde{x_i}$ does not exhibit a time trend but still retains other useful information contained in the data input. The unsupervised clustering approach we review next exploits this information.

Unsupervised clustering using the mode constrastive autoncoder This section introduces and explains the *mode contrastive autoencoder*, which is the key element in the **unFEAR**. Raw data, as Figures 2 and 3 illustrate, are not suitable for clustering analysis. A proper use of an autoencoder could enable us to find a feature representation that facilitates separability. The feature representation should meet two requirements:

- The transformed features should retain a substantial amount of the variation in the original data set to remain informative.

[5] The activation function of this autoencoder, as well as the others **unFEAR** uses, is an exponential linear unit (ELUs) (Clevert et al., 2016). The convergence speed of ELUs outperforms that of rectified linear units (ReLUs) (Klambauer et al., 2017).

- In the transformed space, the data points concentrate in to several separable clusters.

The first requirement is a common one in the construction of multilayer networks. It forces the autoencoder to learn the best representation of the data that yields a low reconstruction error. The second requirement is necessary since the first one, by itself, does not ensure the autoencoder learns to separate the data into clusters.

Enforcing the first requirement needs the autoencoder to minimize a regularized loss function that balances the reconstruction error and the need to group the data points in the transformed space into separate clusters. The regularized loss function, L_{AE}, is:

$$L_{AE} = L_{reconstruct}(x, \tilde{x}) + \lambda L_{cluster}(h, \mu), \tag{1}$$

where x collects all the data points $(x_1, \ldots, x_N)$, $L_{reconstruct}$ is the standard reconstruction error, λ is a weight tuning parameter, and $L_{cluster}$ is the regularization term forcing the autoencoder to separate the data into clusters using as inputs the output of the encoder, $h = g_{encoder}(x)$, and the centroids of the clusters, $\mu = (\mu_1, \ldots, \mu_K)$.[6]

The regularized loss function induces the autoencoder to learn a encoder $g_{encoder}(x)$ such that an original data point, x_i, when transformed into the encoder output, $h_i = g_{encoder}(x_i)$, can be assigned to a single cluster with centroid μ_c.

The specification of the regularization term is the key element in the **unFEAR** method. To specify it, we follow an approach similar to the one van der Maaten and Hinton (2008) used to derive their t-Distributed Stochastic Neighbor Embeddings method (t-SNE). We start by specifying the conditional probability that the data point x_i belongs to the c-th cluster, $P(\mu_c|x_i)$ (or equivalently, that the closer neighbor of the data point x_i is the c-th cluster) as:

$$P(\mu_c|x_i) = \frac{(1 + ||\mu_c - g_{encoder}(x_i)||^2)^{-1}}{\sum_{j=1,\ldots,K}(1 + ||\mu_c - g_{encoder}(x_j)||^2)^{-1}}, \tag{2}$$

where K, a hyperparameter, is the number of clusters and $||.||^2$ is the Euclidean or L_2 norm. Ideally, we want to assign the transformed data point $h_i = g_{encoder}(x_i)$ to a single cluster to ensure the clusters do not overlap and are separable. This implies that the conditional probability distribution in equation (2) should peak at a single value μ_c and take low values, ideally zero, at other cluster centroids. In other words, we want $P(\mu_c|x_i)$ to be a one-peaked probability distribution as close as possible to a delta distribution.

This is equivalent to perform K-means clustering by maximizing the likelihood function:

$$\mathcal{L}(\mu, g; x) = \prod_{c=1}^{K}\prod_{i=1}^{N} P(\mu_c|x_i)P(x_i \in \text{cluster } c) \tag{3}$$

or its log-likelihood. The expectation-maximization algorithm of Dempster et al. (1977) yields the following iterative procedure to maximize the log-likelihood:

E-step: given the centroids $\mu = (\mu_1, \ldots, \mu_K)$ and the encoder g, assign to data point x_i the cluster c_i with the maximum log-probability value:

$$c_i = \arg \max_{c_i \in (1,\ldots,K)} \log(P(\mu|x_i; g)),$$

where the conditional probability is given by equation (2) and we have made explicit its dependence on the encoder g.

[6] A more complex alternative to the use of a regularized loss function, as done here, is to use a denoising autoencoder incorporating the cluster requirement into the reconstruction error. On denoising autoencoders, see Alain and Bengio (2014).

M-step: given the cluster assignments for each data point, find the new centroid μ_c of cluster c solving the minimization problem below:

$$\mu, g = \arg\min_{\mu, g} \left(-\sum_i label_i \odot \log(P(\mu|x_i; g)) \right), \quad c = 1, \ldots, K.$$

where $label_i = (I(x_i \in \text{cluster } 1), I(x_i \in \text{cluster } 2), \dot{;} I(x_i \in \text{cluster } K))$ is the one-hot encoded label for x_i, and $I(x_i \in \text{cluster } c)$ is the indicator function.

If follows naturally to set the $L_{cluster}$ equal to $(-\sum_i \mu_c \log(P(\mu_c|x_i)))$ since we want the autoencoder to perform K-means clustering. The autoenconder, hence, is a mode contrastive autoencoder (MCAE) since it tries to separate the different modes of the clusters.The mode contrastive loss function L_{MCAE} is:

$$L_{MCAE} = L_{reconstruct}(x_i, \tilde{x_i}) + \lambda \left(-\sum_{c=1}^{K} \sum_{i=1}^{N} I(x_i \in \text{cluster } c) \log(P(\mu_c|x_i; g)) \right). \tag{4}$$

Figure 8 illustrates the role of the loss function terms in the mode contrastive autoencoder.

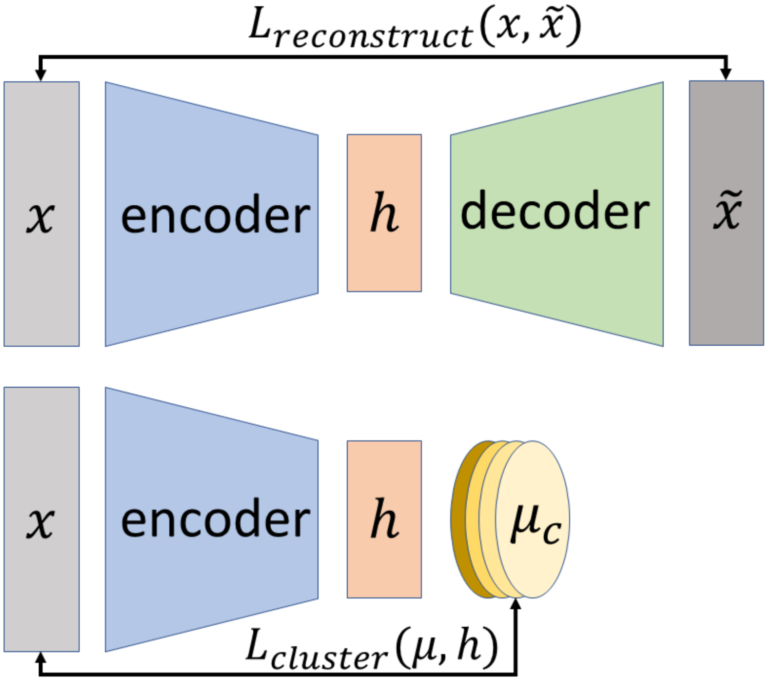

Figure 8: The Mode Contrastive Autoencoder

Minimizing the loss function in equation (4) is possible using expectation maximization iteration for a given encoder g:

E-step: this step is similar to the E-step in the log-likelihood maximization. Given the centroids μ_c, $c = 1, \ldots K$, and the encoder g, assign each data point x_i to a cluster c_i with the maximum log-probability value:

$$c_i = \arg \max_{c_i \in (1, \ldots, K)} \log(P(\mu_c|x_i; g)).$$

M-step: Find the new cluster centroids μ_c, $c = 1, \ldots, K$ that minimize the loss function L_{MCAE}:

$$L_{MCAE} = L_{reconstruct}(x_i, \tilde{x_i}) + \lambda \left(- \sum_c \sum_i I(x_i \in \text{cluster } c) \log(P(\mu_c | x_i; g)) \right).$$

4. Application: Identification of Economic Crisis Clusters

This section illustrates the use of **unFEAR** to identify economic crisis clusters, which in turn, could facilitate the task of crisis prediction. Predicting an economic crisis in advance matters to policy makers and macro-strategists. The goal of the former group is to put in place policy measures to prevent the crisis from realizing, and the goal of the latter is to profit from the event by betting against falling asset prices.

Data

The data in the analysis covers 34 countries during the period 1970 - 2018 (Table 1).

Table 1. Country list

Australia	Austria	Belgium	Canada	Cyprus
Czech Republic	Denmark	Estonia	Finland	France
Germany	Greece	Hong Kong S.A.R	Iceland	Ireland
Israel	Italy	Japan	Korea	Luxembourg
Malta	Netherlands	New Zealand	Norway	Portugal
San Marino	Singapore	Slovakia	Slovenia	Spain
Sweden	Switzerland	United Kingdom	United States	

The data comprises 1688 data points where each data point is a country-year observation, with 75 attributes. The attributes are constructed using levels, differences, and Hodrick-Prescott trends of the following variables:[7]

Global variables

- Oil prices
- 3-month U.S. Treasury bill rate, constant maturity
- 10-year U.S. real interest rate
- Trade-weighted dollar currency index, major currencies

Domestic economic variables

- GDP growth
- Output gap
- Inflation
- Reserves
- Total external debt
- Debt revenue
- Exports and Imports
- Capital flows
- Exchange rate against the U.S. dollar
- Purchasing power parity
- Real exchange rate
- Terms of trade
- Fiscal balance
- Fiscal revenue
- Fiscal expenses

Domestic financial variables

- Probability of default, banking sector
- Probability of default, non-financial sector
- Probability of default, non-bank financial sector
- Investment grade securities, share in total stock of debt securities
- Long-term bond yields
- Stock prices
- Price to income ratio, housing sector
- Price to rent ratio, housing sector
- Aggregate bank capitalization ratio,
- Bank assets to GDP ratio
- Credit to GDP ratio

[7] A detailed description of the attributes is available upon request from the authors. Most variables are available from public IMF databases and/or private data providers. Probabilities of default are from the Credit Research Initiative at the Asian Institute of Digital Finance, National University of Singapore (https://rmicri.org). Researchers can access PD data upon registration.

- Loan to deposit ratio, banking sector
- Short-term deposit rates
- Private sector indebtedness to GDP ratio
- Financial access
- Financial efficiency in the financial sector

Other variables

- Natural disasters, material impact on GDP
- Years elapsed since the end of a crisis episode
- Cumulative number of years recorded as a crisis episode since country entered the database

A data point is labeled as a crisis if an economic crisis affects the country two years after the data point is observed and recorded. The crisis labels correspond to one of each of the following categories: external crisis, as defined in Basu et al. (2017); financial crisis, as defined in Laeven and Valencia (2017); fiscal crisis, as defined in Medas et al. (2018); and real sector crisis, as defined in Basu e et al. (2017). The crisis/no-crisis labels are not used to find the economic crisis clusters to avoid the biased label problem. The labels are used ex-post: once the clusters are identified and data points assigned to them, the labels are disclosed to assess a cluster's crisis frequency.

Feature representation with autoencoders

As explained earlier in section 2 and illustrated in Figure 2 above, the information conveyed by the data in a raw form does not generate clearly separable clusters while tending to cluster data points in time periods, a trivial result. It is possible to achieve a better feature representation using autoencoders, as shown below.

Removing time trends

To remove the time trends or effects we implement a standard autoencoder with five dimensional hidden vectors h, which is trained using the original raw data input x. The autoencoder residuals are obtained subtracting the reconstructed data, $\tilde{x}$, from x, i.e. $r = x - \tilde{x}$. K-means clustering serves to assign the data point residuals, corresponding to country-year observations, to one of ten clusters, where the number of clusters is determined using a scree plot. Figures 9 and 10 show the results.

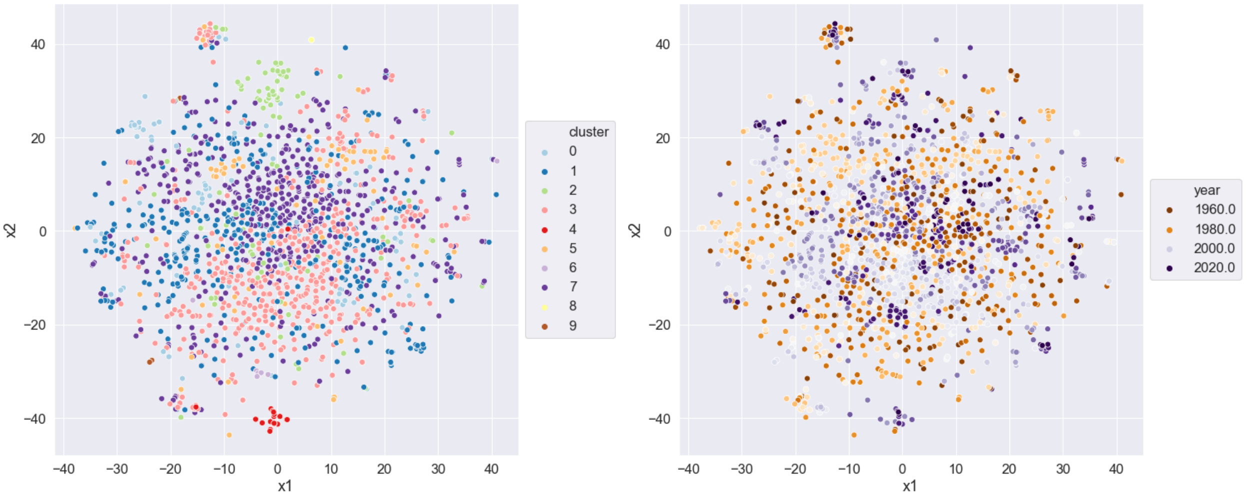

Figure 9: Time detrended data clusters: K-mean clusters and time periods

The residuals obtained from the first pass of the autoencoder tend to cluster in groups not clearly separable (Figure 9, left panel), except for one cluster (cluster 4, bottom center of the figure). However, time clustering has mostly vanished (Figure 9, right panel). We can also examine whether clusters group data points corresponding to the same country: the answer is negative as data clusters comprise data points from different countries (Figure 10, left panel). Similarly, the clusters do not seem to be mainly comprised by data points corresponding to the same label (Figure 10, right panel).[8]

[8] Class 0 corresponds to the no crisis label, class 1 to financial crisis, class 2 to a sudden stop crisis, class 3 to an exchange rate market pressure index event, class 4 to a real sector crisis, and class 5 to a fiscal crisis.

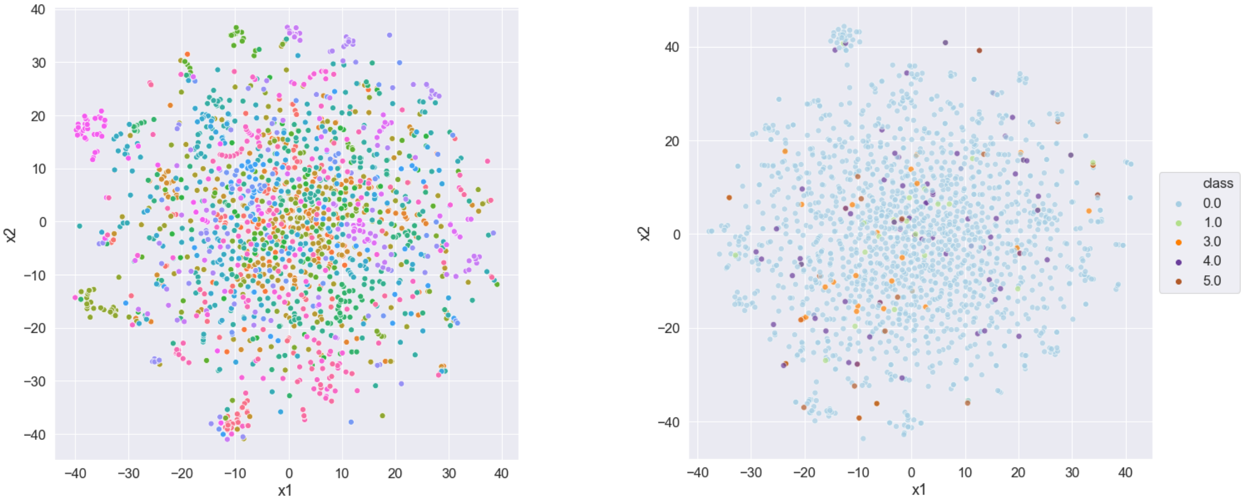

Figure 10: Time detrended data clusters: country and crisis presence

Balancing the data using SMOTE

After removing the time trend, it is necessary to address the imbalanced data problem. In supervised learning imbalanced data could often produce inaccurate predictions. While the problem is less severe in unsupervised learning since the learner does not use the label information. In our setup, however, it is still the case that since the number of data points labeled as no-crisis points is large, the learner may be biased to use mostly these points to identify the clusters.

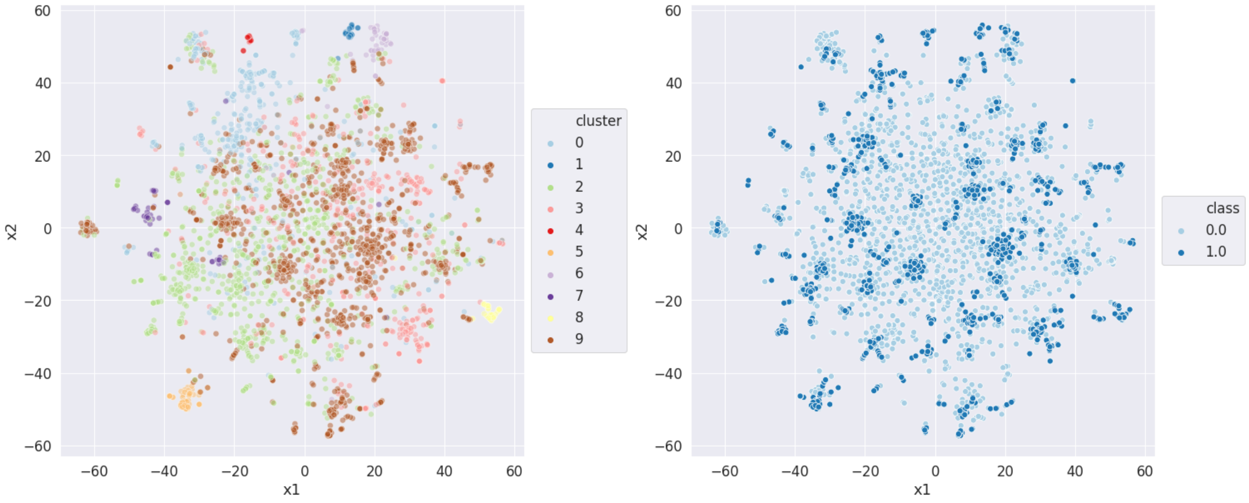

Figure 11: Time detrended balanced data: clusters and crisis/non-crisis observations

To resolve this issue we implement the SMOTE method to create synthetic crisis data points and to improve the accuracy of the unsupervised learner when applied to crisis prediction. Using SMOTE assumes that the feature distribution of data points labeled as crisis, for all crisis labels, is very similar when contrasted with the features distribution of data points labeled as non-crisis.

Figure 11, left panel, shows the K-means clusters obtained after applying SMOTE to the time-detrended features, i.e. the residuals after applying the autoencoders to the raw data. The cluster structure still suggests that there is scope for improving the feature representation. Nevertheless, as the right panel shows, crisis-labeled data points start to separate from the non-crisis labeled data points.

Clustering via Mode Contrastive Autoencoder (MCAE)

The standard autoencoder attempts to minimize a loss function proportional to the difference between the original data points and the reconstructed data points without regard for whether the residuals exhibit a multicluster structure. The mode contrastive autoencoder presented in section 3 is able to capture the data structure, by minimizing the residuals, and to assign the data points to unique clusters, thanks to the inclusion of a negative log-likelihood term associated with cluster assignments as shown in equation (4).

The number of clusters is a hyperparameter in the MCAE. In the absence of specific selection rules in the clustering literature we apply the elbow method to the scree plot of the mean squared distance between the data points and their centroid assignment for different number of clusters. Figure 12 shows the scree plot obtained applying MCAE for a number of clusters ranging from 2 to 20. We base our analysis on nine clusters since there are no substantial gains by adding more clusters.

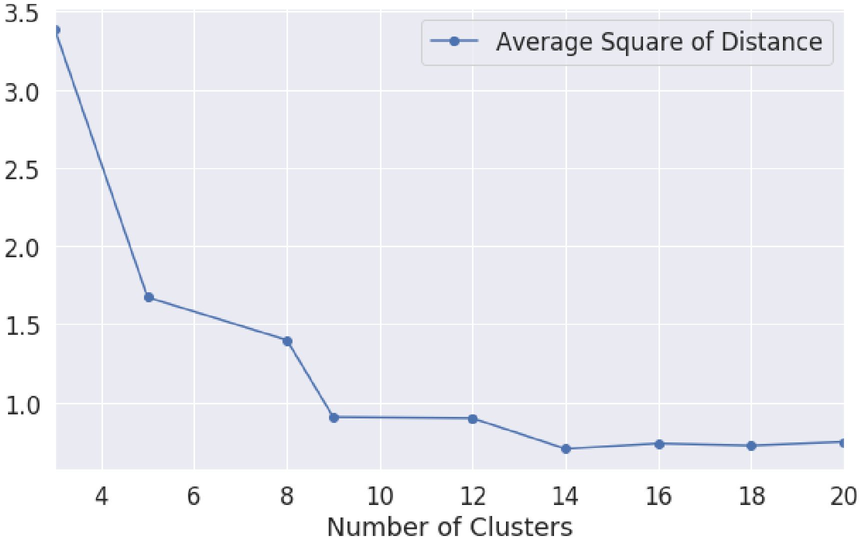

Figure 12: Scree plot for cluster selection

Figure 13 illustrates the results obtained applying the 9-cluster MCAE to the residuals obtained by the first pass of a standard autoencoder. The left panel shows nine well differentiated clusters. Each cluster could be interpreted as a different economic regime. Under the assumption of ergodicity, i.e. the past economic regimes are recurrent, we could expect a current or future data point to belong to one of the clusters.

Recall that the MCAE does not use the labels when performing data reconstruction and clustering assignment. When labels are applied, they reveal that the MCAE clusters contain both crisis and non-crisis points coexist (Figure 13, right panel). This finding indicates that there are no risk-free clusters but some are safer than others in terms of crisis frequency. In addition, compared with raw data clusters, the MCAE clusters show a clear separation between crisis and non-crisis data points.

Before discussing in more detail the crisis prediction task we assess whether some important information may be missing after applying the MCAE. The assessment is based on the distribution of the MCAE residuals. When viewed within the cluster structure (Figure 14, left panel), some of the residuals still tend to aggregate into three small separate clusters, suggesting MCAE may have missed some clustering information. When viewed from the perspective of crisis and non-crisis labeled data points (Figure 14, right panel), the spatial distribution of the residuals is very similar for both classes. These results indicate that the unFEAR method is able to extract an appropriate feature representation useful for identifying recurrent economic regimes and their crisis generating mechanisms.

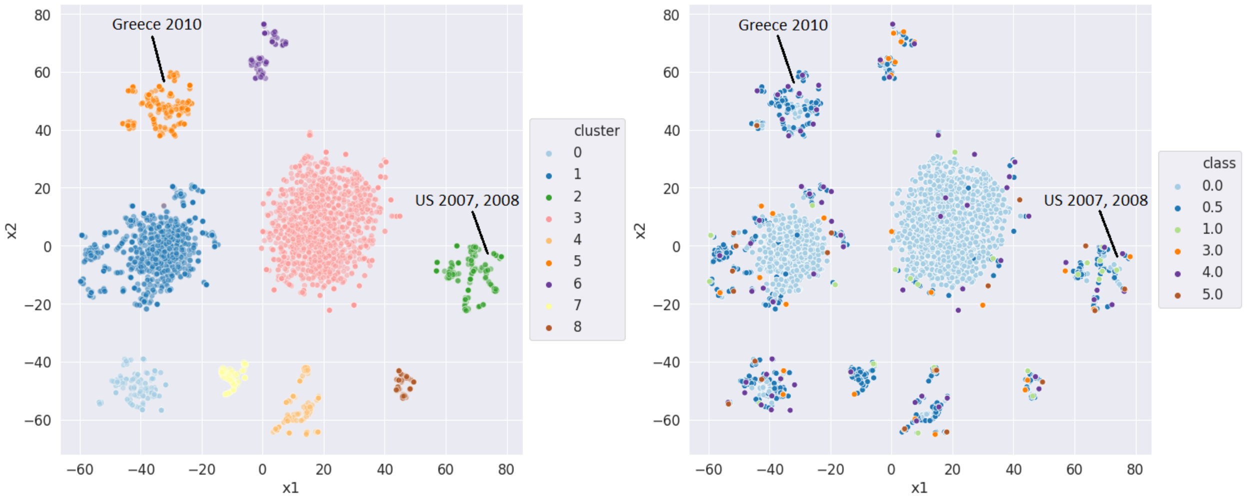

Figure 13: Mode Contrastive Autoencoder: clusters and crisis data points

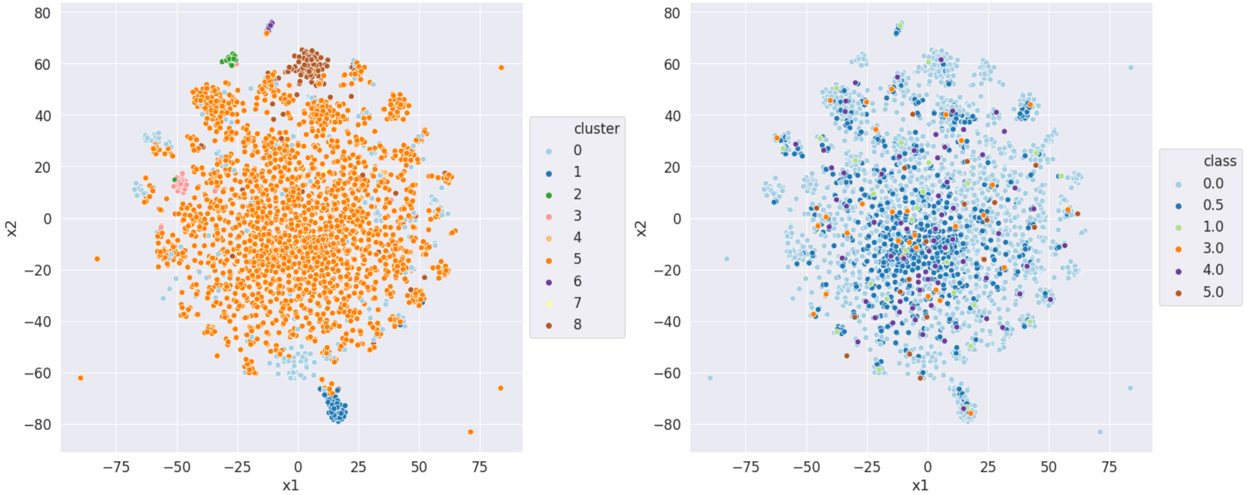

Figure 14: Mode Contrastive Autoencoder: residuals

Crisis risk measurement and crisis prediction

unFEAR, after learning an appropriate feature representation, produces clear and well separated cluster, each characterizing one of ten possible crisis clusters (Figure 13, right panel). For instance, the two larger clusters are safer, from a crisis realization perspective, than the smaller clusters since the number of crisis points is small relative to the number of non-crisis points. We could consider a country has a low crisis risk if its data point falls into any of these two clusters.

Table 2 summarizes the crisis frequencies of each cluster using two different measures. The empirical frequency is the ratio of observed crisis data points to the total number of observed data points. The shadow frequency is the ratio of observed and synthetically generated crisis data points to the total number of data points.

Table 2. Crisis clusters: empirical and shadow crisis frequencies

	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Crisis data points	190	307	211	143	168
Non-crisis data points	24	452	14	1003	6
Empirical crisis frequency, in percent	43	6	61	2	71
Shadow crisis frequency, in percent	89	40	94	12	97
	Cluster 5	Cluster 6	Cluster 7	Cluster 8	
Crisis data points	232	120	94	74	
Non-crisis data points	33	5	0	2	
Empirical crisis frequency, in percent	30	71	100	85	
Shadow crisis frequency, in percent	88	96	100	97	

Crisis and non-crisis data points correspond to the number of data points, both observed and synthetic, classified as crisis and non-crisis respectively. Empirical frequency is the ratio of the number of observed crisis data points to the total number of observed data points, and the shadow empirical frequency is the ratio of the number of observed crisis data points to the total number of data points.

From an empirical frequency perspective two clusters, clusters 1 and 3, are low crisis risk clusters, in which 6 percent and 2 percent of the observed data points are crisis observations. From the shadow crisis frequency perspective only cluster 1 can be characterized as low crisis risk (12 percent). Tables 3 and 4 show the crisis observations in each cluster and highlight that a country could experience several crisis types in the same year.

Table 3. Low empirical crisis frequency clusters

Cluster 1: non-crisis observations = 452, crisis observations = 307,
empirical crisis frequency = 6 percent, shadow crisis frequency = 40 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1974	United States	4	1991	*Finland*	4	1999	Spain	3
1974	Cyprus	4	1991	Iceland	5	2000	Malta	3
1975	Belgium	4	1991	Estonia	5	2000	Iceland	5
1975	Portugal	4	1991	Czech Republic	5	2009	France	4
1976	New Zealand	4	1991	United Kingdom	4	2009	Austria	4
1981	Slovenia	5	1991	Canada	4	2009	Finland	4
1982	United States	4	1992	Estonia	1	2009	Belgium	4
1982	Canada	4	1993	Portugal	3	2010	Estonia	3
1983	Cyprus	5	1995	Malta	3	2011	Italy	3
1993	Slovenia	1	1996	Czech Republic	1			
1991	*Finland*	1	1998	Slovakia	1			

Cluster 3: non-crisis observations = 1003, crisis observations = 143,
empirical crisis frequency = 2 percent, shadow crisis frequency = 12 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1975	Switzerland	4	1991	Australia	1	2000	Denmark	3
1981	Netherlands	4	1991	New Zealand	4	2005	Switzerland	3
1982	Germany	4	1991	Switzerland	4	2007	United Kingdom	1
1982	Switzerland	4	1991	Norway	4	2008	*Luxembourg*	4
1983	Australia	4	1992	Finland	1	2008	*Luxembourg*	1
1983	Portugal	4	1993	Greece	3	2008	San Marino	4
1983	Iceland	5	1997	Japan	5	2008	Spain	1
1988	United States	4	1998	Hong Kong SAR	4	2009	Switzerland	4

Crisis and non-crisis observations correspond to the number of data points, both observed and synthetic, classified as crisis and non-crisis respectively. Country names in italic denote countries that experienced multiple crises in the same year. Crises: financial (1), sudden stop (2), exchange rate market pressure (3), real (4), fiscal (5).

Table 4. High empirical crisis frequency clusters

Cluster 0: non-crisis observations = 24, crisis observations = 190,
empirical crisis frequency = 43 percent, shadow crisis frequency = 89 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1974	Greece	4	1981	Belgium	4	2008	*Denmark*	4
1974	Japan	4	1985	New Zealand	5	2008	*Denmark*	1
1977	Sweden	4	2001	Iceland	3	2008	*Estonia*	4
1978	Norway	4	2006	Slovenia	3	2008	*Sweden*	4
1979	New Zealand	4	2008	Sweden	1	2009	*Spain*	4
1981	Spain	4	2008	Estonia	5	2012	*Spain*	5

Cluster 2: non-crisis observations = 14, crisis observations = 211,
empirical crisis frequency = 61 percent, shadow crisis frequency = 94 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1980	Korea	4	2002	Israel	3	2008	United States	4
1991	Slovakia	5	2007	United States	1	2009	Iceland	4
1991	Sweden	1	2008	Japan	4	2012	*Cyprus*	3
1991	Sweden	4	2008	Portugal	1	2012	*Cyprus*	4
1997	*Korea*	1	2008	Switzerland	1	2012	*Cyprus*	5
1997	*Korea*	3	2008	Netherlands	1	2012	Iceland	3
1997	*Korea*	5	2008	Germany	1			
1998	Korea	4	2008	Belgium	1			

Cluster 4: non-crisis observations = 6, crisis observations = 168,
empirical crisis frequency = 71 percent, shadow crisis frequency = 97 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1992	Israel	3	2008	*Ireland*	1	2011	*Portugal*	5
1998	Singapore	4	2008	*Ireland*	4	2011	*Spain*	3
2001	Singapore	4	2009	Netherlands	4	2011	*Spain*	4
2008	*Iceland*	1	2010	*Ireland*	3	2011	Cyprus	1
2008	*Iceland*	3	2010	*Ireland*	5			
2008	*Iceland*	5	2011	*Portugal*	3			

Cluster 5: non-crisis observations = 33, crisis observations = 232,
empirical crisis frequency = 30 percent, shadow crisis frequency = 88 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1974	United Kingdom	4	2008	*Greece*	1	2009	Slovakia	4
1975	Italy	4	2008	Greece	4	2010	*Greece*	3
1980	United States	4	2009	Germany	4	2010	*Greece*	5
1980	United Kingdom	4	2009	Czech Republic	4	2012	Italy	4
1981	Greece	4	2009	Slovakia	4			

Cluster 6: non-crisis observations = 5, crisis observations = 120,
empirical crisis frequency = 71 percent, shadow crisis frequency = 96 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1980	Denmark	4	1993	Spain	4	2008	*New Zealand*	3
1990	Malta	3	1998	New Zealand	3	2008	*New Zealand*	4
1993	Sweden	3	2008	*United Kingdom*	3	2008	Korea	3
1993	Spain	3	2008	*United Kingdom*	4	2013	Japan	3

Crisis and non-crisis observations correspond to the number of data points, both observed and synthetic, classified as crisis and non-crisis respectively. Country names in italic denote countries that experienced multiple crises in the same year. Crises: financial (1), sudden stop (2), exchange rate market pressure (3), real (4), fiscal (5).

Table 4. High empirical crisis frequency clusters (continued)

Cluster 7: non-crisis observations = 0, crisis observations = 94,
empirical crisis frequency = 100 percent, shadow crisis frequency = 100 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1993	Italy	3	2008	France	1	2008	*Italy*	4
2008	Austria	1	2008	*Italy*	1			

Cluster 8: non-crisis observations = 2, crisis observations = 74,
empirical crisis frequency = 85 percent, shadow crisis frequency = 97 percent

Year	Country	Class	Year	Country	Class	Year	Country	Class
1986	Norway	5	2008	Slovenia	1	2015	Norway	3
1992	Slovenia	1	2009	Norway	4			
1998	Norway	3	2009	Canada	4			

Crisis and non-crisis observations correspond to the number of data points, both observed and synthetic, classified as crisis and non-crisis respectively. Country names in italic denote countries that experienced multiple crises in the same year. Crises: financial (1), sudden stop (2), exchange rate market pressure (3), real (4), fiscal (5).

The cluster structure identified by unFEAR, and illustrated in Figure 13, serves as the starting point for crisis risk measurement. Specifically, a country's economic fundamentals place it in one of the clusters. The crisis frequency serves as a measure of crisis risk and can be further decomposed by crisis type. As an example, suppose a country is assigned to cluster 8. In this case, crisis risk is high since the empirical frequency is 85 percent, which can be decomposed into financial crisis risk (24 percent or 2/7 of 85 percent), exchange market pressure crisis (24 percent), real sector crisis (24 percent) and fiscal sector crisis (13 percent). The risk of simultaneous crises seems negligible.

We want to point here two other extensions not undertaken in this study. First, the examination of crisis and non-crisis observations in a cluster could also serve to understand why some countries may not experience a crisis despite sharing the same economic fundamentals as crisis-affected countries. Second, Figure 13 shows that fitting supervised classification models for each cluster is relatively straightforward compared with fitting a global classification model on all the data set. unFEAR hence provides an adequate feature representation which can improve the precision of the crisis prediction task. In a first stage clusters are identified, and in a second stage, supervised learning models are fitted to each cluster.

5. Conclusions

Crisis prediction in policy making institutions benefits greatly from the increased adoption of machine learning-based predictive models. One potential concern in supervised learning-based models is the biased label problem: countries sharing similar weak economic fundamentals may or may not experience a future crisis due either to luck or policy actions. The biased label problem is more severe the longer the prediction horizon is. The more time elapses since when the prediction was made, the likelier that random events or policies may alter the outcome.

Unsupervised learning methods can avoid the biased label problem and cluster countries based on the similarity of their economic fundamentals. To this end, we introduced a new unsupervised feature extraction clustering method, **unFEAR**, where a novel mode contrastive autoencoder helped to identify observation clusters. Moreover, unFEAR can handle time effects and missing data efficiently.

To illustrate unFEAR's usefulness, we applied it to a sample of advanced economies. The data points to the existence of eight different clusters we associate with economic regimes, only one of which comprising most of the observations could be considered a low risk. A country cluster assignment serve to assess its crisis risk, and the cluster per se could serve as building blocks for simpler and more precise supervised learning-based crisis prediction models.

References

Ahuja, A., K. Wiseman, M. Syed. 2017. “Assessing country risk — selected approaches.” IMF Technical Notes and Manuals 17/08 (Washington: International Monetary Fund).

Alain, G., Y. Bengio. 2014. “What regularized auto-encoders learn from the data-generating distribution.” *J. of Machine Learning Research* 15: 3743—3773.

Alessi, L. et al. 2014. “Comparing different early warning systems: results from a horse race competition among members of the Macro-prudential Research Network.” Mimeo.

Alexander, M., M. Harding, C. Lamarche. 2008. “The human cost of economic crises.” SIEPR Working Paper 08-029. Stanford Institute for Economic Policy Research.

Arora, R., A. Basu, P. Mianjy, A. Mukherjee. 2018. “Understanding Deep Neural Networks with Rectified Linear Units.” *International Conference on Learning Representations.*

Basu, S., M. Chamon, C. Crowe. 2017. “A model to assess the probabilities of growth, fiscal, and financial crises.” IMF Working Paper WP/17/282.

Babecky, J. T. Havranek, J. Mateju, M. Rusnak, KI. Smidkova, B. Vacisek. 2012. “Banking, debt, and currency crises: early warning indicators for developed countries.” ECB Working Paper 1485 (Frankfurt, European Central Bank).

Berg, A., C. Pattillo. 1999. “Are currency crisis predictable? A test.” *IMF Staff Papers* 46(2): 107–138.

Bernanke, B. 2004. “The Great Moderation.” Remarks at the Eastern Economic Association, February 20 (Washington, D.C.)

Beutel, J., S. List, G. von Schweinitz. 2018. “An evaluation of early warning models for systemic banking crises: does machine learning improve predictions?” Discussion Paper No. 48/2018 (Frankfurt, Deutsche Bundesbank).

van Buuren, S. 2007. “Multiple imputation of discrete and continuous data by fully conditional specification.” *Statistical Methods in Medical Research* 16(3): 219—242.

Cerra, V., S. Saxena. 2008. “Growth dynamics: the myth of economic recovery.” *American Economic Review* 98(1): 439—57.

Chamon, M., C. Crowe. 2013. “Predictive indicators of financial crises.” In Caprio, G., T. Beck, S. Claessens, S. Schmukler (eds.). *The Evidence and Impact of Financial Globalization* (Academic Press).

Chawla, N.V. N. Japkowicz, A. Kotcz. 2004. “Special issue on learning from imbalanced data sets.” *ACM Sigkdd Explorations Newsletter* 6(1): 1—117.

Chawla, N.V., K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer. 2002. “SMOTE: synthetic minority over-sampling technique.” *J. of Artificial Intelligence Research* 16: 321—357.

Christiano, L.J., M.S. Eichenbaum, M. Trabandt. 2018. “On DSGE models.” *J. of Economic Perspectives* 32(3): 113—140.

Christofides, C.T., T.S. Eicher, C. Papageorgiou. 2016. “Did established early warning signals predict the 2008 crisis?” *European Economic Review* 81(1): 103—114.

Clevert, D.-A., T. Unterthiner, S. Hochreiter. 2016. “Fast and accurate deep network learning by exponential linear units (ELUs).” *Fourth International Conference on Learning Representations.*

Dempster, A., N. Laird, D. Rubin. 1977. “Maximum likelihood from incomplete data via the EM algorithm.” *J. of the Royal Statistical Society Series B (Methodological)* 39 (1): 1—38.

van Dijk, M. 2013. “The social cost of financial crises.” Mimeo. Rotterdam School of Management, Erasmus University.

Frankel, J.A., A.K. Rose. 1996. “Currency crashes in emerging markets: and empirical treatment.” *J. of International Economics* 41(3-4), 351—366.

Goodfellow, I., Y. Bengio, a. Courville. 2016. *Deep Learning.* MIT Press.

Holpainen, M., P. Sarlin. 2017. “Toward robust early-warning models: a horse race, ensembles and model uncertainty.” *Quantitative Finance* 17(12): 1933—1963.

Jiang, H., O. Nachum. 2019. “Identifying and correcting label bias in machine learning.” Google Research.

Kaminsky, G.L., S. Lizondo, C.M. Reinhart. 1998. “Leading indicators of currency crises.” *IMF Staff Papers* 45(1): 1—48.

Klambauer, G., T. Unterthiner, A. Mayr. 2017. “Self-normalizing neural networks.” *31st Conference on Neural Information Processing Systems (NIPS 2017).* Long Beach, California.

Kotsiantis, S., D. Kanellopoulos, P. Pintelas. 2006. “Handling imbalanced datasets: a review.” *GESTS International Transactions on Computer Science and Engineering* 30: 25—36.

Lang, J., T. Peltonen, P. Sarlin. 2018. “A framework for early-warning modeling with an application to banks.” ECB Working Paper 2182 (Frankfurt, European Central Bank).

Laeven, L., F. Valencia. 2013. “Systemic banking crises: a new database.” *IMF Economic Review* 61(2): 225—270

van der Maaten, L., G. Hinton. 2008. “Visualizing Data using t-SNE.” *Journal of Machine Learning Research* 1: 1—48

Medas, P. T. Poghosyan, Y. Xu, J. Farah-Yacoub, K. Gerling. 2018. “Fiscal crises.” *J. of International Money and Finance* 88: 191—207.

de O?a, J., C. Garrido. 2014. “Extracting the contribution of independent variables in neural network models: a new approach to handle instability.” *Neural Computation & Applicactions 25: 859.*

Pascanu, R., G. Montufar, Y. Bengio. 2014. “On the number of response regions of deep feed forward networks with piece-wise linear activations.” *International Conference on Learning Representations.*

Raghunathan, T.E., J.M. Lepkowski, J. vanHoewyk, P.A. Solenberg. 2001. “A Multivariate technique for multiply imputing missing values using a sequence of regression models.” *Survey Methodology* 27(1): 85—95.

Reinhart, C.M., K.S. Rogoff. 2014. “Recovery from financial crises: evidence from 100 episodes.” *American Economic Review* 104(5): 50—55.

Rezende, D.J., Mohamed, S., Wierstra, D., 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. arXiv Prepr. arXiv1401.4082.

Stiglitz, J. 2017. “Where modern macroeconomics went wrong.” NBER Working Paper No. 23795, National Bureau for Economic Research (Cambridge, Massachusetts).

Vincent, P., H. Larochelle, Y. Bengio, P.-A. Manzagol. 2008. “Extracting and composing robust features with denoising autoencoders.” *Proceeding ICML ’08 Proceedings of the 25th international Conference on Machine learning.*

Vincent, P., H. Larochelle, Y. Bengio, P.-A. Manzagol. 2010. “Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion.” *The Journal of Machine Learning Research* 11: 3371-3408.

Xie, J., R. Girshick, A. Farhadi. 2016. “Unsupervised deep embedding for clustering analysis.” *Proceeding ICML’16 Proceedings of the 33rd International Conference on International Conference on Machine Learning* 48: 478—487