Can Machine Learning Help Forecast the Overall Mortgage Delinquency Rate?

Aaron Kreiner
Oberlin College and Nomura Securities
aaron.kreiner@oberlin.edu

John V. Duca
Oberlin College, Dept. of Economics, 223 Rice Hall, Oberlin, OH 44074, jduca@oberlin.edu
Research Department, Federal Reserve Bank of Dallas, P.O. Box 655906, Dallas, TX 75265
john.v.duca@dal.frb.org

November 1, 2019

Abstract

This paper examines different machine learning models to project the U.S. mortgage delinquency rate one-, two-, and four-quarters ahead. One is a Lasso model that directly scrapes data from FRED and the other two use principal components from these series in a Lasso and artificial neural network (ANN) model. These models can be quickly run using an SQL database to select data from the Federal Reserve Economic Database (FRED) and be fitted ("trained") in-sample from 1970 to 2000 to forecast quarterly mortgage delinquency rates over 2000 to 2018. The training window is updated in each forecast quarter to include new data. A rolling-window and non-rolling window period are tested for the training window. This paper finds that a non-rolling neural network model forecasts better than either Lasso regression model. From experiments dropping broad categories of FRED variables in four-quarter ahead forecasts for the long forecast period, "Money, Banking, and Finance" data were the most important in forecasting this delinquency rate. Over each sub-forecast period, the ANN model outperformed, but to the largest degree over the Great Recession sub-sample 2007-12. This pattern accords with the view that highly nonlinear dynamics can dominate in recessions and the much better ability of ANN models to track nonlinear relationships than the linear, albeit parsimonious Lasso regression approach. The ANN approach outperforms more at longer forecast horizons, which can provide policymakers and regulators more advanced warning to prepare for higher real estate loan losses.

JEL Codes: C45, C10, G51

Key Words: machine learning, forecasting, neural networks, artificial intelligence, mortgage delinquency

* This paper is partly drawn from Kreiner's undergraduate honors thesis, which focused on forecasting the U.S. unemployment rate and which was supervised by Barbara Craig, John Duca, and Ed McKelvey. The views expressed are those of the authors and do not necessarily reflect the views of the Federal Reserve Bank of Dallas or the Federal Reserve System. Any remaining errors are those of the authors.

1. Introduction

Large losses on mortgages played a key role in undermining financial stability and contributing to the Great Recession. To a large extent such losses were not foreseen, prompting economists to revamp their econometric models and policymakers to reform financial regulation to reduce the risk and severity of future credit cycles. This paper assesses whether machine learning techniques could help forecast overall mortgage quality. We do so by assessing two basic machine-learning techniques for deriving information from large datasets to avoid the curse of dimensionality. One technique, the Lasso Regression approach, uses linear methods to drop variables with low information and obtain parsimonious, but yet informative forecasting models. A major disadvantage of linear models in general is that they often do not track the severity of economic downturns or tail risk events, such as mortgage delinquency or foreclosure. Another technique is to build an artificial neural network (ANN) machine learning model, which by mimicking many of the processes of the human brain in a network, can track nonlinear relationships and thus, in principle, better track the severity of recessions (e.g., the unemployment rate) or of loan losses (e.g., the mortgage delinquency rate).[1]

Unlike other published studies that have applied machine learning to mortgage delinquencies we focus on predicting the aggregate rate of mortgage delinquency and not which particular loans in a portfolio are likely to become delinquent by from cross-section characteristics. Doing so keeps the focus on financial stability and detecting major swings in overall loan quality.

This paper implements a four-quarter ahead forecast for the seasonally adjusted, share of mortgages that are 30 to 90 days delinquent. For two major reasons, we pick this rate rather than rate at which mortgages enter the process of foreclosure or the rate at which mortgages are

[1] Indeed, Kreiner and Duca (forthcoming) find that an ANN notably outperforms a Lasso Regression (and also the Survey of Professional Forecasters) in forecasting the unemployment rate.

delinquent for 90 or more days. First, there were a number of legal and regulatory actions taken during the housing crisis that either prevented lenders from foreclosing on late mortgages or affected borrower decisions to prolong delinquency or shorten it with short-sales of homes.[2] As a result, shifts in regulation and legal actions are difficult to track in time series models. Second, mortgages that have been late for a long period of time reflect the impact of past changes more than mortgages that have only recently become delinquent. This makes it more difficult to link the time series behavior of very late mortgages to previous events than for more recent mortgages. Starting from an in-sample training period of 1970-2000, we start forecasting the 30-90 day mortgage delinquency rate four-quarters ahead for 2000:q4 and then progressively add one quarter more data to project the next quarter over 2001:q4-2018:q4. We examine two types of forecasting models—a Lasso regression and a neural network model, each of which can be quickly run using an SQL database.

The latter two models use machine learning techniques to select data from a large number (over 600,000) of macroeconomic variables in the Federal Reserve Economic Database (FRED), ranging from measures of aggregate economic activity such as the money supply, to city-level data. We find that a non-rolling neural network model forecasts bests and outperforms a Lasso regression model across all sample periods. From experiments dropping broad categories of FRED from the ANN model, "Money, Banking, and Finance" data were the most important in forecasting this delinquency rate one-year ahead over the 2000-2018 forecast period. Because the ANN model uses principal components to reduce the number of input variables, it is difficult to pinpoint which individual variables add the most information. One advantage of a Lasso Regression is that we can apply it to either principal components or the underlying variables, and in the latter case, one

[2] Short sales involve homes being sold for less than the mortgage balance outstanding and lenders being willing to receive less than full repayment of principal from such sales.

can determine which particular variables add the most information. The top five variables were credit, financial or labor market indicators.

To establish these findings, the paper is organized as follows. Section 2 reviews relevant literature on machine learning for forecasting mortgage delinquency rates and using marginal information that forecasters may ignore. Section 3 details the data selection methodology that is applied to the FRED Database. Section 4 begins by providing a theoretical overview of the machine learning techniques used—including principal components analysis, Lasso regression, and neural networks—before outlining the forecasting algorithms for each statistical model. The results are presented in Section 5, with perspective and broader lessons discussed in the conclusion.

2. Literature Review

New machine learning models could have widespread use and have a large impact on economics. According to Mullainathan and Spiess (2017), most econometric models focus on the best estimates for coefficients on regressors (the β's in the in-sample data, whereas machine learning focuses on explaining the independent variable $\hat{y}$ on out-of-sample data). In other words, econometricians focus on the weight of factors in-sample, whereas machine learning focuses on the out-of- sample performance in predicting the dependent variable. For these reasons, machine learning is best suited for economic analysis that requires high out of sample performance, such as forecasting. Some examples of machine learning in economics include measuring economic activity using satellite images, classifying industries based on their 10-K filings, and building forecasts for core economic indicators (Mullainathan and Spiess, 2017).

With respect to macro-time series applications, there are several models that apply machine-learning techniques to forecast real GDP growth, and among the published ones are Prasad and Sinha (2018) and Jung et. al. (2018). Many of these were inspired by Giannone,

Reichlin, and Small's (2008), whose pioneering work on now-casting emphasized the advantages posed by deriving information from a myriad of variables of mixed frequency and release dates. A handful of published studies have also applied machine-learning to forecast the civilian unemployment rate, including is Cook and Hall (2017), who use neural networks based on a single macroeconomic indicator (monthly lags of the civilian unemployment rate), and Kreiner and Duca (forthcoming), who use machine-learning techniques to pull data from the large FRED database. Both models produce better forecasts of near-term unemployment than the Survey of Professional Forecasters (SPF), with the latter able to outperform the SPF at somewhat longer (two- and four-quarter ahead) horizons, likely reflecting the advantage of drawing upon a large set of publically available data.

A few studies apply machine-learning techniques to predict the loan quality of individual mortgages in particular portfolios. For example, Kvamme, et al. (2018) are able to forecast Norwegian mortgage defaults on individual loans using machine-learning techniques on household data on checking, savings, and credit account history. And in an unpublished, but posted paper, Ponomareva, Epstein, and Knight (2019) use past mortgage payment behavior plus information on eight characteristics of loans at origination to predict whether future loan payments will be on time, or late according to three categories. Both of these studies apply machine learning to a limited set of data to forecast future loan status.

Perhaps the most impressive of posted or published machine learning models of mortgage quality is a study by Sirignano, Sahwani, and Giesecke (2016). This paper estimates and forecasts the payment behavior of over 120 million U.S. mortgages originated between 1995 and 2014 on detailed loan specific and local area conditions. The authors find that the ability of their ANN enables them to better track the nonlinear behavior of mortgage loan quality than prior models of

loan-level delinquency, and their results accord with the double-trigger approach to mortgage defaults that attributes defaults to shocks to borrower income and to instances of low or negative net equity positions of homeowners.

While these papers make important contributions regarding the prediction of individual loan quality—a very worthwhile topic in and of itself—there remains the question of whether machine learning can help analysts gauge overall trends in mortgage quality. This issue is very relevant for tracking macroeconomic cycles and financial stability. Indeed, several central banking and financial supervisory institutions use heat maps and network analysis to track the myriad of risks to financial stability (*inter alia*, see, Aikman et al., 2017, and Gai, Haldane, and Kapadia, 2011). Our study tries to supplement these efforts with using machine learning to provide an overall forecast of the early-stage overall mortgage delinquency rate by drawing upon a broad set of publicly available information and using relatively long in-sample and out-of-sample periods. Long sample periods are especially desirable because there are only a few major delinquency rate cycles given the long cycles in housing and mortgage behavior. Partly because delinquency rates were not available for sub-classes of mortgages over long samples and partly to focus on the overall tone of loan quality, we model the delinquency rate on all types of mortgages.

Our paper shows that machine learning provides superior out-of-sample forecasts of the mortgage delinquency rate relative to forecasts from linear Lasso regression models. While this method cannot produce a set of β's, this is compensated by adding and dropping key data categories and testing out of sample performance before and after omitting sets of data. The use of this methodology indicates which types of variables have the largest benefit to improving forecasts. If omitting a set of variables causes the out of sample error to increase the most, then the machine learning algorithm puts the highest weight on these variables.

3. Data

We draw data from the Federal Reserve Economic Data (FRED) database, which contains about 600,000 variables that are binned into categories labelled with a category number. Categories and data can lie within other categories. In this way, FRED is structured as a recursive tree database with categories and data stored in trees. For example, category 0 contains 8 of the prime categories in FRED: Money, Banking, and Finance; Population, Employment, and Labor Markets; National Accounts; Production and Business Activity; Prices; International Data; US Regional Data; and Academic Data. Table 1 summarizes which types of data are in each of these prime categories. These categories will be used to assess which types of variables significantly affect the overall mortgage delinquency rate. The algorithm for scraping the data identifies all the categories by making recursive calls to the FRED tree structure. Next, all of the variables are collected from each category. Data are the actual time series for a particular variable. Only data series that meet the following criteria are retained in the final model:

1. *Frequency:* The variable must be monthly or quarterly. If a variable has a frequency higher than monthly, the series is converted to a monthly series using FRED's conversion algorithm. This criterion accommodates the vector construction of inputs and outputs. The forecasting algorithm section will discuss this further.
2. *Consistency:* The variable must have consistent data from January 1970 to September 2018. The data series cannot have any missing values. This criterion ensures the forecast uses variables that are continuously available.
3. *Revision History*: The variable's data are from the first revision if available. Otherwise, use the revision that was first released. This minimizes the use of information available in the future. This represents the first figure published for the first revision available.

Category	Types of Variables Included
Money, Banking, and Finance	Interest Rates, Exchange Rates, Monetary Data, US Financial Indicators, Banking, Business Lending, Foreign Exchange Intervention
Population, Employment, and Labor Markets	Current Population Survey (Household Survey) , Current Employment Statistics (Establishment Survey) , ADP Employment, Education, Income Distribution, Job Openings and Labor Turnover (JOLTS), Labor Market Conditions, Population, Productivity and Costs, Minimum Wage, Weekly Initial Claims, Tax Data
National Accounts	National Income and Product Accounts, Federal Government Debt, Flow of Funds, US Trade and International Transactions
Production and Business Activity	Business Cycle Expansions & Contractions, Construction, Finance Companies, Health Insurance, Housing, Industrial Production and Capacity Utilization, Manufacturing, Retail Trade, Services, Technology, Transportation, Wholesale Trade
Prices	Commodities, Consumer Prices Indexes (CPI and PCE), Cryptocurrencies, Employment Cost Index, Health Care Indexes, House Price Indexes, Producer Price Indexes, Trade Indexes
International Data	Countries, Geography, Indicators, Institutions
US Regional Data	States, Census Regions, BEA Regions, BLS Regions, Federal Reserve Districts, Freddie Mac Regions
Academic Data	Banking and Monetary Statistics 1914-1941, Data on the nominal term structure model from Kim and Wright, Historic Federal Reserve Data, NBER Macrohistory Database, Penn World Table 7.1, Penn World Table 9.0, Recession Probabilities, Weekly US and State Bond Prices, 1855-1865, Economic Policy Uncertainty, Sticky Wages and Comovement, A Millennium of Macroeconomic Data for the UK.

Table 1: Summary of the Types of Variables in FRED.

4. *Forecasts*: All variables that are from a forecast are omitted to preclude the use of future information and ensures that the model does not gain performance from an outside forecast.
5. *Discontinued Series*: If a series is discontinued, then it is omitted. This criterion ensures the forecast uses variables that are currently available.
6. *Lagged*/*Leading Variables:* If a variable is lagged or from the future, it is omitted to avoid using future information and adding noise from time series reported at the wrong date.

The dependent variable is the seasonally adjusted total mortgage delinquency rate for loans 30 to 90 days delinquent. Other filtered variables are the independent variables, including lags of the delinquency rate. The current and four lags of delinquency are used to forecast delinquency four quarters later. After applying the above criteria, most variables are dropped. Table 2 summarizes how many variables in each prime category were considered and selected. On average, fewer than 2% of the variables in each category met the filtering criteria. In particular, none of the

Category	**Number of Variables Before Filter**	**Number of Variables After Filter**	**Survival Rate**
Money, Banking, & Finance	9,424	235	2.49%
Population, Employment, & Labor Markets	25,313	906	3.58%
National Accounts	17,435	2,919	16.74%
Production & Business Activity	11,119	844	7.59%
Prices	14,158	769	5.43%
International Data	158,700	4,356	2.74%
US Regional Data	337,176	617	0.18%
Academic Data	15,642	0	0.00%
Total	588,967	10,646	1.81%

Table 2: Number of Variables in Each Category and Selected for the final model.
(Sources: FRED and authors' calculations. "Survival Rate" measures the percent of variables per category that were selected for the final model.

academic variables were selected while 17% of the national accounts variables were selected (the highest in percentage terms). In absolute terms, international variables had the highest number of variables in the final model.

4. Theoretical Models

This section summarizes and details the mathematics and algorithms behind the statistical models used in the forecasting algorithm section. In particular, principal components analysis is discussed first as a black box that reduces the dimensionality of the variables used for the forecast. Next, neural networks are presented as a highly non-linear black box for forecasting, which relies on computational techniques that mimic those used in the human brain for processing information. Finally, a Lasso regression is presented for forecasting that relies on variable selection (or omitting irrelevant variables) and a dynamically sized error term.

4.1 Principal Components Analysis

Principal components analysis (PCA) is a dimensionality reduction algorithm that inputs a dataset and outputs another with fewer variables and linearly independent data columns. Principal components analysis is used to reduce multicollinearity and redundancy in the variables of the forecasting model. Failure to do so causes overfitting and contributes noise to forecasts from the final model. Subsections 4.2 and 4.3 provide more details on the black boxes used in forecasting. In implementing PCA, the convexity of the problem solved by the PCA algorithm ensures that the computer can solve the problem efficiently (see Crawford, 2015).

4.2 Lasso Regression

A Lasso regression can use the data to create unemployment forecasts. A Lasso or “least absolute shrinkage and selection operator” is a type of regression that uses both variable selection and regularization to maximize prediction accuracy and coefficient interpretation. Regularization

refers to techniques that reduce noise in the training data. A Lasso regression can assign variables a weight of 0 (even if they carry variance) and thus has a dimensionality reduction element embedded within it. This aids in the interpretation of estimated coefficients by reducing the number that need to be interpreted. A Lasso regression also has regularization as a feature. This allows the weights of the regression to not be too high or too low arising from noise or over-fitting.

The mathematics behind the Lasso Regression are similar to those of a linear regression. Suppose we have independent variable data X with k factors and dependent variable Y. X and Y are expressed in matrix notation. The Lasso minimizes the sum of squared errors with an upper bound on the sum of the absolute values of the model parameters (or regularization of the parameters). Let β be the matrix of coefficients for this regression. The Lasso, therefore, solves the following optimization problem:

$$min\ \frac{||Y - X\beta||_2^2}{n} \tag{1}$$

$$\text{subject to: } \sum_{j=1}^{k} ||\beta||_1 < \text{t} \tag{2}$$

We note eq. (1) is equivalent to ordinary least squares. $||Y - X\beta||_2^2 = \sum_{i=0}^{n}(Y_i - \beta X_i)^2$ or the sum of squared residuals and $||\beta||_1 = \sum_{j=1}^{k}|\beta_j|$. t is the upper bound for the absolute sum of the coefficients. The smaller the value of t, the smaller the absolute size of the coefficients will be in the final regression. t therefore controls for overfitting and noise by ensuring that any one coefficient is not too large. Equations (1) and (2) can be rewritten in terms of a new parameter: λ. Shrinkage refers to the degree the magnitude of the coefficients can be reduced as compared to OLS. When λ=0 it corresponds to no shrinkage and t equals infinity; this is OLS. λ must be non-negative because shrinkage refers to the absolute magnitude of the coefficients, which must be positive. Using this theory, the optimization problem can be re-written as follows:

$$\hat{\beta}(\lambda) = \underset{\beta}{\operatorname{argmin}} \left(\frac{\left|\left|Y - X\beta\right|\right|_2^2}{n} + \lambda||\beta||_1 \right) \quad (3)$$

Equation (3) represents the *optimization* problem for $\hat{\beta}$ as a function of the shrinkage parameter λ and the matrix of unknown β's. Equations (1) and (2) describe the same optimization problem as equation (3), but λ is more widely used in the literature and econometric packages as compared to t. As is the case with principal components and OLS, the Lasso regression has a convex optimization function and thus estimates for $\hat{\beta}$ are relatively easy to find. In the optimal solution, a $\hat{\beta}_j$ can be zero meaning the weight is also zero. The shrinkage parameter allows series with non-zero variance to have the potential to have the weight be zero.

Even though the Lasso regression has some embedded functionality to deal with over-fitting and noise, reducing these two characteristics is ideal before the Lasso regression is run. In the context of the Lasso regression, the idea of zero weight on some factors helps reduce overfitting. However, to guarantee overfitting does not occur, it is best to have more rows of data than factors. For this reason, it is recommended that a dimensionality reduction algorithm like PCA be used before using a Lasso regression (see Fonti 2017). This is discussed further in Section 5. A Lasso regression's objective function is convex and runs quickly, making it appealing to analyze big data.

4.3 Artificial Neural Networks (ANN's)

An artificial neural network (or ANN) is a statistical computational system modeled after biological neural networks in the brain. The ANN "learns" by being given examples that pair inputs to outputs. These pairs are called the "training data" or the "fit data". In contrast to OLS or a Lasso regression, the form of the model is not specified beforehand. While an optimization procedure is used, the actual model is dynamic and changes depending on the input. In this way, ANN's are highly nonlinear and can differ drastically depending on the nature of the data. While it is more

difficult to interpret the weights in these models, they potentially offer a higher degree of accuracy in forecasting, which is ideal since the goal is to find the model with the lowest root mean squared error for forecasting out-of-sample or "testing" data. In-sample fit and the interpretation of weights on the fit data are secondary to the model's ability to forecast the test data.

Because an ANN is constructed based on brain neural activity, it is helpful to start by considering a rudimentary model of the brain. For a simple example and applying some key terms associated with ANN's, consider the ubiquitous depiction of thinking in Figure 1. Suppose Figure 1 models the brain's decision of whether to walk left or right. The brain has five inputs, denote them a, b, c, d, and e. Each could represent a binary variable, such as whether a wall is immediately to our left or right. There is only one output, which is the trinary variable: 0 if going right, 1 if going left, and 2 if doing nothing. Each circle depicts a neuron, a computational unit that takes in a series of inputs and produces an output. Consider the bottom-most circle closest to "inputs." This neuron has five inputs and it outputs its computed value to four neurons in the middle column. The neurons in the middle column takes those outputs as inputs, and then computes their output, which

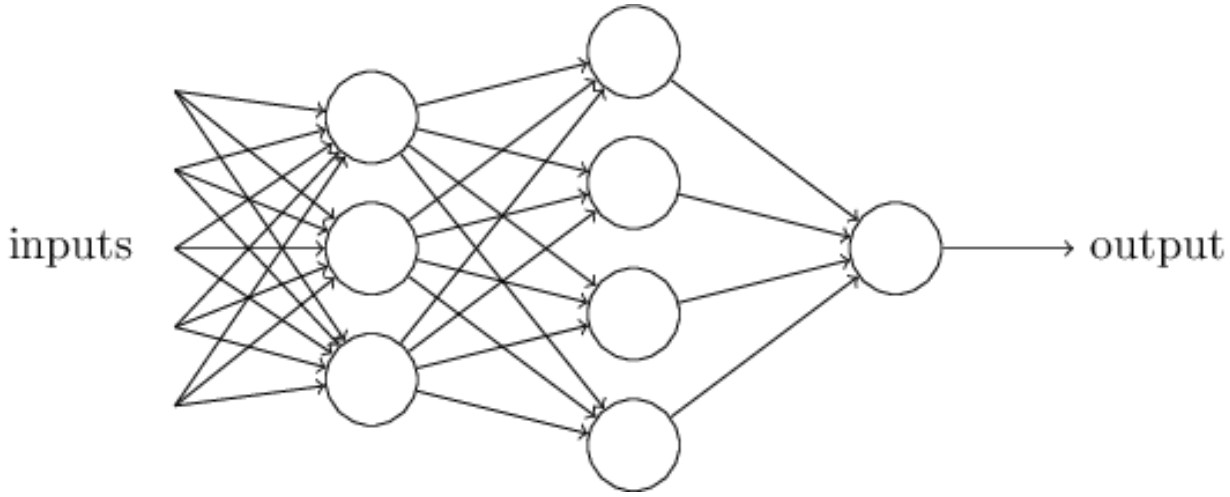

Figure 1: A Schematic of Human Brain Analytical Processes

it provides to the final neuron. This final neuron then makes the final computation and produces the final output.

There are several key terms from this example. The three circles closest to "inputs", the next four circles, and then the circle closest to output form "hidden layers." These are the computational units that transform inputs to outputs. The rule that converts a neuron's input to an output is considered to be an activation function. Because a neuron is a simple processing unit, the activation function cannot be overly complex. The non-linearity of the network derives from the repeated iteration, clustering, and connection of neurons. Each ray in the diagram connects a neuron to another neuron. Consider an arbitrary connection of neuron *a* to neuron *b*, where neuron *a* is an input to neuron *b*. The output of neuron *a* has an attached weight in the propagation function of neuron *b*. This is true for every connection of neurons in the diagram. One can therefore associate a set of weights with the above network.

Each neuron also has an associated bias that measures the intercept of the propagation function. The weights determine the slope of the parameters of the activation function while the bias determines the intercept. This is needed to provide optimal output at each layer of the network.

To provide more details on the calculation embedded in an arbitrary neuron in a neural network, consider an arbitrary neuron: j. Neuron j receives the outputs $o_{i_1}, o_{i_2} \dots, o_{i_n}$ from neurons $i_1,\ i_2, \dots, i_n$ (which are connected to neuron j). The propagation function of neuron j is defined as the transformation of each of the outputs with respect to the weights into one network input. This is the weighted sum of the outputs from neurons $i_1,\ i_2, \dots, i_n$ and the weights $w_{i_1,j}, w_{i_2,j}, \dots, w_{i_n,j}$. $w_{i_1,j}$ represents the weight from neuron $i_1$to neuron j. Let $I = \{i_1,\ i_2, \dots, i_n\}$. Let the output of the propagation function be net_j. net_j is therefore:

$$net_j = bias_j + \sum_{i \in I} o_i * w_{i,j} \tag{4}$$

In equation (4), net_j is a number with singular dimension and $bias_j$ is the calculated intercept. The propagation function thus reduces the dimensionality of the inputs of neuron j from n to 1. Next, the activation function transforms net_j into a probability between 0 and 1. This paper uses Python for the implementation of ANN's.

There are four types of activation functions used in Python's artificial neural networks library: *Identity*, *Logistic*, *Tanh*, and *Relu*. These functions take the output from the propagation function and transform it into the output of the overall neuron. These functions determine if a neuron fires. In other words, a neuron firing determines if the neuron is important in fitting the in-sample data. The functions are described below:

1. *Identity*: f(net_j)= net_j. net_j is rounded to 1 if $net_j > 1$ or rounded to 0 if $net_j < 0$. This is the simplest of activations.
2. *Logistic:* f(net_j)= $\frac{1}{1+e^{-net_j}}$. This is the most commonly used activation as it has the best success on most datasets as compared to the other three activation functions. The function is a sigmoid with asymptotes at 0 and 1.
3. *Tanh*: f(net_j)=tanh(net_j). Once again, this activation has asymptotes at 0 and 1. The curvature of tanh is less steep than the sigmoid, meaning each activation probability is lower than the sigmoid.
4. *Relu*: A rectified unit linear function is f(net_j)=max(0, net_j). If this quantity exceeds 1, then the output is rounded down to 1.

From these, tests select which activation function provides the greatest performance on the out of sample data as gauged by the RMSE. Each of these activations returns a probabilistic value from 0 to 1. We say this neuron *fires* if the probability is greater than the predetermined threshold value. It can be shown that the threshold value in a given neuron is the maximum gradient of the

activation function. A firing neuron represents that the inputs to neuron j have weight on the inputs of the neurons in the next hidden layer. The output function passes the neuron's firing state to the next neuron as an input. It is important to note that the neuron outputs a singular firing state, but sends this output to multiple neurons in the next hidden layer. This process is depicted in Figure 2, which modifies a chart from Kriesel's (2009).

However, it is important to note the final output (which is the unemployment rate in this study) is a continuous positive number. The neurons before the output have activation functions that output real numbers (such as *ReLu*), instead of being restricted between 0 and 1. This is known as MLP Regression, or multilayer perceptron regression.

The above paragraphs describe the calculation behind neural networks. We now describe the process of obtaining weights and biases within the network by explaining the algorithm behind the MLP regression and the calculation of weights and biases within a neural network. An optimization procedure is used that minimizes the error between the predicted output values of the

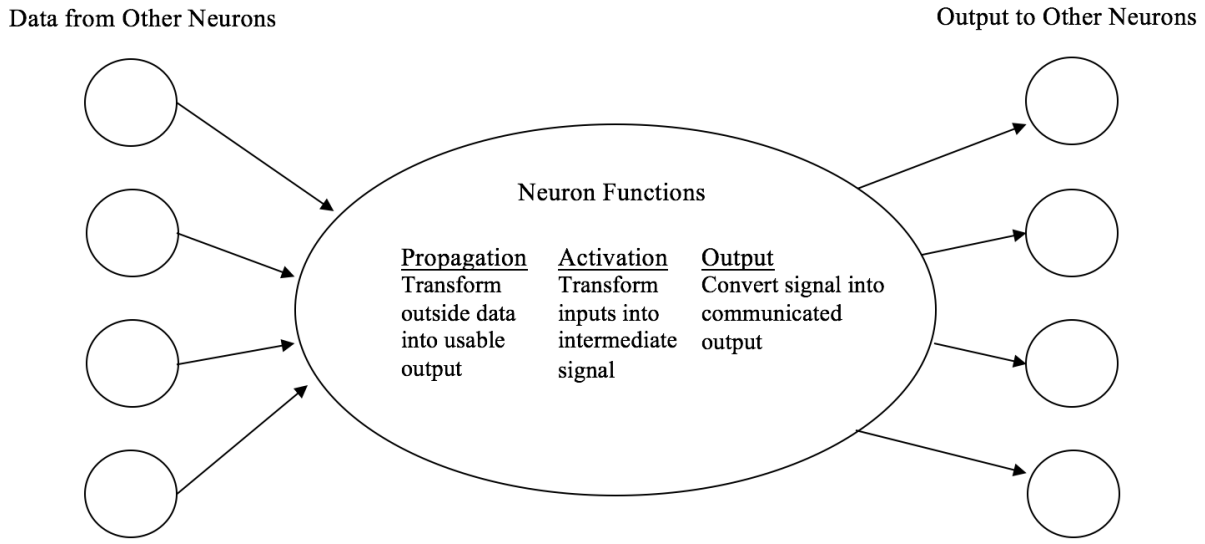

Figure 2: Numerical Processing in a Given Neuron
(Note: each of the data outputs are the same, and then enter as inputs to the neurons in the next hidden layer.)

training data and the actual output values. This can be expressed mathematically as follows. Let there be p pairings of inputs and outputs. Let x(i) be a n-dimensional vector for the i-th input and d(i) be a singleton for the actual output of x(i). In addition, let w be the unknown weight matrix of the ANN and let b be the unknown bias network of the ANN. Thus, we would like to minimize the squared error of the predicted output as per the ANN and the actual output. The predicted output can be calculated per the methodology in the above paragraph. If $y(x: w; b)$ denotes the predicted output of x conditional on w and b, the objective function is:

$$min \sum_{i=1}^{p} \|y(x(i): w; b) - d(i)\|^2 \qquad (5)$$

The y function is dependent on the activation of the neurons in the ANN, the number of hidden layers, and the number of inputs. These determine how large the optimization problem is. However, the optimization function is convex, making it easily solvable on a computer. This process generates the weights and biases for the ANN based on the training data. We can then take the fit data, using the methodology described above, to calculate the predicted outputs. In this way, neural networks learn based on the training data and forecast based on the fit data (see Kriesel, 2009).

As a collective, neural networks are a highly non-linear black box that operates in a similar fashion to neural networks in the brain. Networks are comprised of neurons, which contain propagation, activation, and output functions for computation. The choice of activation function depends on the data and maximizes the performance on the out of sample data. Neurons are connected through a matrix of weights and biases, which weight neurons' outputs into inputs into neurons in the next hidden layer. To obtain the weights and biases, an optimization procedure minimizes the squared error of the predicted output versus the actual output on each of the vectors in the training data.

5. An Overview of Forecasting Algorithms

This section summarizes how forecasts are calculated for each model type. The models include an artificial neural network (ANN) and Lasso Regression using principal components, and a Lasso Regression on all the continuously available (cleaned) FRED set of variables. Each model's forecast performance is assessed relative to each other because a long time series on survey forecasts of the mortgage delinquency rate are unavailable.[3]

5.1 Neural Networks

The forecasting methodology for both machine learning models are depicted in Figure 3. The table at the top of the figure provides the data for the forecast. Each row represents a quarter of data, where the independent variables span from 1970:q1 to 2017:q2. In this figure, there are two types of variables: quarterly (Y) and monthly (X). While the table has only two variables, there are thousands more independent variables in the actual model. Only two are presented for ease of reading the figure. Quarterly data are placed in a row where the "Data Date" matches the reported date of the variable. For example, if Y=1 in 1970:q1, it will be placed in the first empty square. Monthly variables are slightly more complicated. Within a quarter there are three months. Therefore, each monthly independent variable will occupy 3 columns, each of which can be considered a separate variable. The first column represents the first month of the quarter, the second column the second month of the quarter, and so on. For instance, suppose the monthly variable X has 1970:q1 data 5,6,7. Each of these numbers is placed consecutively in the appropriate month. This then produces Table 3 for the independent variables for 1970:q1. Using this methodology, the entire table can be filled for all the quarters and independent variables. The

[3] In contrast, Kreiner and Duca's (forthcoming) study is able to compare ANN and Lasso forecasts of the unemployment rate with the mean forecast from the Survey of Professional Forecasters (SPF). Their ANN model outperformed the Lasso models and also the SPF forecasts—but by a larger degree than the Lasso models.

final column is the dependent variable. This column is a one-year ahead forecast of the independent variables because this paper forecasts the unemployment rate one year ahead.

The first forecast uses data from 1970:q1 to 1999:q4 to forecast 2000:q4. First, the independent variable data from this time period are put into a PCA algorithm with a variance threshold of 0.99. The algorithm is a preprocessing step for the actual forecasts. Then, one year's worth of lags is added for each column of independent variable data. These are appended to the dataset as separate columns for each variable. The forecast takes the training data from 1970:q1 to 2000:q3 and trains the model. The weights, biases, and other metrics (depending on the type of model) from the 1970:q1-1999:q4 data are used to forecast the unemployment rate. The above process—illustrated in Figure 3 and including the PCA algorithm—is repeated for each forecast. For the non-rolling window variant, the beginning of training is fixed. In other words, the first pair of inputs and output will always be 1970:q1. In the rolling window variant, the beginning of training increases by one quarter for each forecast. This means that the second forecast has training start in 1970:q2 instead of 1970:q1. The rolling window, therefore, has a fixed number of rows of data for training. Both types of forecasts will roll in one quarter of new data as available. In particular, for the second forecast, 2000:q1 can now be added to the independent variables of the model.

For the neural network variant, we note that the weights, biases, thresholds, etc. change from forecast to forecast during the 2001-2018 forecasting period. However, there are parameters that are fixed. The number of hidden layers, number of neurons per hidden layer, activation function, and rolling versus non-rolling window remain constant across the forecasting period.

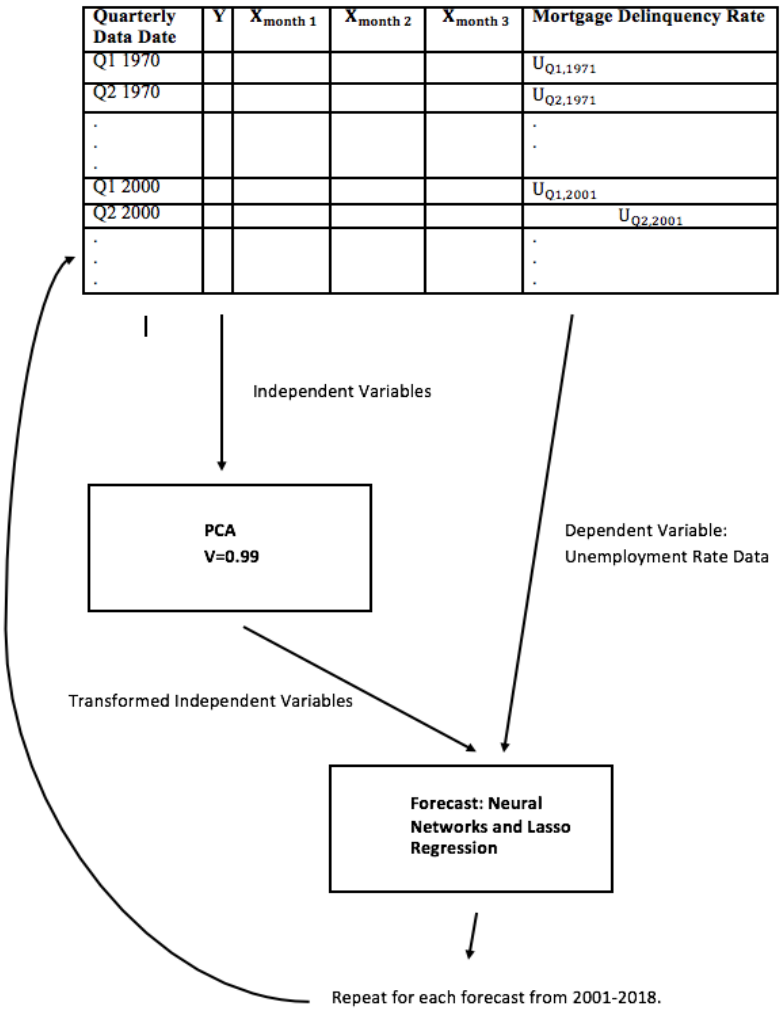

Figure 3: The Forecasting Algorithms for the Lasso and Neural Network Models

Data Date	Y	$X_{month\ 1}$	$X_{month\ 2}$	$X_{month\ 3}$
Q1 1970	1	5	6	7

Table 3: An Example of the Independent Variable Reporting for 1970:q1

However, each of these fixed hyper-parameters is varied to determine the optimal forecast configuration. In other words, the entire forecast is re-run making small changes in each of these parameters. This paper tests 1-150 neuron hidden layer size; one, two, and three hidden layers; four different activation functions; and rolling and non-rolling windows.

5.2 Lasso Regression

The Lasso Regression uses the same overall structure as the neural network model. However, the only hyper-parameter tested is the choice of the shrinkage parameter: λ. λ is varied to generate a series of forecasts from 2001-2018 to determine the optimal forecast configuration.

6. Results

The root mean squared error is used to assess forecast accuracy of the three models. The error is rounded to the nearest 0.01 as forecasts are often similarly rounded.

6.1 Finding the Best Neural Network Model

This subsection details finding the best model for the entire dataset using neural networks. For the neural network model, the optimal neural network configuration is first found. One, two, and three hidden layers were tested—layers greater than three were not tested because they were too computationally excessive for the computer used. Within each hidden layer, the hidden layer size was varied between 1 and 150. Hidden layer sizes over 150 were not tested because the RMSE was at a minimum at about 120 neurons for each number of these hidden layer sizes. Figure 4 plots the RMSE for each off these network configurations. There are three series, one for each number of hidden layers. Within each series, the number of neurons per hidden layer is varied. Based on the chart, two and three hidden layers on average outperform one hidden layer. For the four-quarter ahead horizon, the best configuration, based on RMSE, occurs at a configuration with three hidden layers and 58 neurons per layer. The RMSE for this configuration is 0.24. For the two-quarter

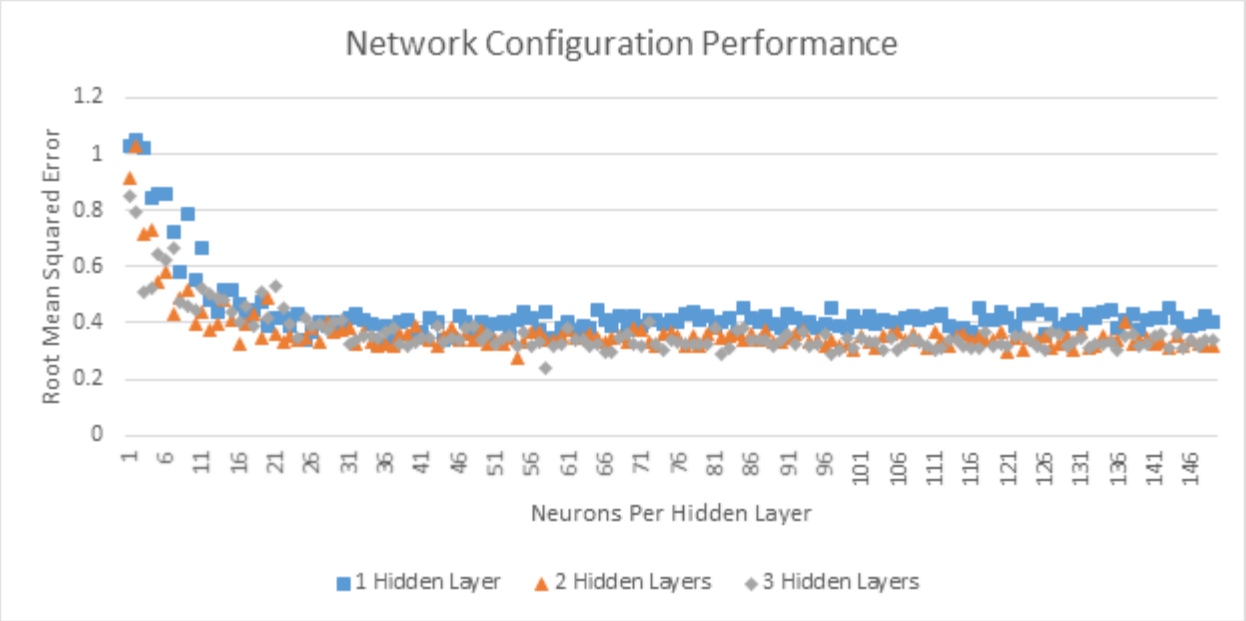

Figure 4: Summary of Hidden Layer Performance.
(Sources: FRED and the authors' calculations.)

horizon, the best fitting configuration had three hidden layers and 53 neurons per layer, and for the one-quarter ahead horizon, the best fitting configuration had three hidden layers and 51 neurons per layer,

We next identify the best activation function for the neural network. We use the optimal configuration found above as a given for this test. While in theory it is best to test all neural network configurations and activations, it would be too time consuming to test all possibilities. Instead, the best choice for each test is passed onto the next test. The *activations logistic, identity, logistic, tanh,* and *relu* are tested. The results of this test are shown in Figure 5. Logistic is the best activation function with respect to RMSE by a landslide.

This finding is also consistent with the neural networks papers discussed earlier in the literature review. Given the above two characteristics, the final attribute tested is the rolling versus non-rolling window of the training or fit period. For the latter, the training starts in 1970:q1. In the

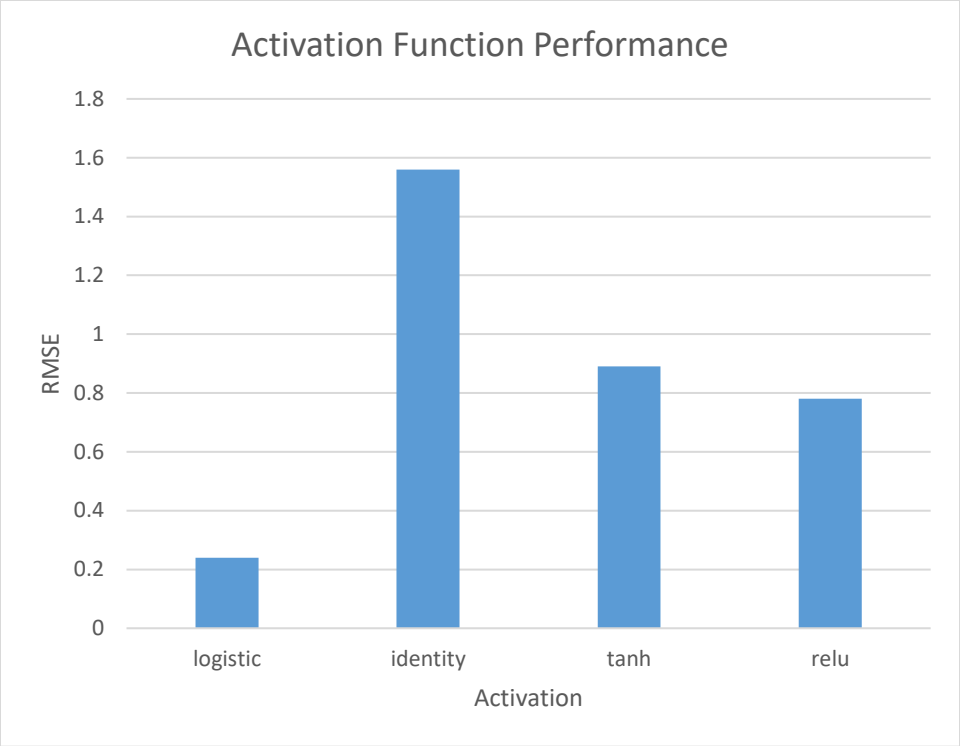

Figure 5: Summary of activation function performance.
(Sources: FRED and authors' calculations.)

rolling variant, the beginning of training rolls ahead by one quarter for each projection. The rolling and non-rolling results are reported in Figure 6.

The fixed time window (or non-rolling) greatly outperforms the rolling time window variant. This result has economic significance. When predicting the future delinquency rate, omitting macroeconomic data from relatively old recessions and expansions has a large positive impact on the error. Rolling out the old data increases error because all past cycles have some effect on new cycles. In other words, including old mortgage and other data patterns helps capture the nature of credit quality cycles. The more cycles embedded within the model, the higher the probability the model can predict the mortgage delinquency rate in new cycles. This finding is consistent with machine learning models of the unemployment rate, for which Montgomery et. al.

(1998) and Kreiner and Duca (forthcoming) find that including lags increases their model's forecast accuracy.

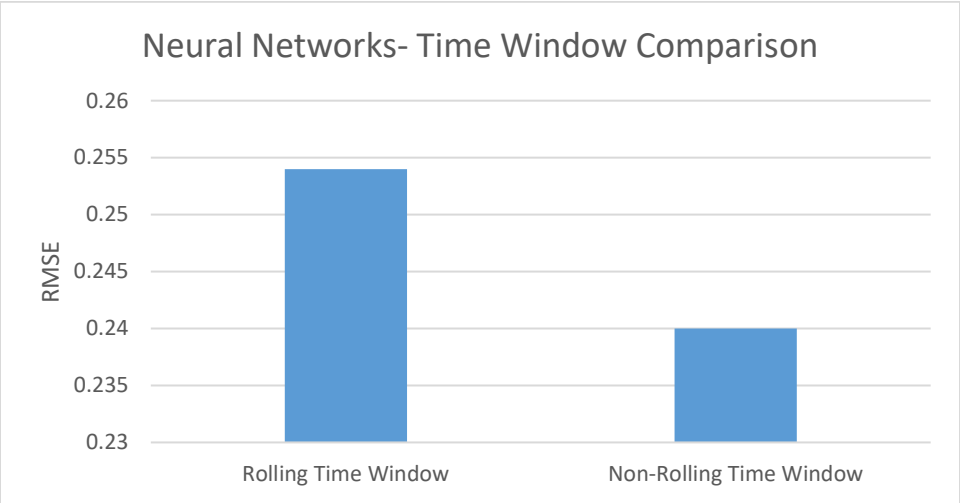

Figure 6: Rolling vs. Non-Rolling model performance
(Sources: FRED and authors' calculations.)

6.2 Finding the Best Lasso Regression Model

Using the findings of Subsection 6.1, we use the non-rolling training window as it had the best RMSE. The only test in this subsection is varying λ to find the appropriate shrinkage parameter that minimizes error. Results are shown in Figure 7. λ is varied from 0 until λ=1. λ is not varied outside this range as λ is shown to monotonically increase for the Lasso model using PCA inputs, after λ equaled 0.38, 0.35, and 0.28 for the one-, two-, and four-quarter ahead horizons, respectively. For the Lasso model drawing directly on data from FRED, after λ equaled 0.80, 0.90, and 1.05 for the one-, two-, and four-quarter ahead horizons, respectively. These λ yielded the lowest RMSE for their horizons and input datasets, which are reported later in Table 4. These will represent the best of the Lasso Regression type models.

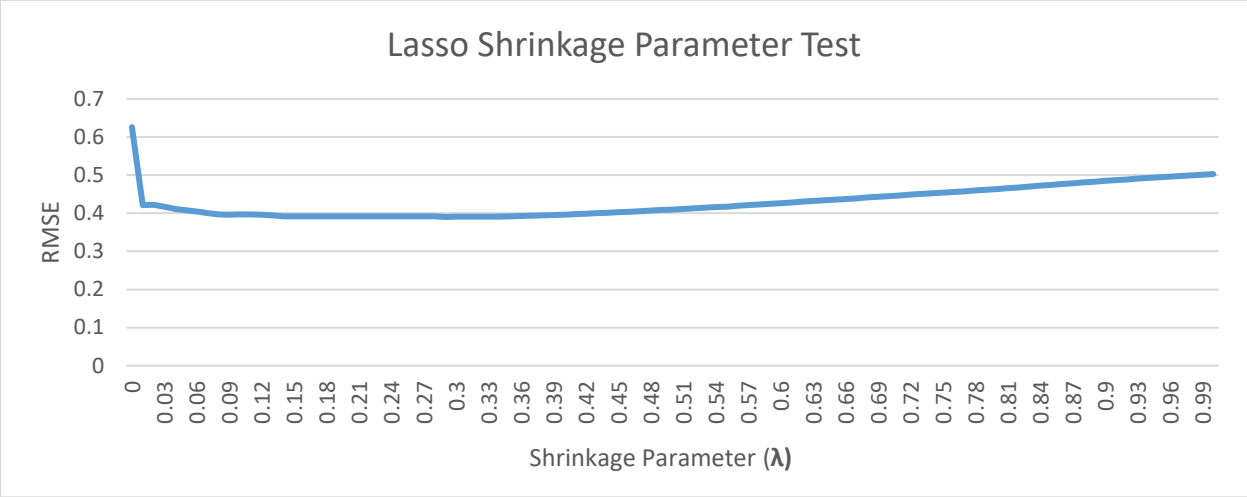

Figure 7: Lasso Shrinkage Parameter Test Results
(Sources: FRED and authors' calculations.)

6.3 Full (2001-2018) Model Results

We compare the RMSE's from four-quarter, two-quarter, and one-quarter ahead forecasts of the mortgage delinquency rate from the ANN and two Lasso models that had the best configurations in sections 6.1 and 6.2. Data through 2018:12 were retrieved on January 7, 2019, and the ANN and one of the Lasso use 58 principal components derived from those data. Figure 8 plots each of the models' delinquency rate projections over time against the benchmark—which will be the Lasso regression on individual FRED variables. This serves as a benchmark to illustrate the potential advantages not only of pre-processing data with principal components before using a Lasso model, but also of using nonlinear ANN learning structures over the linear Lasso framework. Table 4 provides the RMSE for each model over the full forecast period and also stratified into three sub-periods: 2001-2006, 2007-2012, and 2013-2018. These sub-periods were chosen to document model performance over three consecutive 5-year time intervals. We note the data from FRED for the Lasso and neural network models has 10,646 variables and 1,905,870 observations,

with 185 principal components, respectively, for the one-, two-, and four-quarter ahead forecasts, respectively.

As reflected in Figure 8 and Table 4, the ANN forecast notably outperforms both Lasso models over a four-quarter horizon, having an RMSE that is slightly more than 35 percent smaller than that of either Lasso model. While the ANN has a similar, but notable advantage over the 2001-2007 and 2013-2018 sub-sample forecast periods, its forecast accuracy is relatively stronger in the 2007-2012 period that brackets the Great Recession and the Global Financial Crisis. This pattern is sensible given that the ANN is better suited to track nonlinear behavior than the Lasso framework and given the highly nonlinear nature of mortgage defaults and of recessions. In this respect, this pattern is consistent with the loan-level analysis of Sirignano et al. (2018), which also documents superior performance by ANN models. One other noteworthy pattern is that the Lasso using principal components tended to yield more accurate forecasts, implying that the pre-processing step of deriving principal components is valuable in deriving information from underlying variables.

When the forecast horizon is shortened to two- and then one-quarter ahead, the relative superiority of the ANN forecasts is less pronounced over that of the two Lasso models. Indeed, compared to the Lasso model using principal components as inputs, the reduction in RMSE is only 0.01 for each of these horizons. That said, from a regulatory point of view, the strong superiority of the ANN approach at a four-quarter horizon is more relevant insofar as more advanced warning is more helpful for several important reasons. First, regulators would have more time to demand lenders to build up capital buffers. Second, lenders would have more time to build up capital and loan loss reserves ahead of loan quality problems. Finally, in advance of a deterioration in loan

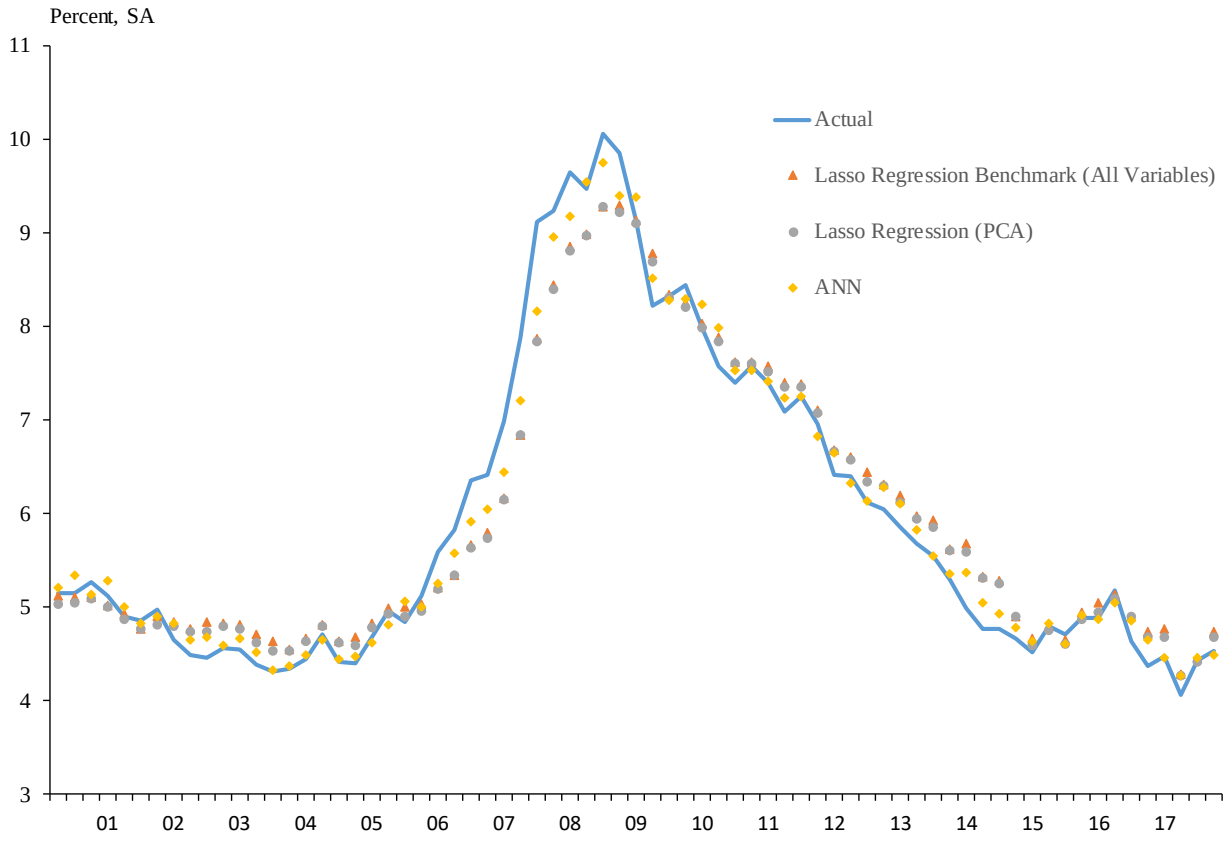

Figure 8: Forecasts of the 30-90 Day Mortgage Delinquency Rate over 2001-2018.
(Sources: FRED and authors' calculations.)

	4Q Ahead				1Q ahead	2Q ahead
Source	**RMSE 2001-18**	**RMSE 2001-06**	**RMSE 2007-12**	**RMSE 2013-18**	**RMSE 2001-18**	**RMSE 2001-18**
Neural Networks	0.24	0.14	0.36	0.17	0.09	0.15
Lasso, PCA	0.38	0.21	0.56	0.28	0.10	0.16
Lasso all variables	0.39	0.23	0.55	0.31	0.12	0.18

Table 4: Forecast Errors By Time Period
(Sources: FRED and authors' calculations.)

quality that could trigger a financial crisis or credit crunch, policymakers would have more advanced waring to set up liquidity and capital backstops, and to build consensus for and enact fiscal or monetary policy actions to stabilize the macro-economy if warranted.

6.4 Categories of FRED that Most Contribute to Mortgage Delinquency Rate Forecasts

This subsection takes a deeper look and examines which types of variables have the biggest impact on unemployment rate forecasts. We use the neural network model for this as it has the best performance as shown earlier, and we treat the ANN based on principal components from all variables as the baseline.

Table 5 reports which types of variables are most informative at a four-quarter horizon, showing how much the RMSE rose above that of the full baseline model when each category is excluded from the input data. For convenience, Table 6 lists the three most important categories for each forecast time period. The biggest loss over the full and 2007-12 samples occurs if Money, Banking, and Finance variables are excluded, consistent with either such variables picking up vulnerabilities from building indebtedness or the ability of some financial variables to more quickly reflect expectations of future events. Not surprisingly, given the impact of income and labor shocks in severe downturns, the "Production & Business Activity" and Population, Employment, & Labor categories, respectively, were the second and third most important during this period dominated by the Great Recession. Interestingly, over the 2001-2006 sub-period the biggest loss in fit occurred when international data are excluded. It is unclear if this might reflect how globalization affected U.S. labor markets—and thereby mortgage quality—especially after China entered the WTO in the early 2000s (see Autor et al., 2016). It may also reflect how capital inflows to the U.S. may have helped fund the mortgage boom of that sub-period.

Another interesting sub-period pattern is that the Money, Banking, and Finance was not among the most informative during the 2013-18 period of recovery. This may reflect the adoption of financial reforms that may have curtailed a large upswing in mortgage lending or stabilized loan quality expectations, both of which could work to limit the relative marginal information provided

Category Taken Out of Model	RMSE 2001-18	RMSE 2001-06	RMSE 2007-12	RMSE 2013-18
Academic Data	N/A	N/A	N/A	N/A
International Data	0.03	0.06	0.02	0.05
Money,Banking,Finance	0.05	0.01	0.08	0.02
National Accounts	0.00	0.01	-0.05	0.08
Population, Employment, Labor Markets	0.03	0.00	0.06	-0.02
Prices	0.01	-0.01	0.00	0.05
Production and Business Activity	0.03	0.01	0.06	-0.01
Regional Data	0.00	0.01	0.00	0.01

Table 5: Dropped Category Results with Respect to the Baseline SPF Forecast

(Sources: FRED and authors' calculations. "+/- above baseline" column compares the best base neural network RMSE with that of the model with the omitted category. "+/- above baseline"=RMSE omitted category- RMSE base. The RMSE base has a value of 0.24 from before. This measures the increase in error of omitting a category with respect to the benchmark. In general, the higher the error, the larger the impact the omitted variable category has on mortgage delinquency projections.)

Time Period	Most Influential Category	2nd Most Influential Category	3rd Most Influential Category
Overall	Money,Banking,Finance	International	Production and Business Activity
2001-06	International	Money,Banking,Finance	National Accounts
2007-12	Money,Banking,Finance	Production & Business Activity	Population, Employment, and Labor
2013-18	National Accounts	International Data	Prices

Table 6: Summary of Categories Having the Largest Impact

(Sources: FRED and authors' calculations. These are with respect to the baseline calculation)

by this category. During the recovery sub-period of 2013-2018, variables from the "National Accounts" category provided the most marginal information. With financial activity relatively stabilized by new financial reforms, the pace of recovery in incomes as tracked by this category could plausibly rise in importance. Second and third in importance are the categories International followed by Prices, respectively. This could reflect that swings in the growth of the global economy were pronounced in this period, and that the Federal Reserve especially emphasized that it would pay much attention to restoring inflation toward its 2 percent target.

The non-PCA Lasso can assess which individual variables are most important. For four-quarter ahead forecasts, the five most informative, in order, were: net interbank borrowing by U.S. depository institutions, net lending or borrowing by the household sector, net accounts receivable of domestic finance companies, the four-week average of initial claims for unemployment, and the gross accounts of domestic finance companies. The first may pick up shifts in expectations associated different phases of the crisis that induced swings in interbank lending, while the second reflects shifts in how much households could borrow. The third and fifth categories likely reflect how the swings in the expansion and contraction of the shadow bank sector mirrored concerns about the health of the financial system, which markets ascribed to real estate loan losses. In terms of double-trigger models of mortgage default (see Bhutta, Dokko, and Shan, 2017), these four categories could plausibly reflect either expectations of future mortgage quality or how much the financial sector allowed households to get into negative net housing equity or both. The fourth ranked factor—smoothed initial claims for unemployment—plausibly foreshadows the impact of labor shocks on loan quality—the second part of the double trigger mechanism driving mortgage defaults.

7. Conclusion

This paper forecasts the mortgage delinquency rate four-quarters ahead using machine-learning techniques applied to wide-ranging variables available from FRED. Models estimate relationships in-sample over 1970-2000 and then forecast the delinquency rate quarterly since 2001. The training window is updated each quarter to include new data. Rolling and non-rolling forecasts are tested for the training window. We find that a non-rolling neural network model performs best and outperforms a Lasso regression approach, and that pre-processing data with principal components helps improve the performance of Lasso models.

From experiments dropping broad categories of FRED, "Money, Banking, and Finance" data were the most important in forecasting the delinquency rate over the full forecast sample, especially over the 2007-2012 period that was dominated by the Great Recession. Income related categories encompassing variables from the national GDP accounts and labor markets were also important. From Lasso models, financial and unemployment variables were among the most important. This pattern is loosely consistent with financial variables quickly picking up expectations of future real estate losses and, perhaps also with financial and labor market data reflecting the factors associated with the double-trigger mechanisms in models of mortgage default. These findings imply that if one were to create a forecast with significantly fewer variables, retaining variables from these categories would maximize prediction accuracy.

Our forecasting method can be efficiently implemented. To generate new forecasts, an SQL database can be used to update variables and produce new forecasts in minutes. Among potential improvements that we plan to test additional network configurations and consider additional macroeconomic variables not included in FRED, such as measures of geopolitical risk and consumer confidence.

References

Aikman, David, Michael Kiley, Seung Jung Lee, Michael G Palumbo, and Missaka Warusawitharana. 2017. "Mapping Heat in the US Financial System." *Journal of Banking and Finance* 81 (C): 36-64.

Autor, David H., David Dorn, and Gordon H. Hanson. 2016. "The China Shock: Learning from Labor Market Adjustment to Large Changes in Trade," *Annual Review of Economics* 8(1): 205-40.

Bhutta, Neil, Jane Dokko, and Hui Shan. 2017. "Consumer Ruthlessness and Mortgage Default During the 2007 to 2009 Housing Bust." *Journal of Finance* 72 (6): 2433-2466.

Cook, Thomas, and Aaron Hall. 2017. "Macroeconomic Indicator Forecasting with Deep Neural Networks." *Research Working Paper RWP 17-11*, Federal Reserve Bank of Kansas City, www.kansascityfed.org/publications/research/rwp/articles/2017/macroeconomic-indicator-forecasting-deep-neural-networks

Crawford, Jesse. 2009. "Principal Components." *Tarleton State University*, Tarleton State University, 2009, faculty.tarleton.edu/crawford/documents/math505/PrincipalComponents.pdf.

Federal Reserve Economic Data (FRED). 2019. Federal Reserve Bank of St. Louis, www.fred.stlouisfed.org

Fonti, Valeria. 2017. *Feature Selection Using LASSO - VU*. University of Amsterdam, March. beta.vu.nl/nl/Images/werkstuk-fonti_tcm235-836234.pdf.

Gai, Prasanna, Andrew Haldane, and Sujit Kapadia. 2011. "Complexity, Concentration, and Contagion." *Journal of Monetary Economics* 58 (5): 453-470.

Giannone, Domenico, Lucrezia Reichlin, and David Small. 2008. "Nowcasting: The Real-Time Informational Content of Macroeconomic Data." *Journal of Monetary Economics* 55 (4): 665-676.

Jung, Jin-Kyu, Manasa Patnam, and Anna Ter-Martirosyan. 2018. “An Algorithmic Crystal Ball: Forecasts-based on Machine Learning.” IMF Working Paper No. WP/18/230.

Kreiner, Aaron and John V. Duca. Forthcoming. “Can Machine Learning on Economic Data Better Forecast the Unemployment Rate?” *Applied Economics Letters*.

Kriesel, David. 2009. *A Brief Introduction to Neural Networks*. Self-Published (PhD in Data Science).

Kvamme, Havard, Nikolai Sellereite, Kjersti Aas, K., and Steffen Sjursen. 2018. “Predicting Mortgage Default Using Convolutional Neural Networks.” *Elsevier Expert Systems with Applications* 102: 207-217.

Mullainathan, Sendhil, and Jann Spiess. 2017. “Machine Learning: An Applied Econometric Approach.” *Journal of Economic Perspectives* 31 (2): 87-106.

Ponomareva, Ksenia, Paul Epstein, and David Knight (2018). “Predicting Mortgage Loan Delinquency Status with Neural Networks.” https://www.nr.no/directdownload/1552287038/MortgageDefaultKvamme18.pdf

Prasad, Shivan and Ashok K. Sinha. 2018. “An Econometric Model of GDP Using Machine Learning Through an Artificial Neural Network.” *International Journal of Computer Applications* 181 (28): 24-27.

Sirignano, Justin, Apaar Sadhwani, and Kay Giesecke. 2016. “Deep Learning for Mortgage Risk." ArXiv e-prints http://adsabs.harvard.edu/abs/2016arXiv160702470S.