Journal of Financial Stability 49 (2020) 100746

Contents lists available at ScienceDirect

Journal of Financial Stability

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

Predicting systemic financial crises with recurrent neural networks

Eero Tölö [a,b,c,*]

[a] Helsinki GSE, University of Helsinki, P.O. Box 17, 00014, Finland
[b] Department of Economics, London School of Economics and Political Science, Houghton Street, WC2A 2AE, London, United Kingdom
[c] Department of Financial Stability, Bank of Finland, P.O. Box 160, FI-00101, Helsinki, Finland

ARTICLE INFO

Article history:
Received 30 August 2019
Received in revised form 21 April 2020
Accepted 22 April 2020
Available online 31 May 2020

JEL classification:
G21
C45
C52

Keywords:
Early warning system
Systemic Banking crises
Neural networks
Validation

ABSTRACT

We consider predicting systemic financial crises one to five years ahead using recurrent neural networks. We evaluate the prediction performance with the Jórda-Schularick-Taylor dataset, which includes the crisis dates and annual macroeconomic series of 17 countries over the period 1870–2016. Previous literature has found that simple neural net architectures are useful and outperform the traditional logistic regression model in predicting systemic financial crises. We show that such predictions can be significantly improved by making use of the Long-Short Term Memory (RNN-LSTM) and the Gated Recurrent Unit (RNN-GRU) neural nets. Behind the success is the recurrent networks' ability to make more robust predictions from the time series data. The results remain robust after extensive sensitivity analysis.

1. Introduction

In their article, Schularick and Taylor (2012) describe two eras of finance capitalism. The latter period started around the mid-1900s and is characterized by unprecedented financial risk and leverage, where credit flows freely, and activist macroeconomic policies react when booms go bust. Following the 2008 financial crisis, proactive financial stability policies have gained ground, as authorities implement such policies to counteract the build-up of systemic risk. For evidence, see the growing list of active policies listed by the European Systemic Risk Board.[1] An example is the countercyclical capital buffer (Basel Committee on Banking Supervision (BCBS), 2011), an additional capital requirement enforced on banks when the authorities assess the credit growth as excessive. Timing and planning of financial stability policies require a timely view of the associated risks. If the authorities decide to curb lending at the wrong time, economic growth can be harmed without a commensurate benefit. Financial crisis prediction models help in timing policies by providing information about the likelihood of a crisis.

In this article, we investigate how well various types of artificial neural networks (ANNs) predict systemic financial crises using time series information. When we say a systemic financial crisis, we essentially mean a systemic banking crisis.[2] We are motivated by the developments in the ability of deep neural nets to handle sequential data that have taken place during the last few decades. From the econometric perspective (see Kuan and White, 2007), ANNs serve as universal function approximations, which avoid the problem of model misspecification, unlike parametric models commonly used in empirical studies. This should be useful for financial crisis prediction because the crises are fundamentally non-linear phenomena. We consider alternative neural net architectures and benchmark them against the logistic regression model, a standard model used in policy institutions at the time of writing (see e.g. Lang et al., 2019). Basic neural net models have often been looked down upon by economists for their lack of interpretability. Thanks to recent advances, drivers of neural net predictions can now be decomposed and understood on par with other econometric models.

* Correspondence to: Helsinki GSE, University of Helsinki, P.O. Box 17, 00014, Finland.
E-mail address: eero.tolo@helsinki.fi

[1] www.esrb.europa.eu.

[2] Following Schularick and Taylor (2012), we define systemic financial crises "as events during which a country's banking sector experiences bank runs, sharp increases in default rates accompanied by large losses of capital that result in public intervention, bankruptcy, or forced merger of financial institutions."

https://doi.org/10.1016/j.jfs.2020.100746
1572-3089/

We postpone the discussion of existing financial crisis prediction literature to the next section. At this point, it suffices to know that these models usually make predictions based on a cross-section of macro-financial variables. The research agrees that neural nets outperform the logistic model in systemic financial crisis prediction in cross-validation. There remains some controversy about whether the benefits carry out to sequential out-of-sample evaluation that only uses information known at the time of forecasting.

We argue that there are gains in considering information in the economic time series beyond the latest cross-section. This should not be surprising since we understand that economic indicators in general exhibit rich dynamics. Moreover, in the context of crisis prediction, we have seen that each early warning indicator shows a distinct time pattern (see for example Drehmann and Juselius, 2014, or Tölö et al., 2018). While there are reasons to avoid multiple time lags in the context of logistic regression (mainly collinearity), for neural nets, the time series dimension presents an untapped source of information, whose potential remains so far mostly uncharted in the context systemic financial crisis prediction.

When it comes to choosing a neural net architecture for a multivariate time series prediction problem, we have some primary candidates. The so-called multilayer perceptron (MLP) is the most basic form of neural net that can produce universal function approximations. Compared to other non-parametric methods, such as Fourier or polynomial expansions, the MLP typically requires a lower number of components and is thus a more parsimonious approach (Barron, 1993). Earlier research has shown that MLPs can give accurate crisis predictions based on a cross-section of observations (Holopainen and Sarlin, 2017; Ristolainen, 2018; Bluwstein et al., 2020). Contrasting evidence is provided by Beutel et al. (2019), who find that the logit model outperforms basic machine learning methods, including the MLP in out-of-sample prediction in a sample of 15 countries. In any case, MLPs can be applied to time series using a time-window approach (Gers et al., 2001).

For sequence problems, there also exists a class of models called recurrent neural networks (RNN) that are designed for modeling temporal dynamic behavior. Binner et al. (2004, 2006) show that simple RNNs produce forecasts comparable to traditional Markov switching models. Since then, the RNNs have evolved to include gating mechanisms that allow them to retain a past event in memory for an extended time. The modern RNNs with Long-Short Term Memory (LSTM; Hochreiter and Schmidhuber, 1997) and Gated Recurrent Units (GRU; Cho et al., 2014) are state-of-the-art techniques for sequence learning. RNN-LSTMs have been used with great success for all kinds of sequence problems: text and speech recognition (e.g., Wu et al., 2016), video recognition, acoustic models, traffic, weather patterns, etc. but much less with economic data.[3] RNN-LSTMs have also been used successfully for demand forecasting and financial market predictions (e.g., Fischer and Krauss, 2018; Borovkova and Tsiamas, 2019; Minami, 2018). Siami-Namini et al. (2018) find RNN-LSTMs outperform ARIMA in time series forecasting. Cook and Smalter Hall (2017) improve forecasts of macroeconomic indicators with an LSTM and other neural nets.

Resolving whether the RNNs help in a systemic financial crisis prediction requires us to take the problem to data. This study uses the Jordá-Schularick-Taylor Macrohistory database (Jordà et al., 2017), which consists of annual time series for 17 economies over the period 1870–2016 and includes 66 systemic banking crises in the first era of finance capitalism (1870–1939) and 24 in the second era (1946–2016). The relatively small amount of data speaks towards less complicated models with fewer parameters. Therefore, we consider MLPs with one hidden layer and non-stacked RNNs. We include a small number of explanatory variables: loans-to-GDP, house prices and stock prices, current account ratio, and real GDP. The dependent variable is a zero/one dummy with 1 indicating a systemic financial crisis at a specific prediction horizon.[4] After establishing that the neural nets produce well-calibrated crisis probabilities, we proceed to test their prediction performance using the Area Under Curve (AUC) statistics.

First, we benchmark the MLP, RNN, and gated RNN neural nets against the logit model in a one-year ahead prediction. We evaluate the models using country-by-country cross-validation and sequential out-of-sample tests in various subsamples motivated by the two eras of finance capitalism and the end of Bretton-Woods. In the cross-validation, we estimate the model for all but one country and evaluate the model for the remaining country, such that we test the model for each country in turn. In the sequential evaluation, we estimate the model for an earlier time-period and test the model for a later time period. We find that the RNN neural nets, especially the gated RNNs, consistently outperform both the MLP neural nets and the logit model in all subsamples and evaluations. We attribute the performance advantage to three sources. First, including multiple lagged predictors (i.e. the time-window) adds useful information that benefits even the logit model in the cross-validation. Second, given the limited amount of data for estimation, the RNNs can make the best use of this information based on their temporal structure. Third, the gating mechanism in gated RNNs helps in extracting information from the time series.

The rest of the analysis focuses on the more recent 1970–2016 subsample. We consider forecast horizons of up to 5 years. For both cross-validation and sequential evaluation, we find that the RNNs, especially the gated RNNs, outperform the MLP and the logit model also at longer forecast horizons. We focus on the LSTM and demonstrate that it produces coherent predictions such that, on average, the predicted probability of crisis peaks at the intended distance to a crisis. Finally, we investigate which explanatory variables are mainly responsible for the performance improvements of the LSTM over the logit model. For this purpose, we analyze the prediction model using subsets of explanatory variables. We show that stock prices actively drive the cross-validated predictions, but also other variables contribute. The sequential predictions are driven more equally by the different predictors.

Overall, the improvement in prediction accuracy from the neural nets is substantial. For example, considering the three-year ahead forecast, when we fix the sensitivity such that the models detect more than 80% of the crises correctly, the LSTM produces less than half the amounts of false alarms in comparison to the logit model (about 20% vs. more than 40%). In the Appendix A, we report a sensitivity analysis. The results are found to be robust for changes in the number of neurons in the hidden units and the length of the time-window.

We contribute to the growing literature of using machine learning methods in crisis prediction (see Section 2 for a review). Our main contribution is to demonstrate that utilizing the time series information via gated RNNs improves the systemic financial crisis predictions. The consistent performance advantage in the sequential evaluation with different subsamples suggests that the gated RNNs are likely to provide out-of-sample predictions that are more robust than generally. Based on our findings, we expect similar techniques to be useful in predicting other related events such as

[3] Of course, some economic information comes in text form, so language models and economics are not mutually exclusive, see Apel et al. (2019) for an example.

[4] Treating each systemic financial crisis as a similar dependent dummy irrespective of the depth of the crisis and other characteristics fades the non-linear characteristics and likely favours the linear logistic regression.

recessions and currency crises. Exploratory results with neural nets could also help in devising better non-linear econometric models.

The rest of the article is organized as follows. Section 2 offers a survey of financial crisis prediction models. Section 3 presents the dataset. Section 4 reviews the neural nets considered in this study, MLPs and RNNs, respectively. Section 5 discusses the performance evaluation framework, which consists of defining the dependent variable for the forecast problem (Section 5.1), the validation and out-of-sample tests (Section 5.2), and performance measurement (Section 5.3). Section 6 presents the results. Section 7 concludes with discussion. Appendix A discusses in detail the neural net models and their estimation, and Appendix B presents the sensitivity analysis.

2. Early warning models for financial crisis prediction

Despite the long history of financial crises, early warning models based on cross-country panel data sets are a relatively new practice that emerged in the late 1990s through pioneering work including Demirgüc-Kunt and Detragiache (1998), (2000), Kaminsky and Reinhart (1999), Hardy and Pazarbasioglu (1999), Caprio and Klingebiel (1997), and Berg and Pattillo (1999). These articles collected datasets of crisis dates and investigated which macro-financial quantities were useful in predicting those events. In those days, the two standard methods were the so-called signaling method and the logit model. The former takes one time series (such as annual credit growth or some other risk measure), and a value that goes beyond a threshold is considered a warning signal. The logit model explains the crisis dummy (or pre-crisis dummy) directly using a set of variables. Numerous articles followed that investigated indicators that precede financial crises.[5] Following the 2008 episode, there was again a spur of interest, and datasets were extended (see Reinhart and Rogoff, 2009; Schularick and Taylor, 2012; Laeven and Valencia, 2012; Babecký et al., 2014; Detken et al., 2014). The findings from this literature are actively used in policy institutions. Tölö et al. (2018) include a convenient summary table of early warning indicators used in a broad set of studies, and whether they were significant in those studies: Numerous studies find that private sector loans and loans/GDP are significant banking crisis predictors. In fact, they are often the most robust predictor, although the result is somewhat dependent on the sample of countries and other characteristics of the dataset. A debt service burden indicator has been available in a smaller number of studies but is generally found to be informative. House prices, stock prices, and credit spreads are also found to be good predictors and have been included in many studies. Current account/GDP has been a significant predictor in many studies, but its performance has been somewhat less consistent. Various other measures of external imbalances tend to be significant about half the time that they have been included in published manuscripts. Variables related to real economy such as GDP, investment, and unemployment are occasionally significant in the prediction models, but never the strongest predictors. Other characteristics such as income inequality, fixed exchange rate, deregulation, and contagion have also been found to play a role.

Recent literature has already moved beyond the logit model; the new approaches include multinomial logistic model and machine learning methods. Caggiano et al. (2014), (2016) and (for currency crises) Bussiere and Fratzscher (2006) show that the multinomial logistic model outperforms the usual binomial logit model. Other articles seek to replace the logit model with machine learning methods such as decision trees, neural nets, random forests, support vector machines, and other classification methods.[6] Early examples are Duttagupta and Cashin (2011); Díaz-Martínez et al. (2011); Davis and Liadze (2011), and Manasse et al. (2013), who all employ classification trees. In these earlier studies, the focus is on identifying nonlinear rules for characterizing pre-crisis developments. For example, Duttagupta and Cashin (2011) find that a specific binomial tree condition that combines low bank profitability with modest export growth significantly increases the probability of a banking crisis in emerging/developing countries.

A fundamental question is whether the machine learning methods can robustly outperform the basic traditional econometric models given the limitations of the data. So far, the logit model has been a standard benchmark. Using a common reference increases comparability because, due to differences in datasets, the attained performance measures can't usually be compared directly with each other. A number of studies find that machine learning methods outperform the logit model: Joy et al. (2017) and Alessi and Detken (2018) use classification trees and random forests, Ristolainen (2018) artificial neural net, Casabianca et al. (2019) adaptive boosting (AdaBoost), Fouliard et al. (2019) model averaging; Holopainen and Sarlin (2017) and Bluwstein et al. (2020) benchmark various machine learning models against the logit model. In these two comparison studies, the MLP neural net performs roughly as well as the top-ranking methods. In contrast to these studies, Beutel et al. (2019) find no improvement for a decision tree, random forest, KNN, SVM, or MLP neural net in comparison to the logit model in sequential out-of-sample prediction. In the analysis of Beutel et al. (2019), the MLP neural net still comes closest to the logit model in terms of performance, and they call for further research with more sophisticated neural nets. Fricke (2017) also finds that various machine learning methods do not bring benefits when compared to the logit model in a sequential evaluation with univariate data.

Related work includes Qi (2001), Gogas et al. (2014), and Nyman and Ormerod (2017), who predict recessions using machine learning, and Fioramanti (2008) and Manasse and Roubini (2009) who predict sovereign debt crises. Suss and Treitel (2019) forecast bank distress in the UK with random forests. Nik et al. (2016) predict financial crises in emerging economies using neural nets.

A dominating practice in the above literature is that while the studies are based on panel data (multivariate time series for each country), the predictions are almost exclusively based on only a cross-section of variables. A few studies do consider the lag structure to a limited extent. While the early work often considered contemporaneous crisis determinants, the analysis soon turned to lagged predictors to avoid simultaneity issues. Importantly, we do not need to worry about endogeneity when we do prediction based on lagged variables, which are not influenced by the predicted event. Demirgüc-Kunt and Detragiache (1998) note that credit growth is significant if lagged by two periods. They also note that lagged GDP growth loses significance indicating that either the contemporaneous GDP causes the banking crisis very quickly or is itself caused by the crisis. Hardy and Pazarbasioglu (1999) note that the lag structure provides a rough indication of the distance to a crisis. Davis and Karim (2008) choose the optimal lag structure based on a grid search and conclude that transforming indicators and using lags and interaction terms improve the per-

[5] The list is long. Some examples are Alessi and Detken, 2011; Barrell et al., 2011; Bordo and Meissner, 2012; Borio and Lowe, 2002; Borio and Drehmann, 2009; Büyükkarabacak and Valev, 2010; Davis and Karim, 2008; Domaç and Martinez Peria, 2003; Drehmann and Juselius, 2014; Lo Duca and Peltonen, 2013; Behn et al., 2013; Jordà et al., 2015; Roy and Kemme, 2012; Kauko, 2012, 2014; Tölö et al., 2018; von Hagen and Ho, 2007.

[6] The multistate approach in Caggiano et al. (2014), (2016) and Bussiere and Fratzscher (2006) is not mutually exclusive with the machine learning methods. An example is Sarlin (2014), who considers four states of financial stability (normal, pre-crisis, crisis, post-crisis) for a visualization application.

formance of the early warning model. Indeed, a common approach in the reviewed literature has been to extract information from the time series through specific transformations such as two, three, or four-year growth, or extracting a cyclical component based on beliefs about the length of the financial cycle (cf. Drehmann et al., 2010; Kauko and Tölö, 2020a, 2020b). If the choice is based on full-sample information, that may have implications for out-of-sample predictions, however. Borio and Drehmann (2009) consider fine-tuning the lag structure of credit and equities such that they peak at the intended distance to the crisis. Schularick and Taylor (2012) and Bordo and Meissner (2012) consider five lags of credit growth and note that the first two lags have opposite signs. Fricke (2017) uses the same univariate five-lag specification in machine learning models. Drehmann and Juselius (2014) compare the performance of the credit-to-GDP gap and the debt-service ratio in banking crisis prediction and conclude that the former dominates at longer horizons and the latter at shorter horizons. Tölö et al. (2018) include a summary table that shows the prediction horizons for which different early-warning indicators were informative in the EU countries.

In summary, the literature has investigated which specific lags of some variables are informative. However, it has not really considered whether the information in the time series as a whole could be utilized in predictions using advanced techniques. An exception is Virtanen et al. (2018), who predict financial crises with unit root tests in an expanding window. Current article seeks to address this gap in the literature.

3. Data

All data for this study come from the Jórda-Schularick-Taylor macro history database (Jordà et al., 2017; and Knoll et al., 2016). The dataset includes 17 countries: Australia, Belgium, Canada, Switzerland, Germany, Denmark, Spain, Finland, France, UK, Italy, Japan, Netherlands, Norway, Portugal, Sweden, and the USA.

The dependent variable that we predict is based on the systemic financial crisis dummy variable, which takes value 1 for the year that marks the start of a systemic financial crisis in a given country. This pre-crisis dummy we discuss in detail in Section 5.1. The dataset classifies systemic financial crises as "events during which a country's banking sector experiences bank runs, sharp increases in default rates accompanied by large losses of capital that result in public intervention, bankruptcy, or forced merger of financial institutions" (Schularick and Taylor, 2012). Financial crises were fairly common in the first era of finance capitalism 1870–1939. The dataset lists 66 crises in this era (though all variables are not available for some of the old crises). The WW2 was followed by a relatively long absence of systemic financial crises until they started again starting from the 70s. The dataset includes 24 financial crises after WW2. The crisis pattern is associated with different economic policy regimes, as discussed more extensively in Schularick and Taylor (2012).

We consider up to five explanatory variables for the systemic financial crisis prediction model. The variables are

1. Loans to non-financial private sector divided by GDP, 1-year growth (*abbr.* l/gdp)
2. Real stock prices, 1-year growth (*abbr.* rsp)
3. Real house prices, 1-year growth (*abbr.* rhp)
4. Current account-to-GDP ratio, level (*abbr.* ca/gdp)
5. Real GDP, 1-year growth (*abbr.* gdp)

Note that we imposed the one-year growth to make the data stationary and comparable between countries. All the variables can potentially contain some relevant information that helps predict systemic banking crises. The five variables were chosen based on their generality and the fact that they describe distinct economic developments. As depicted in Fig. 1, the variables exhibit rich dynamics around systemic financial crises. Table 1 presents descriptive statistics for the variables. Based on the summary table in Tölö et al. (2018), which summarizes many of the early warning indicator studies in the literature, we expect loans/GDP, stock prices, and house prices to be the strongest predictors. Also, the current account-to-GDP ratio and real GDP have been found informative in some studies, but the evidence is not as strong.

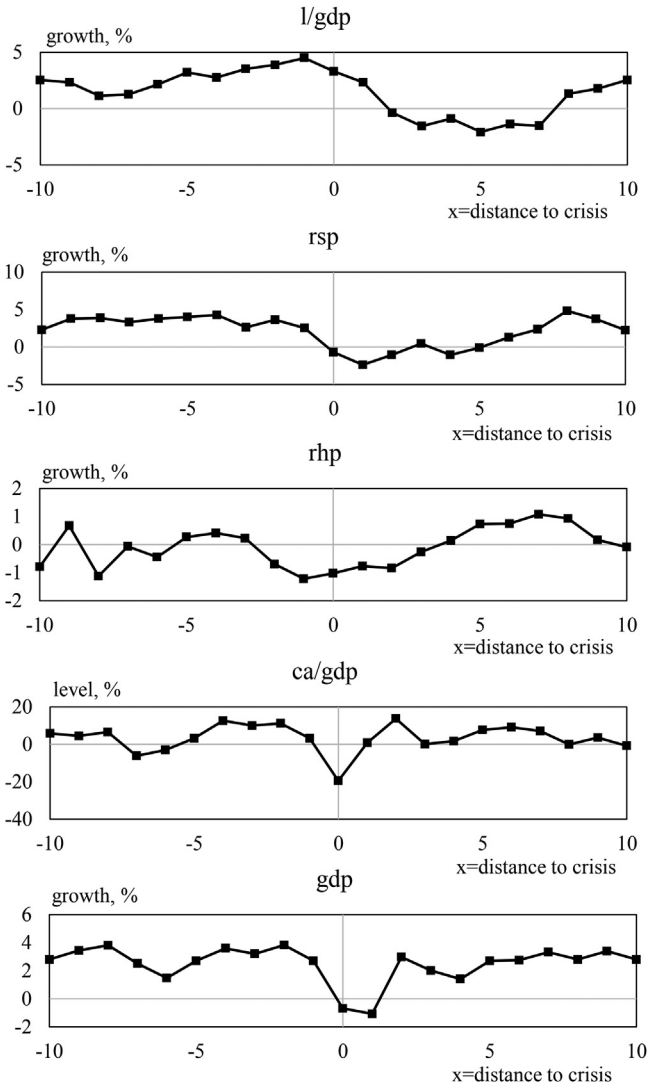


Fig. 1. Development of predictors around systemic financial crisis dates. The horizontal axis shows the distance in years to the systemic financial crisis that starts at x = 0.

In practice, apart from the stock prices, the explanatory variables have some information lags. Studies with quarterly data typically use one quarter publication lag. Like other studies with annual data, we do not introduce publication lags (see e.g. Borio and Lowe, 2002).

4. Neural nets for time series prediction

Artificial neural networks (ANNs), neural nets for short, are a class of nonlinear models that bear a resemblance to the biological neural structure. As discussed in the introduction, ANNs are interesting because they can provide parsimonious non-linear function approximations. ANNs typically consist of layers of nodes that are connected to subsequent layers of nodes through non-linear

Table 1
Descriptive statistics for predictors for the period 1870–2016.

Variable	Mean	Median	Std.	10th percentile	90th percentile	Observations
l/gdp	1.84	1.56	7.08	−5.17	8.63	1542
rsp	2.43	1.86	9.51	−7.26	11.70	1542
rhp	−0.06	−0.05	4.03	−4.36	4.39	1542
ca/gdp	3.84	3.41	19.64	−19.81	27.82	1542
gdp	3.16	3.07	4.48	−1.50	7.63	1542

Units are one-year percentage growth except for ca/gdp, which is a percentage level.

activation functions. In econometric terms, the nodes in the first layer present explanatory variables. The nodes in the middle layer present intermediate transformations of the explanatory variables. The node(s) in the final layer presents the predicted dependent variable. Each function is associated with a set of parameters called weights and biases. Training the neural net means estimating these parameters by minimizing a loss function that depends on the predicted dependent variable and its true values. Because neural nets typically have a large number of parameters and easily achieve close to perfect in-sample fit, it is crucial to estimate the model with one sample and test it with a different sample. In the following, we shortly describe the structure of the neural nets used in this study. Details of the models and their estimation are provided in Appendix A.

A feedforward neural network is the earliest and most straightforward type of artificial neural net. In this model, the explanatory variables are fed into the neural net in the input layer. They are transformed through activations functions as they pass through the net until they reach the output layer. Today, the most common example of a feedforward neural network is a multilayer perceptron (MLP). The MLP is characterized by all nodes in adjacent layers being connected with each other (see Appendix A.1 for details). The fundamental architectural question for the MLP is the number of hidden layers and the number of nodes in each hidden layer. According to a universal approximation theorem for neural nets (Hornik, 1991), every continuous function on a bounded domain can be approximated with an MLP with just one hidden layer. The problem is that the required size for such a network can be impractically large, making the system prone to overfitting. In large scale problems, empirical evidence suggests that depth can be beneficial. However, the systemic financial crisis is not a large-scale problem (in terms of data), and following the literature, we only include one hidden layer in the MLP.

We consider two alternative MLPs: MLP(1) and MLP(τ). MLP(1) makes predictions based on a cross-section of variables at time t. MLP(τ) makes predictions based on τ latest observations. The MLP sees all the predictor data within the time-window simultaneously. This can lead to predictions that do not appreciate the conceptual difference between current and past data, and the predictions might not generalize well outside the sample. In the following, we discuss a class of neural nets that seek to avoid this problem.

Recurrent neural networks (RNNs) are a family of neural nets designed for sequential data such as language processing and time series. An RNN takes in a time series of explanatory variables, which allows the RNN to use its hidden states (akin to memory) dynamically to process a sequence of input data. Importantly, the RNN preserves the temporal ordering of the time series, which can help reduce overfitting in an application where the temporal ordering is relevant. Conceptually, these two properties could make the RNNs ideal for systemic financial crisis prediction. By observing the time series, the RNN would identify when vulnerabilities are building up. The RNN would remember the accumulated level of vulnerability. Later, when a triggering event takes place (such as a domestic slowdown of the economy or an international shock), it would signal the alarm.

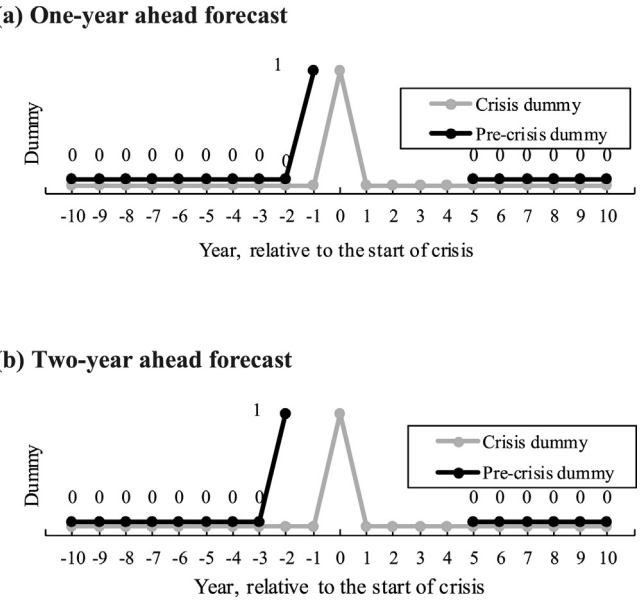


Fig. 2. Illustrations of the pre-crisis dummies around crisis dates.

In this study, we consider three different RNN architectures: basic RNN, RNN with Long-Short Term Memory (LSTM) cells, and RNN with GRU cells. Basic RNNs have few parameters but are sometimes susceptible to an exploding or vanishing gradient problem. LSTMs and GRUs avoid the stability problems of the basic RNN by using gating mechanisms (see Appendices A.3 and A.4 for details). That is why we refer to them together as gated RNNs. The gated RNNs have a great capacity to remember past developments in the time series.

The prediction models are summarized in Table 2. For MLPs, we set the number of nodes in the hidden-layer equal to 10. Similarly, for the RNNs, we set the dimension of the hidden recurrent state to 10. For the time-window MLP and the RNN nets, we use a time-window of 5 years, which corresponds to the five-lag specification used by Schularick and Taylor (2012), Bordo and Meissner (2012), and Fricke (2017). In principle, all these hyperparameters could be optimized using a separate validation. Since we will consider many different neural nets, estimation samples, and test setups, we prefer to calculate all the results with the same reasonable parameter values outlined above. Afterward, we confirm with robustness checks that the specific values of the hyperparameters are not important (see Appendix B).

5. Performance evaluation framework

5.1. Dependent variable

We aim to predict systemic financial crisis events. Hence, the dependent variable is the pre-crisis dummy related to a specific forecast horizon. This is best explained using an illustration. Fig. 2(a) shows the crisis dummy and the pre-crisis dummy for

Table 2
Summary of the prediction models.

Abbreviation	Model name	Hyperparameters	*d*	*k*
Logit(1)	Logistic regression model	none	5	6
Logit(5)	Logistic regression model	none	25	26
MLP(1)	Multilayer perceptron	hidden layers = 1, units in hidden layer = 10, L2 weight = 0.001	5	71
MLP(5)	Multilayer perceptron	hidden layers = 1, units in hidden layer = 10, L2 weight = 0.01	25	271
RNN	Recurrent Neural Network	time steps = 5, dimension of hidden state = 10, L2 weight = 0.001	25	171
RNN-LSTM	Long Short-Term Memory Recurrent Neural Network	time steps = 5, dimension of hidden state = 10, L2 weight = 0.001	25	691
RNN-GRU	Gated Recurrent Unit Recurrent Neural Network	time steps = 5, dimension of hidden state = 10, L2 weight = 0.001	25	521

d = number of explanatory variables.
k = number of parameters.

a one-year ahead forecast around the crisis event. Crisis dummy takes value 1 for the year that marks the start of the systemic financial crisis. The pre-crisis dummy is obtained by shifting the crisis dummy backward in time (according to the forecast horizon) and by removing observations corresponding to crisis and post-crisis periods. Excluding these years is a common approach in the literature and helps alleviate the post-crisis bias (see e.g. Drehmann and Juselius, 2014).

Fig. 2(b) illustrates the crisis dummy and the pre-crisis dummy for a two-year ahead forecast. In this case, in addition to the crisis and post-crisis years, we exclude the one year before the crisis. Conceptually this means that we do not care what the model predicts at the excluded observation. This handling of the pre-crisis and post-crisis periods is largely similar to other sources (see Drehmann and Juselius, 2014; Detken et al., 2014; Holopainen and Sarlin, 2017; or Ristolainen, 2018). As Ristolainen (2018) points out, the pre-crisis dummy is often set equal to one for multiple pre-crisis periods to avoid considering multiple lagged predictors.

5.2. Cross-validation and sequential out-of-sample evaluation

We consider two alternative out-of-sample performance evaluation frameworks: country-by-country cross-validation and sequential evaluation.

5.2.1. Country-by-country cross-validation

In the country-by-country cross-validation, we exclude each country in turn, estimate the network model, and then perform the out-of-sample prediction for each year for the country that was excluded.[7] Then we pool all the out-of-sample predictions together and evaluate the AUC statistics. The country-by-country cross-validation does not fully preserve temporal ordering in the sense that the full information about the other countries is used to estimate the network. This information from other countries is, of course, only limited to the estimation phase. For example, when the neural net makes its prediction for the UK in 2006, it does not explicitly know that a bunch of other countries is going to have a crisis in a few years. Also, because we do not use any information from the predicted country in the estimation, the test can be considered quite robust.

```
Cross-validation algorithm pseudocode:
[1]: Loop C over countries:
[2]:     Estimate the model excluding the data of country C
[3]:     Test the model using data of country C only
[4]:     Store the predicted probabilities for country C.
[5]: end loop
[6]: Calculate the out-of-sample AUC by pooling the predicted probabilities.
```

5.2.2. Sequential out-of-sample evaluation

In the sequential evaluation, we split the sample into two parts. The earlier part is used for estimating the parameters, and the latter part is used for testing. In this case, the temporal structure is fully preserved as we do not use any future information for prediction. We consider four alternative sample splits, as discussed further in the results. Similar sequential evaluation setups are used by Fricke (2017), Beutel et al. (2019), Bluwstein et al. (2020), Holopainen and Sarlin (2017), and Alessi and Detken (2018).

5.3. Performance measurement

We want to evaluate our prediction models using the area under the ROC curve (AUC), which is a common measure used in the systemic financial crisis prediction literature (for further discussion, see Drehmann and Juselius, 2014). As we will see in a moment, computing AUC makes sense because our prediction models output something that can be interpreted as a probability estimate.

Each of the models outputs a value that lies between 0 and 1. If the output is larger than some threshold *h*, then we say that the model predicts a crisis at the given forecast horizon. Otherwise, the prediction is that there won't be a crisis. Correctly predicted crisis is labeled a true positive (TP). Correctly predicted normal state is labeled a true negative (TN). A false alarm is labeled a false positive (FP), and a missed crisis is labeled a false negative (FN).

Sensitivity is defined as TP/(TP + FN), and specificity is defined as TN/(TN + FP). If we plot sensitivity vs. 1-specificity for all possible threshold values *h*, we get the receiver operating characteristic (ROC) curve. The area under the ROC curve is an approximately proper scoring rule for the classification task. Higher AUC corresponds to better predictions. The maximum value of the AUC, one, is achieved for a perfect model that can distinguish the two states perfectly. AUC = 0.5 corresponds to a random guess. AUC below 0.5 means that the predictions are worse than a random guess.

Let us now illustrate that the outputs from the prediction models correspond to probabilities. Earlier, Niculescu-Mizin and Caruana (2005) have demonstrated that simple neural net models produce well-calibrated probabilities, while many other machine learning methods do not.

[7] Country-by-country cross-validation is often replaced with random k-fold cross-validation. Note that k-fold cross-validation is not applicable in our setup because we would inevitably end up having overlapping observations in different folds.

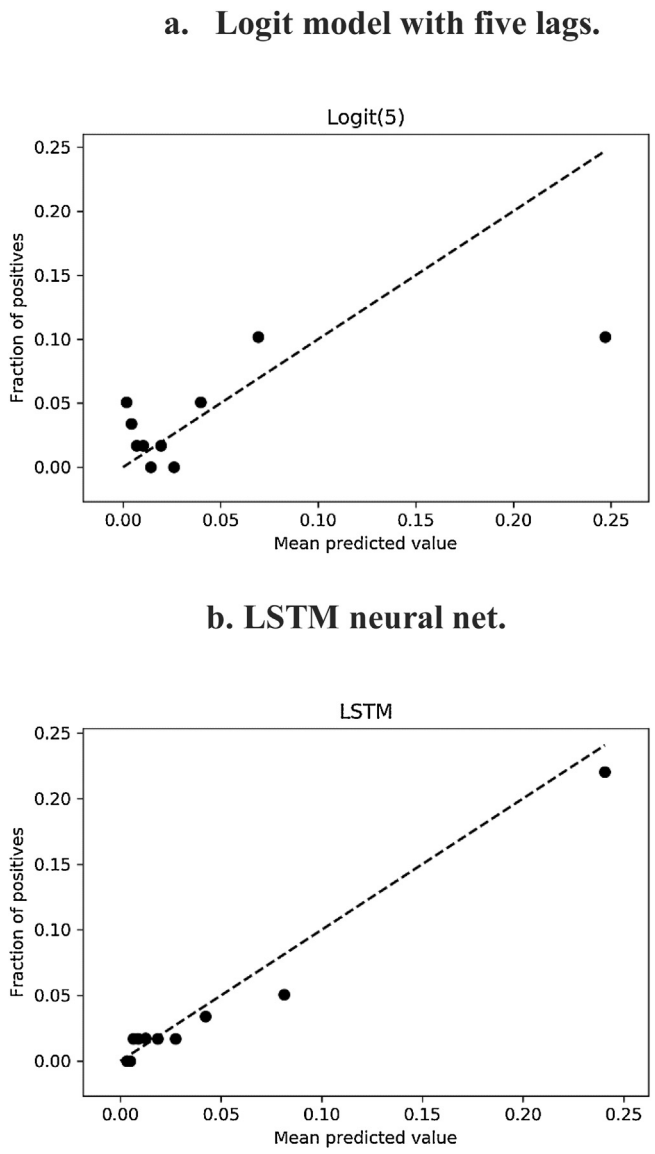


Fig. 3. Reliability diagrams. The horizontal axis shows the mean predicted value for each quantile bucket. The vertical axis shows the fraction of positives (pre-crisis observations) in the corresponding bucket. Both graphs correspond to country-by-country cross-validation in the 1970-2016 subsample. Well-calibrated probabilities should scatter close to the 45-degree line (dashed line).

The relationship between the prediction model output and probabilities can be investigated via reliability diagrams (DeGroot and Fienberg, 1983; Niculescu-Mizin and Caruana, 2005). At first, the sample is ordered according to the predicted probabilities and split into ten bins. To get an equal number of observations in each bin, we use quantile bins.[8] For each bin, we calculate the true fraction of the positive states and plot it as a function of the mean value of the bin. If the probabilities are well-calibrated, the points should scatter around a 45-degree line.

We calculate reliability diagrams for the one-year ahead crisis prediction models using the country-by-country cross-validation for the period 1970–2016. Fig. 3(a) and (b) presents the reliability diagrams for the logit model and the LSTM neural net. The probabilities produced by the LSTM are scattered close to the 45-degree line, so they are well-calibrated. The probabilities produced by the logit model do not seem as reliable, but this is only because the logit model has lower explanatory power.

6. Results

6.1. Benchmark evaluation

In the following, we evaluate the prediction performance of neural net models by benchmarking against a logistic model with a single lag and multiple (5) lags, denoted Logit(1) and Logit(5), respectively. Recall the prediction models summarized in Table 2. In the first stage, we evaluate the models using a one-year forecast horizon and different subsamples. In the second stage, we extend the analysis to forecast horizons of up to five years. In both stages, we first consider performance in cross-validation and then in sequential out-of-sample evaluation.

Recall that in our cross-validation setup, each country, in turn, is used as a test sample while the other countries are used for estimating the model. We consider four alternative subsamples: (1) 1870–2016 (43 crises), (2) 1870–1939 (19 crises), (3) 1946–2016 (24 crises), and (4) 1970–2016 (23 crises). The subsamples are motivated by the first and second era of finance capitalism (Schularick and Taylor, 2012), and the end of the Bretton-Woods era in 1971.

The cross-validated AUC statistics are shown in Table 3(a), where each column corresponds to a different subsample. The obtained AUCs lie between 0.597 and 0.867. The corresponding standard errors adjusted for clustering at the country level range from 0.027 to 0.095. The standard errors are so large partly because the data is annual and partly because we only have one positive pre-crisis dummy for each crisis. Despite the relatively large standard errors, the ranking of the methods tends to hold across subsamples. The first two rows of Table 3(a) show that the logit model with multiple lagged predictors outperforms the logit model with a single lag in all subsamples. Using the test by Delong et al. (1988), we confirm that the differences in AUC are typically statistically significant at the 10% significance level. Thus, in cross-validation, the logit model gains from the information in the additional lagged predictors.

Similarly, the next two rows in Table 3(a) indicate that the time-window MLP(5) outperforms the static MLP(1) except for the short pre-WW2 sample, where the methods have similar performance. The lower rows in Table 3(a) reveal that the gated RNNs are among the top-performing models. They consistently outperform the logit models and the MLPs in all subsamples. The differences in AUC tend to be statistically significant at the 5% significance level. Also, the gated RNNs always outperform the basic RNN. In the cross-validation, all the prediction models tend to perform better in the post-WW2 era. This could be due to the larger number of observations, better data quality, or increased homogeneity among countries through globalization, trade, and development.

We move on to the sequential evaluation, where we again consider four alternative sample splits. In the pre-WW2 era, the sample is naturally split by the WW1. Here we estimate the model using pre-WW1 data and test it using the period between WW1 and WW2. Data during WW1 is excluded, as in Schularick and Taylor (2012). For the second era, we divide the sample into two parts, either at the year 1990 or 2000. Thus, the four alternative subsamples are:

(1) Estimation sample: 1870–1914 (8 crises), Test sample: 1920–1939 (11 crises).
(2) Estimation sample: 1946–1989 (7 crises), Test sample: 1990–2016 (17 crises).

[8] Sometimes the reliability diagrams use equally spaced bins. Since we have one class that is considerably underrepresented and limited amount of data, we use the quantile-based variant of the reliability diagram to avoid bins with too few observations.

Table 3
Performance for one-year ahead crisis prediction in different subsamples.

(a) Country-by-country cross-validation

Model	(1)		(2)		(3)		(4)	
Logit(1)	0.598	(0.044)	0.609	(0.095)	0.621	(0.062)	0.610	(0.058)
Logit(5)	0.662	(0.043)	0.624	(0.056)	0.673	(0.063)	0.642	(0.071)
MLP(1)	0.597	(0.044)	0.705	(0.064)	0.678	(0.046)	0.641	(0.052)
MLP(5)	0.698	(0.046)	0.652	(0.083)	0.710	(0.050)	0.735	(0.056)
RNN	0.736	(0.051)	0.691	(0.058)	0.807	(0.050)	0.782	(0.049)
RNN-LSTM	0.747	(0.035)	0.716	(0.049)	0.867	(0.031)	0.844	(0.039)
RNN-GRU	0.715	(0.039)	0.700	(0.056)	0.866	(0.027)	0.801	(0.039)
Period	1870–2016		1870–1939		1946–2016		1970–2016	
Crises	43		19		24		23	
N	1142		273		869		589	

(b) Sequential out-of-sample evaluation

Model	(1)		(2)		(3)		(4)	
Logit(1)	0.621	(0.111)	0.600	(0.065)	0.524	(0.070)	0.535	(0.068)
Logit(5)	0.656	(0.088)	0.521	(0.072)	0.395	(0.066)	0.374	(0.062)
MLP(1)	0.652	(0.092)	0.545	(0.059)	0.512	(0.072)	0.485	(0.063)
MLP(5)	0.601	(0.100)	0.490	(0.077)	0.493	(0.081)	0.474	(0.074)
RNN	0.701	(0.076)	0.576	(0.060)	0.652	(0.083)	0.651	(0.086)
RNN-LSTM	0.724	(0.077)	0.702	(0.054)	0.726	(0.045)	0.743	(0.055)
RNN-GRU	0.667	(0.095)	0.685	(0.069)	0.645	(0.063)	0.734	(0.066)
Training period	1870–1914		1946–1989		1946–1999		1970–1999	
Crises train	8		7		12		12	
N train	151		512		642		418	
Test period	1920–1939		1990–2016		2000–2016		2000–2016	
Crises test	11		17		12		12	
N test	122		357		227		227	

The numbers in the table are AUC. Inside parentheses are standard errors adjusted for clustering at the country level.
Panels (a) and (b) show the AUC statistics for cross-validation and sequential evaluation, respectively. A higher value is better. Columns correspond to different subsamples. The dependent variable is the pre-crisis dummy defined in Section 5.1. See Table 2 and Appendix A for details of the models. See Appendix A.5 for details of the neural net training. All the models use up to five lags of the same variables: real annual house price growth, real annual stock index growth, annual growth in credit-to-GDP ratio, current account-to-GDP ratio, and annual growth in real GDP.

(3) Estimation sample: 1946–1999 (12 crises), Test sample: 2000–2016 (12 crises).
(4) Estimation sample: 1970–1999 (12 crises), Test sample: 2000–2016 (12 crises).

The AUC statistics for the sequential evaluation are shown in Table 3(b), where the columns again correspond to different subsamples. The AUCs lie in the range of 0.374−0.743, and the standard errors are between 0.045 and 0.111.

In Table 3(b), the performance measures drop overall compared to the cross-validation in Table 3(a), reflecting how hard the sequential out-of-sample prediction is. The RNN based neural nets still perform best across all subsamples. The gated RNNs (LSTM and GRU) are the top performers. In subsamples (3) and (4), the GRU is a bit unlucky in timing the 2008 financial crisis and tends to give signals one year too early. Overall, the performance differences between RNN based models and the other models are clearly statistically significant at conventional significance levels. Even if the performance of the RNN based neural nets deteriorated when we moved to the sequential evaluation, this drop in performance is similar as for the Logit(1) and the MLP(1). In contrast, the Logit(5) and the MLP(5) suffered considerably more. Hence, it seems that the structure of the RNNs is advantageous for out-of-sample predictions at the one-year forecast horizon.

Note that in the sequential evaluation of Table 3(b), the MLP and the logit model are frequently no better than a random guess. We will soon see that the one-year prediction is a particularly tough problem compared to predictions at the longer horizons. The reason is that we are quite strict about the timing and require that the predicted crisis must take place no later than at the specified forecast horizon (see the definition of the pre-crisis dummy in Section 5.1).

This concludes the analysis for the one-year forecast horizon. To sum up, in both cross-validation and sequential evaluation using a one-year forecast horizon, we find that the RNN based models, especially gated RNNs, consistently outperform the usual MLP neural nets and logit models in all subsamples. For practical purposes, policymakers may need predictions with a longer forecast horizon, especially if they plan to put in place policy measures that are intended to slow down the build-up of vulnerabilities.

Next, we will consider prediction performance for forecast horizons extending up to five years. For brevity, we will focus on the 1970−2016 subsample because that is the currently relevant period. Table 4 presents the results for both cross-validation and sequential evaluation. Each column corresponds to one of the five alternative prediction horizons. Additionally, we show the average in-sample AUC for the different cross-validation folds. Comparing the in-sample AUC in Table 4 with the model descriptions in Table 2, we see that a higher number of model parameters correspond to a higher in-sample fit. All the models (even the most parsimonious Logit(1) model) have a considerably higher performance in the estimation sample than in the test sample.[9]

Let us, for a moment, concentrate on the cross-validated results. Looking at the columns in Table 4, we can see that the best performance is found at a forecast horizon of 2–3 years. Except for the Logit(1) and the MLP(1), which are quite sensitive to the forecast horizon, the models perform quite well even at the 4 to 5-year forecast horizons. As previously, the additional information in the time series is demonstrated by the fact that the cross-validated Logit(5) and MLP(5) outperform their single lag counterparts for all prediction horizons. While all neural nets generally outperform the

[9] The in-sample AUCs are comparable to the AUC reported in the in-sample horse race for various financial crisis prediction models in Alessi et al. (2015).

Table 4
Performance for different forecast horizons in the 1970-2016 subsample.

		Forecast horizon (years)									
Model	Evaluation	1		2		3		4		5	
	In-sample	0.689		0.711		0.726		0.762		0.615	
Logit(1)	Cross-validation	0.610	(0.058)	0.655	(0.064)	0.671	(0.061)	0.695	(0.065)	0.411	(0.045)
	Sequential	0.535	(0.068)	0.592	(0.094)	0.716	(0.080)	0.694	(0.073)	0.273	(0.077)
	In-sample	0.843		0.863		0.892		0.897		0.868	
Logit(5)	Cross-validation	0.642	(0.071)	0.689	(0.062)	0.724	(0.051)	0.746	(0.062)	0.722	(0.056)
	Sequential	0.374	(0.062)	0.243	(0.055)	0.251	(0.049)	0.517	(0.084)	0.566	(0.064)
	In-sample	0.842		0.892		0.908		0.815		0.822	
MLP(1)	Cross-validation	0.641	(0.052)	0.743	(0.067)	0.757	(0.050)	0.660	(0.063)	0.593	(0.076)
	Sequential	0.485	(0.063)	0.669	(0.087)	0.802	(0.070)	0.595	(0.081)	0.515	(0.083)
	In-sample	0.959		0.968		0.983		0.978		0.985	
MLP(5)	Cross-validation	0.735	(0.056)	0.776	(0.060)	0.829	(0.037)	0.784	(0.041)	0.793	(0.031)
	Sequential	0.474	(0.074)	0.485	(0.082)	0.715	(0.050)	0.668	(0.085)	0.703	(0.084)
	In-sample	0.920		0.956		0.972		0.978		0.983	
RNN	Cross-validation	0.782	(0.049)	0.780	(0.063)	0.814	(0.041)	0.845	(0.038)	0.806	(0.047)
	Sequential	0.651	(0.086)	0.692	(0.087)	0.769	(0.061)	0.724	(0.065)	0.777	(0.054)
	In-sample	0.998		0.998		0.998		0.997		0.997	
RNN-LSTM	Cross-validation	0.844	(0.039)	0.870	(0.044)	0.873	(0.033)	0.835	(0.037)	0.823	(0.035)
	Sequential	0.743	(0.055)	0.783	(0.064)	0.801	(0.071)	0.751	(0.057)	0.817	(0.065)
	In-sample	0.996		0.994		0.995		0.993		0.990	
RNN-GRU	Cross-validation	0.801	(0.039)	0.863	(0.052)	0.858	(0.032)	0.784	(0.041)	0.782	(0.050)
	Sequential	0.734	(0.066)	0.850	(0.062)	0.818	(0.058)	0.660	(0.060)	0.732	(0.069)
Period		1970–2016		1970–2016		1970–2016		1970–2016		1970–2016	
Crises		23		22		22		22		22	
N		589		566		544		522		500	

The numbers in the table are AUC. Inside parentheses are standard errors adjusted for clustering at the country level.
The table shows the AUC statistics for in-sample, cross-validation, and sequential evaluation. In-sample numbers correspond to average in-sample AUC across the cross-validation splits; hence, no standard error is available for that measure. Columns correspond to different forecast horizons. The dependent variable is the pre-crisis dummy for each forecast horizon defined in Section 5.1. Higher AUC is better. See Table 2 and Appendix A for details of the models. See Appendix A.5 for details of the neural net training. All the models use up to five lags of the same variables: real annual house price growth, real annual stock index growth, annual growth in credit-to-GDP ratio, current account-to-GDP ratio, and annual growth in real GDP.

logit models, the RNN based models have the highest AUC statistics. The RNN-LSTM has overall the most robust performance across prediction horizons. The test for equality of two AUCs by Delong et al. (1988) reveals that the difference to the MLP(5) is statistically significant at the 5% significance level only at the 2-year horizon.

Let us now turn to the sequential evaluation results for different forecast horizons, which are also included in Table 4. Again the predictions are most accurate for 2–3-year forecasts. Similar to Table 3(b), the important findings in this sequential evaluation are the drastic drop of performance for the Logit(5) and the MLP(5) relative to other models, and that the RNN based methods still perform well and generally reach the highest AUC statistics. The Logit(1) and the MLP(1) have their strongest performance at the 3-year forecast horizon. This is because, in this subsample, the explanatory variables often peak 2–3 years before the crisis. In general, carefully tailoring the lag structure or transformation for each prediction horizon would possibly improve model predictions for all the models.[10]

So far, we have demonstrated that RNN based models predict the crises more accurately, as measured by the AUC, than the logit model and the MLP neural nets. Differences in AUC as such are not very tangible, however, so we end this section with some illustrations.

Fig. 4 presents the ROC curves for the LSTM and the Logit(5) model for cross-validation in the 1970–2016 subsample. Let us first focus on Fig. 4(a), which is for the 1-year forecast horizon. The vertical axis is the sensitivity; in other words, the share of correctly predicted crises. The horizontal axis is one minus specificity; in other words, the frequency of false alarms. For sensitivities above 0.5, the ROC curve for the LSTM lies approximately halfway between the vertical axis and the ROC curve for the Logit(5). Hence, for a given level of sensitivity (>0.5), the LSTM produces about half the amount of false alarms compared to Logit(5). For example, at the sensitivity of 80% (vertical axis = 0.8), Logit(5) yields a false alarm 80 percent of the time (horizontal axis = 0.8), while the LSTM yields a false alarm less than 40 percent of the time (horizontal axis<0.4). At low sensitivities (<0.5), the distance between the ROC curve for the LSTM and the vertical axis is about one-half of the gap between the two ROC curves. Hence, for a given level of sensitivity (<0.5), the LSTM produces about 1/3 the number of false alarms compared to Logit(5). In Fig. 4(b), we have similar ROC curves for the 3-year forecast horizon. Here too, the ROC curve of the LSTM lies closer to the vertical axis than the ROC curve of Logit(5). So again, we conclude that the LSTM produces visibly less false alarms for a given level of sensitivity.

Fig. 5 illustrates the average predicted crisis probabilities around systemic financial crises for Logit(1) and the LSTM, again for the one and three-year forecast horizon. This time we choose to plot the static Logit(1) instead of the Logit(5) to illustrate the advantages of the time series predictors. The gray area in Fig. 5 presents the crisis and post-crisis period. Consistent with the high AUC statistics, the LSTM gives, on average, a correctly placed strong signal. The LSTM optimized for 1-year forecasts (black dashed line), starts to signal the crisis relatively late (as it should); a clear peak extends from one year before the crisis to one year after the crisis. In contrast, the logit model (gray dashed line) gives its maximum signal three years before the crisis, and the signal slightly decreases during the following two years. This serves to illustrate that the logit model does not automatically give signals at the intended horizon. Ex-post, the explanatory variables of the Logit(1) model could be lagged to produce the peak at the intended horizon. However, such tweaking is unnecessary with the LSTM.

When optimized for 3-year forecasts, both models (black and gray solid lines in Fig. 5) give their strongest signal three years

[10] The lag structure should then be optimized for each estimation sample separately and tested out of sample. This is outside the scope of present study.

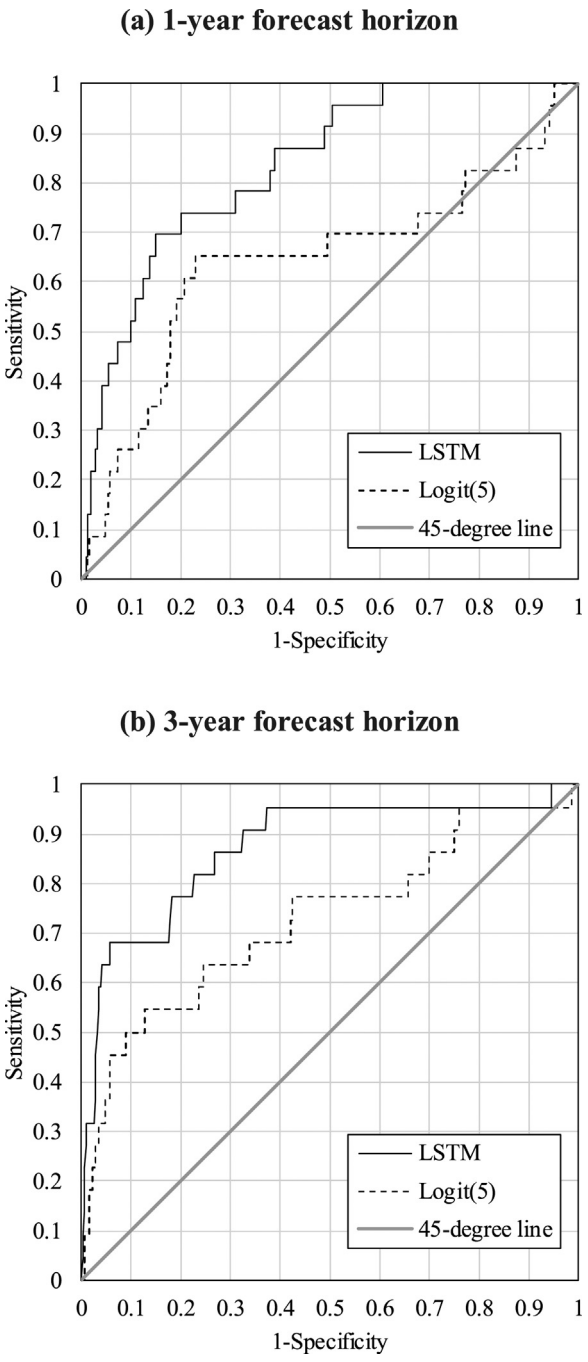


Fig. 4. ROC curves for the five-lag logit model and the LSTM neural net. Panels (a) and (b) show the ROC curves for the cross-validated one and three-year ahead crisis prediction in the 1970-2016 subsample, respectively. The notes of Table 4 apply.

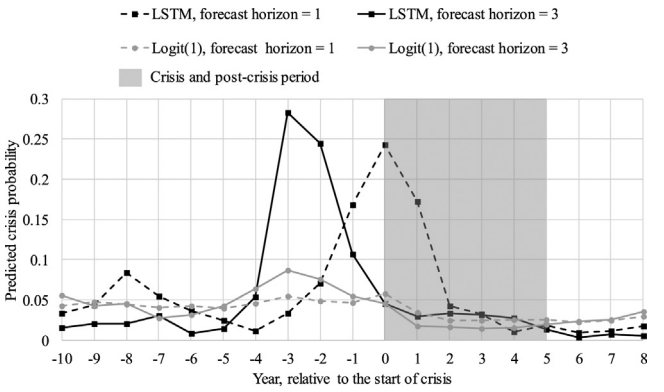


Fig. 5. Average predicted crisis probabilities around crisis dates for the one-lag logit model and the LSTM neural net. The line markers show the average predicted crisis probabilities at different years, relative to the year in which the crisis starts. The dependent variable is the pre-crisis dummy for the forecast horizon specified in the legend of each line. The optimized weights are for the 1970–2016 subsample. The notes of Table 4 apply.

before the crisis, and the signal fades after that. The fading signal is fine, but it has practical implications as the fading of the signal does not necessarily mean fading of the financial instability. It would be important to keep monitoring the probabilities for shorter forecast horizons as well.

6.2. Drivers of RNN predictions

Unlike the logit model, the neural nets include multiple layers of weights. Hence, we can't readily understand what drives their predictions. To determine each variable's contribution to the predictions, we calculate neural net predictions for subsets of input variables. A fundamental choice is whether we use the same neural net for predicting with a subset of predictors or estimate a new neural net. While training a new neural net for each input combination is computationally intensive, it is reasonable for our relatively small neural nets. It also ensures that the included predictors do not underperform due to ignoring interaction with excluded predictors.

We focus on the one and three-year forecast horizons. For brevity, we will focus on the LSTM, which had the most reliable performance in the previous section and compare it to the Logit(5). We train the LSTM neural net for each non-empty subset of the original predictor set {l/gdp, rsp, rhp, ca/gdp gdp}. Neural net hyperparameters remain the same as previously, and estimation is performed just as before.

The results for both the cross-validation and sequential evaluation are shown in Table 5. Each row shows the performance using a subset of predictors marked with x. Let us first discuss the cross-validated results (the left half in Table 5). We discuss the results for the one and three-year forecast horizons in parallel. Starting from the rows 1–5 of Table 5, we have the models with only one predictor. Here, the stock price (row 2) is the most informative predictor for both models and for both prediction horizons. Among the single predictor models, even if the LSTM has larger AUC statistics than the Logit(5) for both forecast horizons, the difference between the models is not statistically significant at conventional significance levels. We see this from the columns that show p-values for the Delong et al. (1988) test for the equality of the two AUCs.

Still discussing the cross-validated results, rows 6–12 in Table 5 show the models with two predictors. Again, the combinations that include the stock price generally perform the best, and the LSTM outperforms the Logit(5) by a statistically significant margin. The combinations of the stock price with other predictors also improve on the stock price alone (row 2). The same story continues to the three-variable models (rows 16–25) and the four-variable models (rows 25–30). The LSTM models that include the stock price perform best and are significantly better than the Logit(5). Hence, stock price seems to be an important variable among the selected predictors for the LSTM for this subsample. However, the LSTM model works without the stock price too. For the one-year forecast horizon, combinations of house price and GDP, and house price, current-account, and GDP also outperform the Logit(5) model by a statistically significant margin.

Table 5
Performance for different variable combinations in the 1970–2016 subsample.

Variables					Cross-validation						Sequential evaluation					
					1-year forecast			3-year forecast			1-year forecast			3-year forecast		
l/gdp	rsp	rhp	ca/gdp	gdp	LSTM	Logit(5)	p	LSTM	Logit(5)	p	LSTM	Logit(5)	p	LSTM	Logit(5)	p
x	–	–	–	–	0.667	0.652	0.482	0.670	0.636	0.167	0.645	0.584	0.183	0.576	0.531	0.511
–	x	–	–	–	0.744	0.674	0.341	0.846	0.788	0.313	0.632	0.449	0.000	0.829	0.460	0.000
–	–	x	–	–	0.694	0.656	0.465	0.655	0.621	0.346	0.664	0.635	0.730	0.712	0.613	0.192
–	–	–	x	–	0.694	0.659	0.601	0.591	0.538	0.551	0.590	0.605	0.862	0.571	0.480	0.474
–	–	–	–	x	0.653	0.550	0.062	0.686	0.627	0.354	0.462	0.324	0.029	0.617	0.570	0.429
x	x	–	–	–	0.807	0.686	0.013	0.891	0.798	0.075	0.695	0.484	0.000	0.785	0.428	0.000
x	–	x	–	–	0.671	0.659	0.794	0.668	0.641	0.715	0.650	0.572	0.307	0.680	0.504	0.044
x	–	–	x	–	0.681	0.654	0.483	0.628	0.604	0.606	0.695	0.569	0.032	0.592	0.534	0.360
x	–	–	–	x	0.684	0.644	0.536	0.711	0.688	0.526	0.625	0.533	0.005	0.562	0.585	0.743
–	x	x	–	–	0.777	0.671	0.094	0.896	0.787	0.054	0.674	0.440	0.000	0.887	0.348	0.000
–	x	–	x	–	0.821	0.698	0.041	0.819	0.784	0.538	0.693	0.503	0.000	0.710	0.389	0.000
–	x	–	–	x	0.792	0.613	0.004	0.871	0.748	0.013	0.582	0.350	0.000	0.869	0.390	0.000
–	–	x	x	–	0.737	0.662	0.177	0.680	0.612	0.225	0.717	0.632	0.376	0.741	0.538	0.027
–	–	x	–	x	0.736	0.603	0.015	0.707	0.699	0.880	0.649	0.423	0.006	0.721	0.638	0.118
–	–	–	x	x	0.660	0.634	0.550	0.569	0.612	0.476	0.593	0.521	0.078	0.663	0.560	0.316
x	x	x	–	–	0.836	0.677	0.011	0.857	0.777	0.054	0.698	0.428	0.000	0.834	0.288	0.000
x	x	–	x	–	0.835	0.693	0.013	0.891	0.786	0.021	0.766	0.474	0.000	0.763	0.385	0.000
x	x	–	–	x	0.810	0.649	0.009	0.876	0.748	0.007	0.695	0.422	0.000	0.752	0.380	0.000
x	–	x	x	–	0.728	0.646	0.068	0.677	0.619	0.464	0.747	0.565	0.017	0.718	0.486	0.000
x	–	x	–	x	0.709	0.633	0.168	0.745	0.697	0.312	0.645	0.439	0.005	0.675	0.503	0.026
x	–	–	x	x	0.674	0.651	0.609	0.682	0.660	0.588	0.701	0.530	0.000	0.537	0.584	0.538
–	x	x	x	–	0.813	0.678	0.013	0.869	0.775	0.075	0.717	0.504	0.002	0.791	0.317	0.000
–	x	x	–	x	0.808	0.622	0.001	0.890	0.753	0.033	0.632	0.303	0.000	0.920	0.329	0.000
–	x	–	x	x	0.804	0.645	0.007	0.849	0.741	0.034	0.672	0.419	0.000	0.692	0.367	0.001
–	–	x	x	x	0.741	0.653	0.043	0.707	0.688	0.773	0.682	0.550	0.161	0.761	0.562	0.022
x	x	x	x	–	0.832	0.673	0.013	0.861	0.763	0.023	0.752	0.426	0.000	0.786	0.271	0.000
x	x	x	–	x	0.831	0.641	0.000	0.878	0.730	0.002	0.688	0.355	0.000	0.843	0.263	0.000
x	x	–	x	x	0.832	0.652	0.007	0.897	0.738	0.002	0.783	0.412	0.000	0.705	0.346	0.000
x	–	x	x	x	0.680	0.646	0.556	0.684	0.676	0.781	0.698	0.481	0.005	0.732	0.468	0.001
–	x	x	x	x	0.829	0.649	0.011	0.897	0.747	0.014	0.707	0.411	0.000	0.810	0.316	0.000
x	x	x	x	x	0.833	0.645	0.004	0.871	0.726	0.008	0.748	0.374	0.000	0.801	0.251	0.000

The numbers in the table are AUC. p-values are for the Delong et al. (1988) test with $H_0 : AUC_1 = AUC_2$, where AUC_1 and AUC_2 are for the LSTM and Logit(5), respectively. p-values below 0.05 indicate that the AUCs are different at the 5% significance level.
The table shows the AUC statistics for the RNN-LSTM and the Logit(5) in cross-validation and sequential evaluation with one and 3-year forecast horizons. Higher AUC is better. Variables marked with x are included in the model specification of the corresponding row. The dependent variable is the pre-crisis dummy for each forecast horizon defined in Section 5.1. See Appendix A.5 for details of the neural net training. l/gdp = real annual house price growth, rsp = real annual stock index growth, rhp = annual growth in credit-to-GDP ratio, ca/gdp = current account-to-GDP ratio, and gdp = annual growth in real GDP.

Overall, we can conclude from Table 5 that the LSTM network outperforms the logistic model so long as we provide it with a sufficiently rich set of predictors.

Let us now briefly discuss the sequential evaluation results shown in the right half of Table 5. In the sequential evaluation, the stock price is relatively less important for the prediction at the 1-year forecast horizon, but still quite important for the prediction at the 3-year forecast horizon. The performance of the LSTM increases as we include more predictors. In contrast, the performance of the Logit(5) suffers as we include more predictors. Hence, the performance differences tend to be highly significant when we have multiple predictors.

The models calculated for Table 5 allow us to conveniently decompose the predictions of the original five variable LSTM into additive explanatory variable contributions, using a so-called Shapley value decomposition (see Lundberg and Lee, 2017; Bluwstein et al., 2020; Shapley, 1953). Fig. 6(a) and (b) present the average of this decomposition around systemic financial crisis dates for the one and three-year forecast horizon, respectively. Based on Fig. 6(a), the LSTM predictions at the one-year forecast horizon are driven on average most strongly by stock prices and then by house prices. The current account contributes strongly to the predicted probability at the crisis year, which is too late in terms of our performance statistics but may have practical implications. Fig. 6(b) shows that stock prices contribute strongly to the predictions at the three-year forecast horizon as well. Loans/GDP start to contribute two years before the crisis. House prices and GDP also help the predictions two to three years before the crisis.

Table 6
Decomposition of AUC for the RNN-LSTM neural net.

	LSTM			
	Cross-validation		Sequential evaluation	
Forecast horizon	1-year	3-year	1-year	3-year
l/gdp	0.040	0.042	0.075	−0.007
rsp	0.152	0.229	0.067	0.167
rhp	0.056	0.043	0.052	0.122
ca/gdp	0.051	0.005	0.072	−0.011
gdp	0.034	0.053	−0.018	0.030

The Shapley value decomposition of AUC is calculated directly from the formula $\phi_{AUC}(k) = \sum_{S \subseteq N-\{k\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}(AUC_{S\cup\{k\}} - AUC_S)$, where $S \subseteq N$ is a subset of the set of five predictors and AUC_S is the AUC achieved by the corresponding neural net. The AUCs are the same as in Table 5 and $AUC_\emptyset = 0.5$. For example, the sum of the first column is 0.040 + 0.152 + 0.056 + 0.051 + 0.034 + 0.5 = 0.833 i.e. it decomposes the AUC of the 5 variable the LSTM into payoff contributions of each variable.

The Shapley value formula can also be used to calculate the contribution of each predictor to the AUC performance measure.[11] Table 6 conveniently summarizes the information in Table 5 using such a decomposition. From Table 6, we see that in the cross-

[11] An alternative measure would be to simply calculate the average AUC. However, we prefer the Shapley formula because it properly summarizes the value-added relative to other predictors.

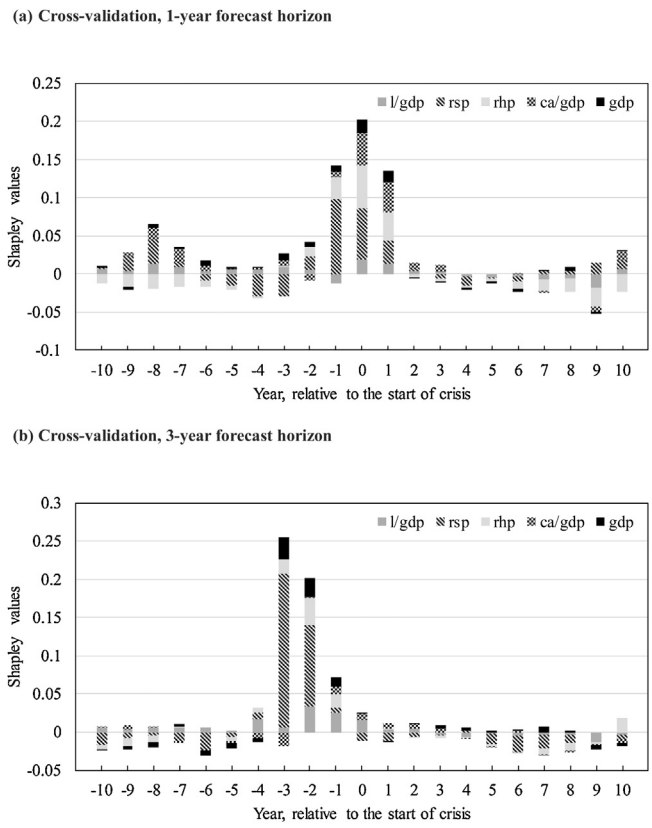


Fig. 6. Average Shapley values around crisis dates for the LSTM neural net. The bars show the average Shapley values of each predictor at different years relative to the year in which the crisis starts. The Shapley values are calculated directly from the formula $\phi(k) = \sum_{S \subseteq N-\{k\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}(f_{S\cup\{k\}} - f_S)$, where $S \subseteq N$ is a subset of the set of five predictors and f_S is the output probability of the corresponding neural net. The optimized weights are for the 1970-2016 subsample, i.e. the same as in Table 5, and notes of Table 5 apply.

validated one-year forecast and three-year forecast, the stock price indeed emerges as the most important predictor, but also all other variables contribute. Only ca/gdp has a negligible contribution at the three-year horizon. For the sequentially evaluated one-year forecast, variables other than the gdp contribute roughly equally. In the corresponding three-year forecast, the stock price and house price are the most important predictors. These results are not surprising in relation to the earlier literature, which (as discussed in Section 2) also finds that, except for gdp, our variables are typically good predictors of banking crises. Nevertheless, it seems that price variables are particularly valuable inputs for the LSTM predictions, which to some extent reduces the role of loans/gdp.

7. Discussion and conclusions

We have investigated systemic financial crisis prediction with neural nets using the Jordá-Schularick-Taylor 1870–2016 dataset. We find that using time series input can lead to more accurate predictions. We also find that the RNNs, especially the gated RNNs (RNN-LSTM and RNN-GRU), outperform the logit model and the MLP neural nets. The results hold for both cross-validation and sequential evaluation across different subsamples. Thanks to their temporal structure, the RNN-based models show good performance in the sequential out-of-sample evaluation. The performance results hold for different forecast horizons of up to five years. We find that the LSTM neural net, on average, produces coherent signals at the intended forecast horizon.

We analyze the drivers of neural net predictions by considering subsamples of input variables. We find that at least two predictors are needed to realize a statistically significant benefit from the LSTM model. Among the considered set of predictors (loans/GDP, stock prices, house prices, current account/GDP, GDP), stock prices are found to be a key driver of the model predictions in cross-validation, but also other variables contribute. In the sequential evaluation, the variables contribute more evenly (except for the GDP, which is found to be less relevant).

Our results relate to the growing literature involved in predicting crises with machine learning and artificial intelligence. The findings for the RNN and the gated RNN neural nets are new since they have (to best of our knowledge) not been considered before in the crisis prediction literature. In general, our findings in cross-validation are consistent with studies that find that the new machine learning methods beat the logit model (Bluwstein et al., 2020; Holopainen and Sarlin, 2017; Ristolainen, 2018). Be that as it may, the logit model remains a useful policy tool for its ease of communication. In sequential evaluation, we do not find a substantial difference between the performance of MLP and the logit model. This is largely consistent with Beutel et al. (2019), who find that the logit model outperforms basic machine learning methods, including the MLP, in the sequential evaluation. In the cross-validation, we find the best forecast accuracy at the 3-year prediction horizon and using six years of time series input in the recurrent neural network. These numbers are consistent with the relatively long length of the financial cycle reported in the literature (8–20 years, see e.g. Borio, 2014 or Filardo et al., 2018). Findings on the minimum number of predictors are consistent with Fricke (2017), who finds that MLP neural nets based on only five lags of credit growth do not bring benefits in comparison to the logit model. The set of predictors that are found to be informative is largely consistent with the earlier literature. The findings for the contemporaneous informativeness of the current account are consistent with Davis and Karim (2008), who note that trade shocks play little part in the build-up of systemic risk, but a sudden deterioration in terms of trade could precipitate a crisis. Similarly, Bordo and Meissner (2012) found that the current account deficit bears no significant relationship with credit growth. However, due to endogeneity associated with simultaneous predictions, we cannot say whether the change in the current account is a cause or itself caused by the financial crisis. The issue of endogeneity is also important in the future if central banks or other financial stability authorities begin to exert influence on macro variables in anticipation of a crisis. Assessing whether a policy-induced reduction in risk indicators decreases the likelihood of a crisis is outside the scope of common financial crisis prediction models. Analysis of this important issue will require structural models and carefully controlled empirical studies.

In future work, a straightforward extension to ours would be to consider systemic financial crises prediction with RNNs using alternative predictors and datasets with quarterly or monthly frequency. Like Bussiere and Fratzscher (2006) and Caggiano et al. (2014), (2016), one could consider more than one class in the prediction task or a continuous dependent variable. The non-linearities associated with crises are expected to play a relatively larger role with a continuous dependent variable. One could also consider other types of crises or events, ranging from currency crises to recessions.

Acknowledgments

The author would like to thank Esa Jokivuolle, Helinä Laakkonen, Antti Ripatti, Matti Virén, Milan Vojnovic, seminar participants at the Helsinki GSE Time Series Econometrics Seminar, and the anonymous referees for useful comments and suggestions. The research

was primarily conducted at the LSE in 2019 and supported by a HYMY travel grant from the University of Helsinki. The views expressed in this paper are those of the author and do not necessarily reflect the views of the Bank of Finland or the Eurosystem.

Appendix A. Neural net models

A.1 MLP

A multilayer perceptron (MLP) consists of three or more dense layers (see Fig. A1): an input layer, one or more hidden layers, and an output layer. "Dense layer" means that there is a connection between each node in successive layers. We consider two alternative MLPs, the MLP(1) and the MLP(τ). The MLP(1) makes predictions based on a cross-section of variables at time s. (We use s to denote the true time of the time series because t is conventionally reserved for the time-steps.) An MLP with one hidden layer can be defined recursively as:

$$\boldsymbol{h}(\boldsymbol{X}) = a_{relu}(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{b}), \tag{A1}$$

$$o(\boldsymbol{h}) = a_{sigmoid}(\boldsymbol{V}\boldsymbol{h} + \boldsymbol{c}), \tag{A2}$$

where $a(.)$ are activation functions (applied elementwise). If we use a time-window of length τ, then $\boldsymbol{X}$ is the $d\tau$-dimensional input vector (d is the number of input time series and τ is the length of the time window). More precisely, for the prediction at time s, $\boldsymbol{X}$ is actually $vec([X_s X_{s-1} \ldots X_{s-\tau+1}])$, where X_s is the set of predictors at time s, and vec operator stacks the lagged values of the predictor into a one long column vector. The $\boldsymbol{h}$ is h-dimensional hidden layer vector, $o(.) \in [0, 1]$ is the output, $\boldsymbol{W}$ and $\boldsymbol{V}$ are $h \times d\tau$ and $h \times 1$-dimensional weight matrices, respectively, and $\boldsymbol{b}$ and $\boldsymbol{c}$ are h and one-dimensional bias vectors, respectively. We apply rectified-linear [relu, $a(x) = \max(0, x)$] activation function at the hidden nodes and sigmoid activation [$a(x) = 1/(1 + \exp(-x))$] at the single output node.

In our main results, the number of explanatory time series $d = 5$, the number of nodes in the hidden layer is h = 10, and the length of the time-window is $\tau = 1$ or $\tau = 5$. The latter corresponds to the number of lags used in Schularick and Taylor (2012) and Fricke (2017).

A.2 Basic RNN

In a basic RNN presented in Fig. A2, there is a hidden recurrent state $\boldsymbol{h}_t$ of dimensionality h, which evolves through time steps t = 1,2,... τ. (Note that the index of the time series is s = 1, 2, ..., T). As previously τ corresponds to the window-length (say, five years), which is typically much less than the length of the whole sample T. The evolution of $\boldsymbol{h}_t$ depends on the previous hidden state $\boldsymbol{h}_{t-1}$ (if any) and the current explanatory variable $\boldsymbol{X}_{s-\tau+t}$. For ease of notation, let us consider the prediction at time $s = \tau$ such that $\boldsymbol{X}_{s-\tau+t} = \boldsymbol{X}_t$. At the final time step, the hidden state $\boldsymbol{h}_\tau$ is mapped to output prediction via a sigmoid function. The basic RNN can be defined recursively as

$$\boldsymbol{h}_t(\boldsymbol{h}_{t-1}, \boldsymbol{X}_t) = a_{tanh}\left(\boldsymbol{W}^{(t)}\boldsymbol{X}_t + \boldsymbol{U}^{(t)}\boldsymbol{h}_{t-1} + \boldsymbol{b}^{(t)}\right),\ t = 1,\ 2,\ 3,\ \ldots,\ \tau \tag{A3}$$

$$o(\boldsymbol{h}_\tau) = a_{sigmoid}(\boldsymbol{V}\boldsymbol{h}_\tau + \boldsymbol{c}). \tag{A4}$$

Here $a(.)$ are activation functions, $\boldsymbol{X}_t$ is the d-dimensional input vector of explanatory variables at the time step t, $\boldsymbol{h}_t$ is the h-dimensional recurrent state vector, $o(.) \in [0, 1]$ is the output (assumed to be one-dimensional in the binary classification task), $\boldsymbol{W}^{(t)}$, $\boldsymbol{U}^{(t)}$, and $\boldsymbol{V}$ are $h \times d$, $h \times h$ and $h \times 1$-dimensional weight matrices, respectively, and $\boldsymbol{b}^{(t)}$ and $\boldsymbol{c}$ are h and one-dimensional bias vectors, respectively. Usually, RNNs assume time-invariance[12] such that $\boldsymbol{W}^{(t)} = \boldsymbol{W}$, and $\boldsymbol{U}^{(t)} = \boldsymbol{U}$, and $\boldsymbol{b}^{(t)} = \boldsymbol{b}$. We assume time-invariance for the RNNs. In Appendix B, we verify that relaxing this assumption does not improve performance. We apply hyperbolic tangent activation function at the hidden nodes and sigmoid activation at the single output node (they correspond to the default setting in Keras).

In our main results, we consider a 5-dimensional time series of length $\tau = 5$, and recurrent state dimension $h = 10$. A sensitivity analysis is provided afterward. It should be noted that due to repeated multiplication of the hidden state by the same $\boldsymbol{U}$, estimating the basic RNN is susceptible to the problem of vanishing or exploding gradient when the number of time steps is large. RNN-LSTMs were originally developed to deal with this problem.

A.3 RNN-LSTM

The LSTM (Long Short Term Memory) Recurrent Network was proposed by Hochreiter and Schmidhuber (1997), and it has turned out to be quite popular. The idea is to make the recurrence going from $\boldsymbol{h}_t$ to $\boldsymbol{h}_{t+1}$ more subtle such that the network can accurately control what information propagates from one time step to another. Again, we consider the prediction at time $s = \tau$ such that $\boldsymbol{X}_{s-\tau+t} = \boldsymbol{X}_t$. To visualize the LSTM net, think of each hidden node in Fig. A2 being replaced by an LSTM cell depicted in Fig. A3.[13] The hidden state is now composed of two components: the recurrent hidden state $\boldsymbol{h}_t$ and the cell state $\boldsymbol{s}_t$, which both have dimensionality h. There is some optionality on how these two states are mapped to the output layer(s), as we will discuss momentarily.

Another new concept is the gating units σ, which are elementwise sigmoid functions that control the flow of information at points x in Fig. A3 (here x denotes Hadamard product $\odot$ i.e. elementwise multiplication). To understand the operation of the LSTM cell, let us start from the lower-left corner of Fig. A3. *Forget gate:* At time step t, the previous recurrent state and the input $(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$ first feed into the forget gate. The forget gate outputs a vector of numbers $\boldsymbol{f}_t$ that lie in the interval [0,1]. At point x (above the forget gate in Fig. A3), we take a Hadamard product of this vector and the previous cell state $\boldsymbol{f}_t \odot \boldsymbol{s}_{t-1}$. In other words, the forget gate controls what information from the previous cell state is retained. The corresponding model equation reads

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}^f \boldsymbol{x}_t + \boldsymbol{U}^f \boldsymbol{h}_{t-1} + \boldsymbol{b}^f), \tag{A5}$$

where the $\boldsymbol{W}^f$ is $h \times d$-dimensional weight matrix, the $\boldsymbol{U}^f$ is $h \times h$-dimensional weight matrix, and the $\boldsymbol{b}^f$ is a h-dimensional bias vector.

Input gate: The input gate uses the information in $(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$ to control what information from the $(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$ themselves is stored to the cell state $\boldsymbol{s}_t$. The elementwise sum operation (at the center-top of Fig. A3), then combines the information that the forget gate retained from the previous cell state and the information that the input gate picked from the new input data and previous recurrent hidden state. The corresponding model equations read

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}^i \boldsymbol{x}_t + \boldsymbol{U}^i \boldsymbol{h}_{t-1} + \boldsymbol{b}^i), \tag{A6}$$

$$\boldsymbol{s}_t = \boldsymbol{f}_t \odot \boldsymbol{s}_{t-1} + \boldsymbol{i}_t \odot a_{tanh}(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{b}), \tag{A7}$$

where the dimensions of $\boldsymbol{W}$, $\boldsymbol{U}$, and $\boldsymbol{b}$ (with and with superscripts) are the same as those in Equation (5).

[12] This is called parameter sharing in the neural net literature.

[13] There exists other variants of the LSTM such as a peephole LSTM (Gers et al., 2002).

Output gate: Finally, the output gate controls, again based on $(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$, to what extent the value in the new cell state is used to compute the new recurrent hidden state $\boldsymbol{h}_t$. The equations are

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}^o \boldsymbol{x}_t + \boldsymbol{U}^o \boldsymbol{h}_{t-1} + \boldsymbol{b}^o), \tag{A8}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot a_{tanh}(\boldsymbol{s}_t), \tag{A9}$$

where the dimensions of $\boldsymbol{W}^o$, $\boldsymbol{U}^o$, and $\boldsymbol{b}^o$ are the same as those in Equations ((5)–(7)).

The output of the LSTM: The final cell of the LSTM outputs a pair of vectors $(\boldsymbol{h}_\tau, \boldsymbol{s}_\tau)$. Because $\boldsymbol{h}_\tau$ already depends on $\boldsymbol{s}_t$ through Equation (9), the default approach is to discard the final cell state $\boldsymbol{s}_\tau$, and only use information in $\boldsymbol{h}_\tau$. However, there is no guarantee that discarding the cell state is optimal. Hence, while we discard $\boldsymbol{s}_\tau$ in the main results, we report the performance for the neural net that retains $\boldsymbol{s}_\tau$ in Appendix B. The output $(\boldsymbol{h}_\tau)$ [or $(\boldsymbol{h}_\tau, \boldsymbol{s}_\tau)$ in Appendix B] is connected to a single output unit using sigmoid activation (similarly as Eq. (A4)). The sigmoid and hyperbolic tangent activations in the recurrent layers correspond to default values in Keras.

In our main results, we consider inputs of dimension $d = 5$, time steps $\tau = 5$, and recurrent state dimension $h = 10$. A sensitivity analysis is provided afterward.

A.4 RNN-GRU

GRU is a gating mechanism proposed by Cho et al. (2014) with a similar purpose as the LSTM. It has only two gates - a reset gate and an update gate - and a single vector presents the hidden recurrent state. It has somewhat fewer parameters than the LSTM so that it can be computationally more efficient. While LSTM cells can do more complex tasks than GRU cells, GRUs have been shown to exhibit better performance in some relatively small datasets. Thus, in crisis prediction task, the GRU neural net could perform well for the same reasons as the LSTM.

In the GRU (see Fig. A4), the gates also have sigmoid activation and take $(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$ as the input. We denote by $\boldsymbol{u}_t$ and $\boldsymbol{r}_t$ the results from the update and reset gate. In the following, the $\boldsymbol{W}$s (with or without superscripts) are $h \times d$-dimensional weight matrices, the $\boldsymbol{U}$s (similarly) are $h \times h$-dimensional weight matrices, and the $\boldsymbol{b}$s (similarly) are h-dimensional bias vectors. The update controls to what extent information from the past is passed to the future. The gates have similar sigmoid structure as in the LSTM:

$$\boldsymbol{u}_t = \sigma(\boldsymbol{W}^u \boldsymbol{x}_t + \boldsymbol{U}^u \boldsymbol{h}_{t-1} + \boldsymbol{b}^u). \tag{A10}$$

The reset gate further helps in deciding what of the past information is forgotten. The formula is similar to the update gate, but the gate has different weights and connects differently to parts of the GRU cell.

$$\boldsymbol{r}_t = \sigma(\boldsymbol{W}^r \boldsymbol{x}_t + \boldsymbol{U}^r \boldsymbol{h}_{t-1} + \boldsymbol{b}^r) \tag{A11}$$

First, the reset gate is used to calculate an intermediate memory state $\boldsymbol{h'}_t$:

$$\boldsymbol{h'}_t = a_{tanh}(\boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{U}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}). \tag{A12}$$

Then we use the output of update gate, to produce the next recurrent state as a convex combination of the previous recurrent state and the intermediate memory state:

$$\boldsymbol{h}_t = \boldsymbol{u}_t \odot \boldsymbol{h}_{t-1} + (1 - \boldsymbol{u}_t) \odot \boldsymbol{h'}_t. \tag{A13}$$

Like RNNs and LSTMs, the hidden recurrent state from the last GRU cell feeds into a single unit dense output layer with sigmoid activation (similarly as Eq. (A4)). The sigmoid and hyperbolic tangent activations in the recurrent layers correspond to the default values of Keras at the time of writing. We consider 5-dimensional times series of length 5, hence $d = 5$ and $\tau = 5$. Like with other models, we set the recurrent state dimension $h = 10$. A sensitivity analysis is provided afterward.

A.5 Estimation of neural net parameters

The estimation of neural net parameters is normally called training the neural net. In the following, we review some neural net training concepts and summarize our training setup.

We train the neural nets by minimizing a loss function with Adam, an adaptive variant of the stochastic gradient descent algorithm. Training means optimizing the weights and biases such that the loss function is minimized based on a training data set. The prediction model is subsequently tested out-of-sample. For loss function, we use the cross-entropy given by

$$L\left(\{y_i, \hat{y}_i\}_{i=1}^N\right) = \sum\nolimits_{i=1}^N l\left(y_i, \hat{y}_i\right)$$

where the components are given by (A14)

$$l(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y}), \tag{A15}$$

where y is the outcome (1 for a pre-crisis period and 0 for a normal period) in the training data, $\hat{y}$ is the neural-net prediction, and N is the number of training observations.

As a typically non-convex optimization problem, training a neural net with a large number of weights is not trivial. Two optimization runs with different algorithms or different initial values hardly ever converge to the same local minimum. We overcome this issue by training the network several times, starting from random initial weights, and taking an average prediction. Of course, it is still important to choose suitable parameters for the optimization algorithm as in any other optimization problem. We experiment with different training parameters using cross-validation in the 1970–2016 subsample, as explained below.[14] We prefer parameters that lead to rapid convergence (to save computer resources) and high validated AUC statistics.

Key parameters in gradient descent based algorithms are the *learning rate*, the *batch size*, and the number of *epochs*. An epoch refers to one cycle through the full training dataset. The learning rate controls how much the neural weights change in each step of the gradient descent. The batch size is the number of observations used in each step of the gradient descent algorithm. Too large learning rate leads to non-convergence, while too small learning rate is computationally expensive and may cause the algorithm to converge too hastily to a local minimum. The batch size controls the randomness in training. Smaller batch size leads to more random paths for the parameters, which may help the algorithm to cover larger parameter space and find a better minimum. We considered learning rates in multiples of 10 (0.1, 0.01,0.001, . . .) and batch sizes in multiples of 2 (16, 32, 64, . . .). We found the learning rate 0.01 and the batch size 16 to be a suitable combination for all the neural nets, although the batch size had little effect on the results. Somewhat lower learning rates also work, but the training would take a longer time. Note that we shuffle the data after every epoch, so each epoch uses different random batches.

To avoid overfitting, we use L2 regularization, which adds a penalty term proportional to the sum of the squared weights to the cross-entropy loss function, $\lambda \sum_j w_j^2$. We apply the regularization to weight matrices in all the dense layers (including the output layers) and the weight matrices in the RNN, LSTM, and GRU cells. We don't apply the regularization to the bias vectors, however. The

[14] Because we use a coarse grid for the training parameters, we expect that we would have ended-up with the same parameters had we used the full-sample.

strength of the regularization is controlled by the parameter λ. We consider values of λ in multiples of 10 (1, 0.1, 0.01,0.001, ...). The time-window MLP is most sensitive to this parameter, and the preferred value is $\lambda = 0.01$. For other neural nets, we use $\lambda = 0.001$, although smaller values seem to work equally well.

The training time can impact the out-of-sample prediction performance of a neural net. If we train too little, the model performs poorly both in-sample and out-of-sample. If we train too much, the out-of-sample performance can (despite L2 regularization) start to deteriorate due to overfitting. To make sure that the neural nets are trained on par with each other, we apply a variant of so-called early-stopping algorithms adapted from Hansen et al. (1997). We train an ensemble of neural nets independently several times and stop at a time that is optimal for the average prediction made by the neural nets in the validation sample. Hence, the stopping time may not be optimal for the individual neural nets, but it is optimal for the ensemble of neural nets conditional on them being trained equal number of iterations. We set the maximum number of epochs to 100, which is enough for all our neural nets (results are largely unchanged if we set maximum epochs to only 20). The size of the ensemble is limited by computational resources because we need to train each neural net for each subsample and forecast horizon. To control the stochastic variation, we train a fifty-neural-net ensemble in the sequential evaluation. In cross-validation, we train an ensemble of 5 neural nets for each validation split in the main results and one in the driver analysis. For the cross-validation, the ensemble is largely unnecessary because we train an independent neural net for each validation split anyway; besides, the corresponding size of the validation sample is much larger than in the sequential evaluation. With such ensembles, the stochastic variation is already very small compared to the standard errors of our performance statistics.

In practice, we implement the neural nets and train them with Keras using a Tensorflow backend. For hardware, we use both Google Colab cloud service GPU runtime and a desktop computer with Nvidia GeForce GTX 1070. For example, training the LSTM neural net 50 times for 100 epochs using the 1970−1999 subsample takes approximately 20 minutes.

Appendix B. Sensitivity analysis and other robustness checks

In the following, we do a sensitivity analysis of the neural net parameters. Among the parameters, we consider the number of units in the hidden layer of the MLPs, the dimension of the hidden state in the RNNs, and the length of the time-window. We also verify that the time-invariance property of the RNNs is helpful. We consider both cross-validation and sequential evaluation. For brevity, we focus on the 1970−2016 subsample.

The number of units in the hidden layer or hidden state controls the complexity of the neural nets. Fig. B1(a) shows the AUC statistics for cross-validation as a function of the number of units (in the hidden layer for the MLP and in the hidden state in the RNNs). Consistently with the main text, the LSTM and the GRU rank at the top as long as there are at least five units in the hidden state. The third and fourth place generally goes to the basic RNN and the MLP(5), while the MLP(1) and the logit models have the weakest performance. Fig. B1(b) shows the same graph for the sequential evaluation. Again, the LSTM and the GRU generally perform the best. The RNN ranks third. The MLPs have poor performance in the sequential evaluation irrespective of the number of units. Hence, the results presented in the main article are robust for using a different number of units in the neural nets.

The length of the time-window controls how long periods of the time series the models use for each prediction. Fig. B2 presents the prediction performance (AUC) as a function of the length of the time-window for the neural net input data. For RNNs, the length of the time-window is usually called the number of time-steps. Panels (a) and (b) show the cross-validated and sequential AUC statistics, respectively. In both panels (a) and (b), we see that the optimal time-window is longer than 2. Also, the LSTM and the GRU generally rank the highest, followed by RNN, MLP, and then the logistic model. This confirms that the results in the main article do not depend on the specific choice of the time-window. Generally, the optimal window length seems to be six years for the cross-validation and four years for the sequential evaluation. With the MLP, long windows seem to result in overfitting. For the RNN based models, the longer window does not lead to overfitting. Thanks to the time-invariance, the longer window does not lead to a larger number of parameters in the RNNs.

Now, we consider relaxing the time-invariance assumption in the RNNs. Recall that in the RNN, we assume that $\boldsymbol{W}^{(t)} = \boldsymbol{W}$, and $\boldsymbol{U}^{(t)} = \boldsymbol{U}$, and $\boldsymbol{b}^{(t)} = \boldsymbol{b}$, for each time step $t = 1, \ldots, \tau$. Now, we relax this assumption and optimize each $\boldsymbol{W}^{(t)}$, $\boldsymbol{U}^{(t)}$, and $\boldsymbol{b}^{(t)}$ separately. This introduces many more parameters to the neural net, however. As an alternative, we consider partial time-invariance whereby $\boldsymbol{W}^{(t)} = \boldsymbol{W}$, and $\boldsymbol{U}^{(t)} = \boldsymbol{U}$, and $\boldsymbol{b}^{(t)} = \boldsymbol{b}$, for each time step $t = 1, \ldots, \tau - 1$, and $\boldsymbol{W}^{(\tau)}$, $\boldsymbol{U}^{(\tau)}$, and $\boldsymbol{b}^{(\tau)}$ are optimized separately. The motivation is that the additional parameters could be most valuable at the last time step just before the RNN outputs the prediction. We set $\tau = 5$ as in the main text.

Table B1 shows the prediction performance (AUC) for RNNs with different time-invariance assumptions. Panels (a) and (b) show

Table B1
Impact of alternative RNN assumptions.

Model	Time invariance	Cross-validation		Sequential evaluation	
RNN	yes	0.782	(0.049)	0.651	(0.086)
RNN	no	0.764	(0.056)	0.557	(0.083)
RNN	partial	0.782	(0.054)	0.645	(0.068)
RNN-LSTM	yes	0.844	(0.039)	0.743	(0.055)
RNN-LSTM+	yes	0.833	(0.046)	0.655	(0.062)
RNN-LSTM	no	0.756	(0.051)	0.526	(0.065)
RNN-LSTM	partial	0.801	(0.053)	0.708	(0.060)
RNN-GRU	yes	0.801	(0.039)	0.734	(0.066)
RNN-GRU	no	0.768	(0.054)	0.508	(0.079)
RNN-GRU	partial	0.767	(0.054)	0.705	(0.058)
Period		1970–2016		1970–2016	
N		589		418 + 227	

The numbers in the table are AUC. Inside parentheses are standard errors adjusted for clustering at the country level.
The table shows the AUC statistics for the RNN neural nets using different time-invariance assumptions and, in the case of LSTM, the use of the final cell state (LSTM+). The columns correspond to cross-validation and sequential evaluation with a one-year forecast horizon in the 1970−2016 sample. The dependent variable is the pre-crisis dummy for each forecast horizon defined in Section 5.1. Higher AUC is better. See Appendix A.5 for details of the neural net training. l/gdp = real annual house price growth, rsp = real annual stock index growth, rhp = annual growth in credit-to-GDP ratio, ca/gdp = current account-to-GDP ratio, and gdp = annual growth in real GDP.

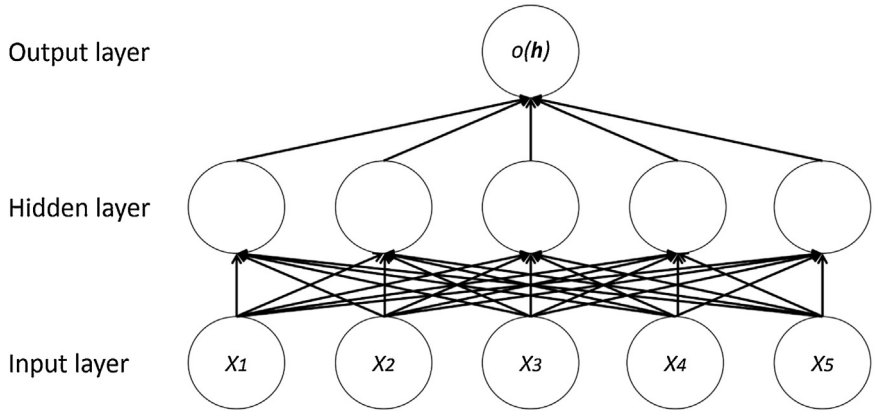


Fig. A1. A perceptron with one hidden layer.

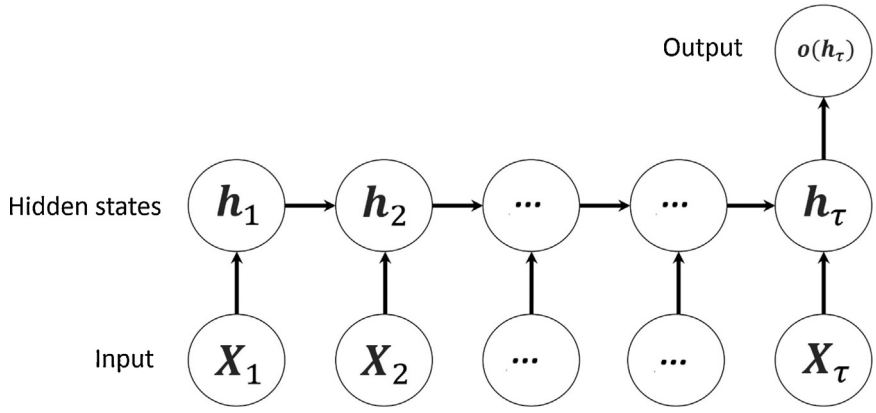


Fig. A2. A basic recurrent neural network.

the cross-validated and sequential results, respectively, using a one-year forecast horizon. We see that the time-invariant RNN models consistently outperform their more complex counterparts. The performance differences are larger in the sequential out-of-sample evaluation than in the cross-validation. In other words, we have found that the assumption of time-invariance helps make more robust predictions. Row 5, starting with "LSTM+. . ." shows the results for an LSTM neural net that additionally uses the final cell state output. As anticipated in the main text, using the final cell state output does not improve the result. Hence, the information in the final cell state is sufficiently captured in the other output state of the LSTM, which obtains the relevant information from the final cell state via the output gate (see Appendix A.3).

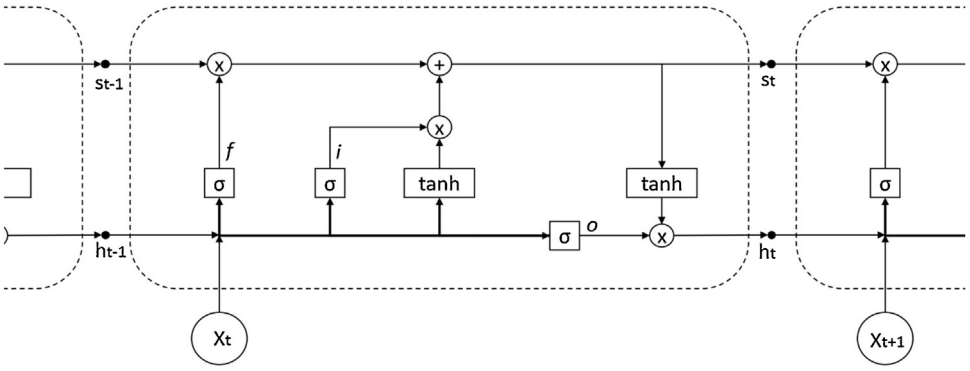


Fig. A3. A Long-Short Term Memory cell (adapted from illustrations based on Olah, 2015).

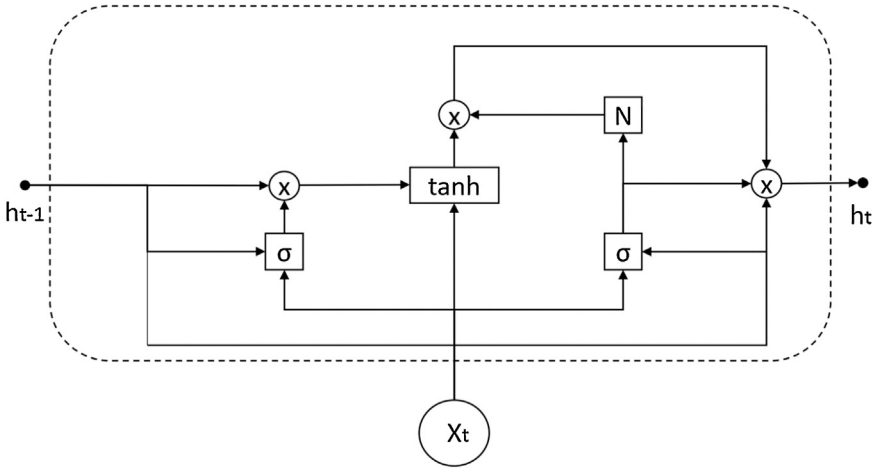
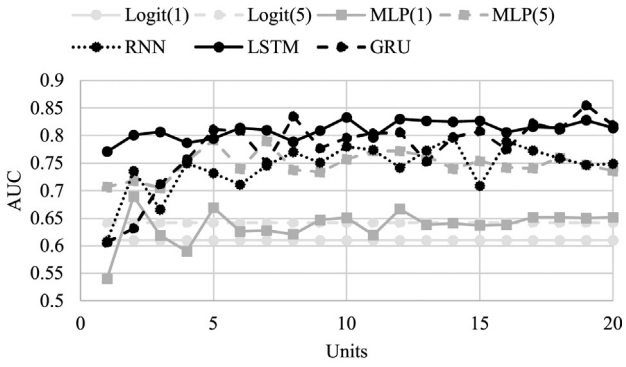


Fig. A4. A Gated Recurrent Unit (adapted from illustrations based on Olah, 2015).

(a) Cross-validation, the size of the ensemble = 17.



(b) Sequential evaluation, the size of the ensemble = 50.

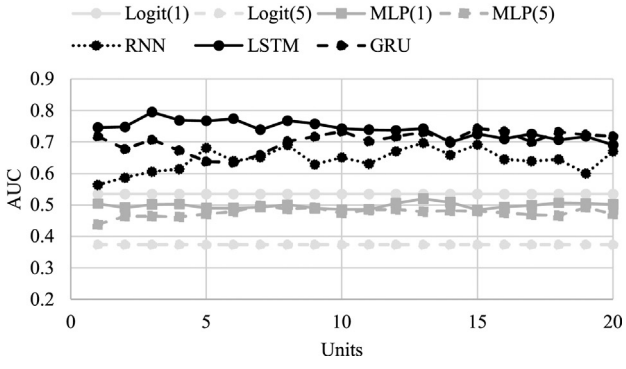
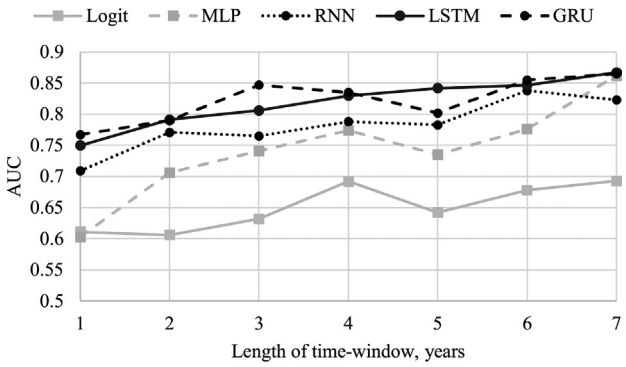


Fig. B1. Sensitivity analysis with respect to the number of units. Panels (a) and (b) show the cross-validated and sequential AUC statistics, respectively. The sample is 1970-2016. Higher AUC is better. The neural nets are estimated as explained in Appendix A.5.

(a) Cross-validation, the size of the ensemble = 85.



(b) Sequential evaluation, the size of the ensemble = 50.

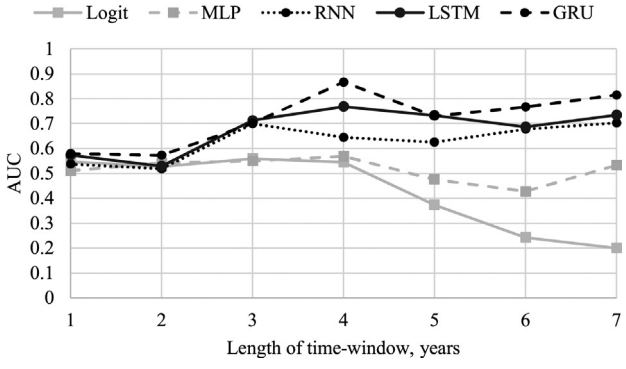


Fig. B2. Sensitivity analysis with respect to the length of the time-window. Panels (a) and (b) show the cross-validated and sequential AUC statistics, respectively. The sample is 1970-2016. Higher AUC is better. The neural nets are estimated as explained in Appendix A.5.

References

Alessi, L., Detken, K., 2011. Quasi real time early warning indicators for costly asset price boom/bust cycles: a role for global liquidity. Eur. J. Polit. Econ. 27 (3), 520–533.

Alessi, L., Detken, K., 2018. Identifying excessive credit growth and leverage. J. Financ. Stab. 35, 215–225.

Alessi, L., Antunes, A., Babecky, J., Baltussen, S., Behn, M., Bonfim, D., et al., 2015. Comparing Different Early Warning Systems: Results From a Horse Race Competition Among Members of the Macro-prudential Research Network, Available at SSRN: https://ssrn.com/abstract=2566165 or https://doi.org/10.2139/ssrn.2566165.

Apel, M., Grimaldi, M.B., Hull, I., 2019. How much information do monetary policy committees disclose? In: Evidence From the FOMC's Minutes and Transcripts, Sveriges Riksbank Working Paper Series 381.

Babecký, J., Havránek, T., Matěju, J., Rusnák, M., Šmídková, K., Vašíček, B., 2014. Banking, debt and currency crises in developed countries: stylized facts and early warning indicators. J. Financ. Stab. 15, 1–17.

Barrell, R., Davis, E.P., Karim, D., Liadze, L., 2011. How idiosyncratic are banking crises in OECD countries? Inst. Econ. Rev. 216, R53–R58.

Barron, A.R., 1993. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans. Inf. Theory 39 (3), 930–945.

Basel Committee on Banking Supervision (BCBS), 2011. Basel III: a Global Regulatory Framework for More Resilient Banks and Banking Systems — Revised Version. Bank for International Settlements.

Behn, M., Detken, C., Peltonen, T., Schuedel, W., ECB Working Paper No. 1603 2013. Setting Countercyclical Capital Buffers Based on Early Warning Models: Would It Work?

Berg, A., Pattillo, C., 1999. Are currency crises predictable? A test. IMF Staff Papers 46 (2), 1.

Beutel, J., List, S., von Schweinitz, G., 2019. Does machine learning help us predict banking crises? J. Financ. Stab. 45, 100693.

Binner, J.M., Elger, T., Nilsson, B., Tepper, J.A., 2004. Tools for non-linear time series forecasting in economics – an empirical comparison of regime switching vector autoregressive models and recurrent neural networks. Adv. Econometrics 19, 71–91.

Binner, J.M., Elger, T., Nilsson, B., Tepper, J.A., 2006. Predictable non-linearities in U.S. Inflation. Econ. Lett. 93 (3), 323–328.

Bluwstein, K., Buckmann, M., Joseph, A., Kang, M., Kapadia, S., Simsek, Ö., 2020. Credit growth, the yield curve and financial crisis prediction: evidence from a machine learning approach. Bank of England Staff Working Paper No. 848.

Bordo, M.D., Meissner, C.M., 2012. Does inequality lead to a financial crisis? J. Int. Money Finance 31 (8), 2147–2161.

Borio, C., 2014. The financial cycle and macroeconomics: What have we learnt? J. Bank. Financ. 45, 182–198.

Borio, C., Drehmann, M., 2009. Assessing the Risk of Banking Crises – Revisited. BIS Quarterly Review, pp. 29–46, March.

Borio, C., Lowe, P., 2002. Assessing the Risk of Banking Crises. BIS Quarterly Review, December 2002.

Borovkova, S., Tsiamas, I., 2019. An ensemble of LSTM neural networks for high-frequency stock market classification. J. Forecast. 38 (6), 600–619.

Bussiere, M., Fratzscher, M., 2006. Towards a new early warning system of financial crises. J. Int. Money Finance 25 (6), 953–973.

Büyükkarabacak, B., Valev, N.T., 2010. The role of household and business credit in banking crises. J. Bank. Financ. 34, 1247–1256.

Caggiano, G., Calice, P., Leonida, L., 2014. Early warning systems and systemic banking crises in low income countries: a multinomial logit approach. J. Bank. Financ. 47, 258–269.

Caggiano, G., Calice, P., Leonida, L., Kapetanios, G., 2016. Comparing logit-based early warning systems: Does the duration of systemic banking crises matter? J. Empir. Finance 37, 104–116.

Caprio, G., Klingebiel, K., 1997. Bank insolvencies: Bad luck, Bad policy, or Bad Banking? Annual World, Bank Conference on Development Economics, pp. 79–04.

Casabianca, E.J., Catalano, M., Forni, L., Giarda, E., Passeri, S., 2019. An early warning system for banking crises: from regression-based analysis to machine learning techniques. Marco Fanno Working Papers 235.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1724–1734.

Cook, T.R., Smalter Hall, A., 2017. Macroeconomic indicator forecasting with deep neural networks, Federal reserve Bank of Kansas City. Research Working Paper 17-11, September, http://dx.doi.org/10.18651/RWP2017-11.

Davis, E.P., Karim, D., 2008. Comparing early warning systems for banking crises. J. Financ. Stab. 4 (2), 89–120.

Davis, E.P., Liadze, I., 2011. Should Multivariate Early Warning Systems for Banking Crises Pool Across Regions? Rev. World Econ. 147, 693–716.

DeGroot, M.H., Fienberg, S.E., 1983. The comparison and evaluation of forecasters. Statistician 32 (1/2), 12–22.

DeLong, E., DeLong, D., Clarke-Pearson, D., 1988. Comparing the areas under two or more correlated receiver operating characteristics curves: a nonparametric approach. Biometrics 44, 837–845.

Demirgüc-Kunt, A., Detragiache, E., 1998. The determinants of banking crises in developed countries. IMF Staff Pap. 45 (1), 81–109.

Demirgüc-Kunt, A., Detragiache, E., 2000. Monitoring banking sector fragility: a multivariate logit approach. World Bank Econ. Rev. 14, 287–307.

Detken, K., Weeken, O., Alessi, L., Bonfim, D., Boucinha, M., Castro, C., et al., 2014. Operationalising the countercyclical capital buffer: indicator selection, threshold identification and calibration options. ERSB Occasional Paper Series No. 5 / June 2014.

Díaz-Martínez, Z., Sánchez-Arellano, A., Segovia-Vargas, M., 2011. Predicción de crisis financieras mediante conjuntos imprecisos (rough sets) y árboles de decisión. Innovar 21 (39), 83–100.

Domaç, I., Martinez Peria, M.S., 2003. Banking crises and exchange rate regimes: is there a link? J. Int. Econ. 61 (1), 41–72.

Drehmann, M., Juselius, M., 2014. Evaluating early warning indicators of banking crises: satisfying policy requirements. Int. J. Forecast. 30, 759–780.

Drehmann, M., Borio, C., Gambacorta, L., Jiménez, G., Trucharte, C., 2010. Countercyclical capital buffers: exploring options. BIS Working Papers 317.

Duttagupta, R., Cashin, P., 2011. Anatomy of banking crises in developing and emerging market countries. J. Int. Money Finance 30 (2), 354–376.

Filardo, A., Lombardi, M., Raczko, M., 2018. Measuring financial cycle time. BIS Working Papers 755.

Fioramanti, M., 2008. Predicting sovereign debt crises using artificial neural networks: a comparative approach. J. Financ. Stab. 4 (2), 149–164.

Fischer, T., Krauss, C., 2018. Deep learning with long-short term memory networks for financial marker predictions. Eur. J. Oper. Res. 270 (2), 654–669.

Fouliard, J., Howell, M., Rey, H., 2019. Answering the queen: online machine learning and financial crises. Presentation at the BIS Annual Conference.

Fricke, D., 2017. Financial Crisis Prediction: A Model Comparison (November 29, 2017). Available at SSRN: https://ssrn.com/abstract=3059052 or https://doi.org/10.2139/ssrn.3059052.

Gers, F.A., Eck, D., Schmidhuber, J., 2001. Applying LSTM to time series predictable through time-window approaches. International Conference on Artificial Neural Networks ICANN 2001, 669–676.

Gers, F.A., Schraudolph, N.N., Schmidhuber, J., 2002. Learning precise timing with LSTM recurrent networks. J. Mach. Learn. Res. 3, 115–143.

Gogas, P., Papadimitriou, T., Matthaiou, 2014. Yield curve and recession forecasting in a machine learning framework. Comput. Econ. 45 (4), 635–645.

Hansen, L.K., Larsen, J., Fog, T., 1997. Early stop criterion from the bootstrap ensemble. 1997 IEEE International Conference on Acoustics, Speech, and Signal Processing 4, 3205–3208.

Hardy, D.C., Pazarbasioglu, C., 1999. Determinants and leading indicators of banking crises: further evidence. IMF Staff Paper 46 (3), 1.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Holopainen, M., Sarlin, P., 2017. Toward robust early-warning models: a horse race, ensembles and model uncertainty. Quant. Finance 17 (12), 1933–1963.

Hornik, K., 1991. Approximation capabilities of multilayer feedforward networks. Neural Netw. 4 (2), 251–257.

Jordà, Ò., Schularick, M., Taylor, A.M., 2015. Leveraged bubbles. J. Monet. Econ. 76, S1–S20.

Jordà, Ò., Schularick, M., Taylor, A.M., 2017. Macrofinancial history and the New business cycle facts. In: Eichenbaum, Martin, Parker, Jonathan A. (Eds.), NBER Macroeconomics Annual 2016, vol. 31. University of Chicago Press, Chicago.

Joy, M., Rusnák, M., Šmídková, K., Vašíček, B., 2017. Banking and currency crises: differential diagnostics for developed countries. Int. J. Financ. Econ. 22 (1), 44–67.

Kaminsky, G., Reinhart, C., 1999. The twin crises: the causes of banking and balance-of-payments problems. Am. Econ. Rev. 89 (3), 473–500.

Kauko, K., 2012. External deficits and non-performing loans in the recent financial crisis. Econ. Lett. 115 (2), 196–199.

Kauko, K., 2014. How to foresee banking crises? A survey of the empirical literature. Econ. Syst. 38, 289–308.

Kauko, K., Tölö, E., 2020a. On the long-run calibration of the credit-to-GDP gap as a banking crisis predictor. Finnish Economic Papers (forthcoming).

Kauko, K., Tölö, E., 2020b. Banking crisis prediction with differenced relative credit. Appl. Econ. Q. (forthcoming).

Knoll, K., Schularick, M., Steger, T., 2016. No price like home: global house prices 1870-2012. Am. Econ. Rev. 107 (2), 331–353.

Kuan, C.-H., White, H., 2007. Artificial neural networks: an econometric perspective. Econom. Rev. 13 (1), 1–91.

Laeven, L., Valencia, F., 2012. Systemic Banking crises database: an update. Working Paper No.12/163.

Lang, J.H., Cosimo, I., Fahr, S., Ruzicka, J., 2019. Anticipating the bust: a new cyclical systemic risk inidicator to assess the likelihood and severity of financial crises. ECB Occasional Paper Series No. 219.

Lo Duca, M., Peltonen, T., 2013. Assessing systemic risks and predicting systemic events. J. Bank. Financ. 37 (7), 2183–2195.

Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. Proceedings of the 31st International Conference on Neural Information Processing Systems, 4768–4777.

Manasse, P., Roubini, N., 2009. "Rules of thumb" for sovereign debt crises. J. Int. Econ. 78 (2), 192–205.

Manasse, P., Savona, R., Vezzoli, M., 2013. Rules of thumb for Banking crises in emerging markets. Universita' di Bologna Working Papers No. 872.

Minami, S., 2018. Predicting equity price with corporate action events using RNN-LSTM. J. Math. Financ. 8 (1), 58–63.

Niculescu-Mizin, A., Caruana, R., 2005. Predicting Good probabilities with supervised learning. ICML 05 Proceedings of the 34th International Conference on Machine Learning, 625–632.
Nik, P., Jusoh, M., Shaari, A.H., Sarmdi, T., 2016. J. Econ. Cooperation Dev. 37 (1), 25–40.
Nyman, R., Ormerod, P., arXiv:1701.01428 2017. Predicting Economic Recessions Using Machine Learning Algorithms.
Olah, C., Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ 2015. Understanding LSTM Networks.
Qi, M., 2001. Predicting US recessions with leading indicators via neural network models. Int. J. Forecast. 17 (3), 383–401.
Reinhart, C.M., Rogoff, K.S., 2009. This Time Is Different: Eight Centuries of Financial Folly. Princeton University Press.
Ristolainen, K., 2018. Predicting banking crises with artificial neural networks: the role of Nonlinearity and heterogeneity. Scand. J. Econ. 120 (1), 31–62.
Roy, S., Kemme, D.M., 2012. Causes of banking crises: deregulation, credit booms and asset bubbles, then and now. Int. Rev. Econ. Financ. 24, 270–294.
Sarlin, P., 2014. Mapping financial stability. Ai Commun. 27 (3), 285–297.
Schularick, M., Taylor, A.M., 2012. Credit booms gone bust: monetary policy, leverage cycles and financial crises, 1870-2008. Am. Econ. Rev. 102 (2), 1029–1061.
Shapley, L.S., 1953. A value for n-person games. Contributions to the Theory of Games 2 (28), 307–317.
Siami-Namini, S., Tavakoli, N., Namin, A.S., 2018. A comparison of ARIMA and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1394–1401.
Suss, J., Treitel, H., 2019. Predicting bank distress in the UK with machine learning. Bank of England Staff Working Paper No. 831.
Tölö, E., Laakkonen, H., Kalatie, S., 2018. Evaluating Indicator for use in setting the countercyclical capital buffer. Int. J. Cent. Bank. 14 (2), 51–111.
von Hagen, J., Ho, T.-K., 2007. Money market pressure and the determinants of banking crises. J. Money Credit Bank. 39 (5), 1037–1066.
Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, Ł., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., Dean, J., arXiv:1609.08144 2016. Google's Neural Machine Translation System: Bridging the Gap Between Human and Machine Translation.