Modelling Systemic Risk using Neural Network Quantile Regression

Master Thesis submitted

to

Prof. Dr. Weining Wang

and

Prof. Dr. Wolfgang Karl Härdle

Humboldt-Universität zu Berlin
School of Business and Economics
Ladislaus-von-Bortkiewicz Chair of Statistics

by

Georg Keilbar
(546268)

in partial fulfillment of the requirements
for the degree of
Master of Science in Economics

Berlin, July 9, 2018.

Abstract

We propose a novel approach to estimate the conditional value at risk (CoVaR) of financial institutions. Our approach is based on neural network quantile regression. Building on the estimation results we model systemic risk spillover effects across banks by considering the marginal effects of the quantile regression procedure. We obtain a time-varying risk network represented by an adjacency matrix. We then propose three measures for systemic risk. The *Systemic Fragility Index* and the *Systemic Hazard Index* are measures to identify the most vulnerable and most critical firms in the financial system, respectively. As a third risk measure we propose the *Systemic Network Risk Index* which represents the overall level of systemic risk. We apply our methodology to the global systemically relevant banks from the United States in a time period from 2007 until 2018. Our results are similar to previous studies about systemic risk. We find that systemic risk increased sharply during the height of the financial crisis in 2008 and again after a short period of easing in 2011 and 2015.

Keywords: Systemic Risk, CoVaR, Neural Networks, Quantile Regression

Contents

List of Abbreviations

ATAE	Average tilted absolute error
CoVaR	Conditional value at risk
G-SIB	Global systemically important bank
iid	independent and identically distributed
NNQR	Neural network quantile regression
NYSE	New York Stock Exchange
ReLU	Rectifier linear unit
RQEX	Ratio of quantile exceedances
SFI	Systemic Fragility Index
SHI	Systemic Hazard Index
SNRI	Systemic Network Risk Index
VaR	Value at risk

List of Figures

List of Tables

1 Introduction

The issue of systemic risk attracted a lot of attention from academics as well as from regulators in the aftermath of the financial crisis of 2007-2009. Systemic risk refers to banks and other economic agents with substantial importance to the financial system due to their size (*too big to fail*) or their centrality (*too interconnected to fail*). Conventional quantitative risk measures such as value at risk (VaR) are not suitable for capturing systemic risk adequately.

To tackle these issues, Adrian et al. (2016) [1] came up with conditional value at risk (CoVaR), a systemic extension of VaR. Their original approach is however restricted to analyze systemic risk in a bivariate context. Thus, Hautsch et al. (2014) [14] and Härdle et al. (2016) [10] extended the CoVaR framework further to analyze systemic risk in a multivariate and nonlinear context.

This master thesis offers a novel approach for the estimation of CoVaR using neural network quantile regression. Neural networks have become one of the most popular tools for prediction in recent years. They have been employed extensively and successfully to image classification as well as speech recognition problems. Our neural network based approach is highly suited for estimating CoVaR due to its flexibility and nonparametric nature. Also it allows for a multivariate context.

In a first step we estimated the VaR for each global systemically important financial institution (G-SIB) from the United States by regressing their stock returns on a set of risk factors using linear quantile regression. Next we estimated the CoVaRs of the same firms using neural network quantile regression. Here we regressed the stock returns of each bank on the stock returns of the remaining banks. By approximating the conditional quantile with a neural network we ensure to capture possible nonlinear effects. In order to estimate risk spillover effects across banks we calculated the marginal effects by taking the derivative of the fitted quantile with respect to the other banks' stock returns, evaluated at their VaR. By doing so we came up with a network of spillover effects represented by an adjacency matrix. This adjacency matrix is time-varying, i.e. we estimated a network for each trading day.

In a final step we proposed three systemic risk measures building on the previous results. As a first measure we proposed the *Systemic Fragility Index* which identifies the most vulnerable banks in a given financial risk network. The second measure is the *Systemic Hazard Index* which identifies the financial institutions which impose the biggest threat to the financial system. These two measures characterize the firm-specific aspects of systemic risk. Thus we proposed a third measure which estimates the total level of systemic risk, the *Systemic Network Risk Index*.

Our estimation results show that systemic risk increased sharply during the height of the financial crisis after the bankruptcy of Lehman Brothers in 2008. Systemic risk has stabilized over the last years with two minor spikes in 2011 and 2016. We have also identified the most systemically relevant financial institutions during the financial crisis.

The remainder of this thesis is organized as follows. Section 2 provides a brief introduction to neural networks in general and neural network quantile regression in particular. Section 3 describes in detail the methodology of this master thesis. After establishing the research framework step by step, we present the results in section 4. Section 5 discusses the results and concludes.

2 Neural Networks

2.1 Architecture of Neural Networks

A neural network is a nonlinear input-output model inspired by the processing of biological neurons in the human nervous system (Kuan et al., 1994 [22]). Mathematically, neural networks can be represented by a function, $f : \mathbb{R}^p \to \mathbb{R}^q$, mapping from the input to the output space:

$$(y_1, \ldots, y_q)^\top = f(x_1, \ldots, x_p). \tag{2.1}$$

Neural networks have a multiple-layer structure of directed graphs with one input layer, one or several hidden layer(s) and one output layer. While the input and output layers contain the input and output variables, respectively, the hidden layers function as intermediaries between those two. Neural networks are called feedforward since the directed graphs are acyclical.

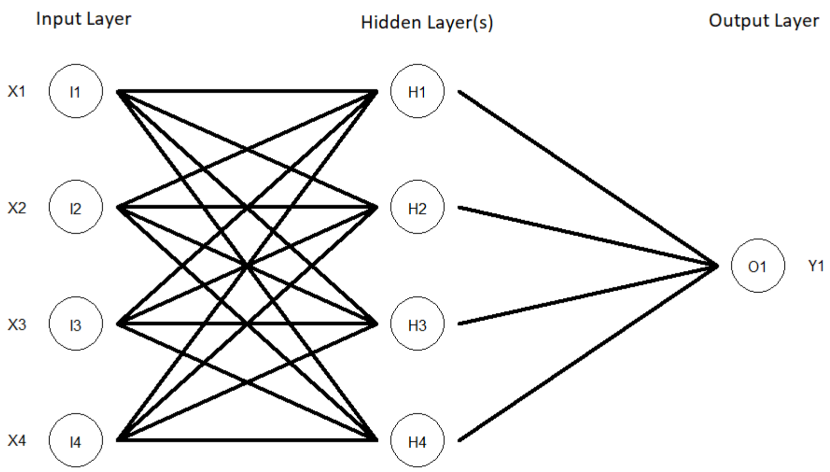

Figure 1: Neural network with a single hidden layer.

An individual neuron is displayed by one node within a neural network. It can be represented by a function, $g : \mathbb{R}^I \to \mathbb{R}$, mapping from the input space to the one-dimensional output space. Such a function has three tasks: weighting, aggregation and transformation of inputs. The optimal choice of weights will be explained in the next subsection. The aggregation is usually done by summation of weighted inputs. To introduce nonlinear effects, the aggregate is then transformed by an activation function, which often has a sigmoid shape (e.g. $\tanh(z)$). In the recent past the rectifier linear unit (ReLU) function, $\max(0, z)$, became the most popular choice (Glorot et al., 2011) [9].

Neural networks are suitable for function approximation, as they have a high degree of

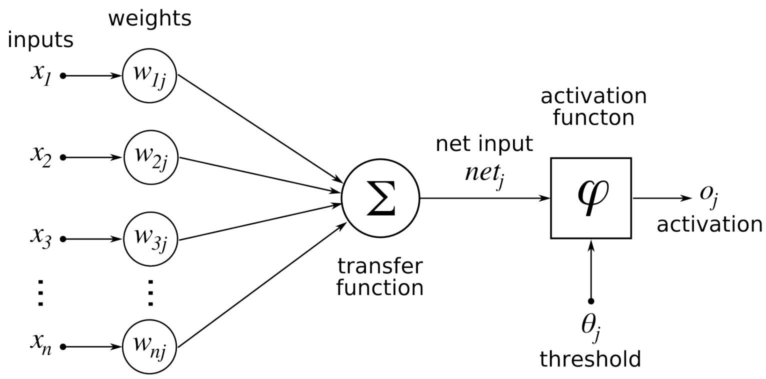

Figure 2: Architecture of a single neuron in a neural network [5].

flexibility since no parametric structure has to be assumed on the functional relationship. The universal approximation theorem states that a feedforward neural network with at least one hidden layer and a finite number of hidden nodes is able to approximate any continuous function under some mild conditions on the activation function. Cybenko (1989) [6] provides a proof for the case of sigmoid activation functions.

Universal Approximation Theorem. *Let σ be any continuous sigmoidal function. Then finite sums of the form*

$$G(x) = \sum_{j=1}^{n} \alpha_j \sigma(\gamma_j^\top x + \theta_j)$$

are dense in $C(I_n)$. In other words, given any $f \in C(I_n)$ and $\epsilon > 0$, there is a sum $G(x)$ in the above form, for which

$$|G(x) - f(x)| < \epsilon \quad \textit{for all } x \in I_n.$$

While the universal approximation theorem makes a strong statement about the possibility of function approximation, it is silent on how to find such an approximation. In particular it is silent about the required number of hidden nodes. And in general, the universal approximation theorem does not guarantee good results in practical applications with limited data. Therefore the next subsection will explain methods for obtaining weights for a neural network.

Neural networks can be understood as a generalization of a standard regression problem (in the case of a continuous output variable) or a standard classification problem (in the case of a discrete output variable). But instead of using a linear equation the dependence of the output on the inputs is explained by a neural network. Nonlinearity is hereby introduced by the nonlinear transformation within the individual neurons and by the multiple-layer structure of the network. Linear regression is equivalent to neural network regression if the activation function is the identity for all nodes.

2.2 Learning in Neural Networks

The learning process of feedforward neural networks belongs to the paradigm of supervised learning (Hastie et al., 2009 [12]), so the teaching inputs include observed values of the independent variables as well as of the dependent variable.

For the training of neural networks linear optimization methods are no longer feasible. The most widely used estimation method for neural networks is the backpropagation algorithm (Werbos, 1974 [24] and Hinton et al., 1986 [15]). Backpropagation is an optimization method based on gradient descent.

Consider a neural network with one hidden layer, one output node, an activation function $g(\cdot)$ and a quadratic loss function L,

$$L(y, \widehat{y}) = \sum_{1=1}^{n} \frac{1}{2}(y - \widehat{y})^2, \tag{2.2}$$

where y is the observed and $\widehat{y}$ is the fitted value. The initial weights of the neural network are chosen randomly but should be close to zero. The output is then obtained by passing the inputs *forward* through the network using the initial weights. Now the error can be calculated at the output node and also the gradient of the loss function with respect to the weights.

For the output layer the weight from a hidden node h is adjusted proportionally to its derivative:

$$\Delta w_h^o = -\eta \frac{\partial L}{\partial w_h^o} \tag{2.3}$$

$$= -\eta(y - \widehat{y}) g' \left(\sum_{m=1}^{M} w_m^h o_m \right) o_h, \tag{2.4}$$

with o_h being the output of the hidden layer neuron h, M is the number of hidden nodes and η is the learning rate.

Hidden layer weights are also adjusted according to the gradient. Since the error is brought *backward* through the network the weight adjustment depends on the adjustments at subsequent nodes. The weight between an input j and hidden node h is adjusted in the following way:

$$\Delta w_{j,h}^h = -\eta \frac{\partial L}{\partial w_{j,h}^h} \tag{2.5}$$

$$= -\eta w_h^o \delta_o g' \left(\sum_{i=1}^{K} w_{i,h}^h x_i \right) x_j, \tag{2.6}$$

where

$$\delta_o = (y - \widehat{y})g'\left(\sum_{m=1}^{M} w_m^h o_m\right) \tag{2.7}$$

and x_j represents the j-th input and K is the number of input variables.

This procedure of propagation of the error and subsequent weight adjustment is repeated until a stopping criterion is fulfilled. This can be a predefined number of maximal iterations or a threshold for the loss function.

Algorithm 1 Backpropagation

```
Initialize network weights randomly
repeat
    for all Training examples do
        Propagation of the input to obtain the output of the neural network
        Calculate the error
        Pass the error back through the network
        Update the weights according to the gradient of the error
until No. iterations > max no. of iterations OR other stopping criterion
```

In standard gradient descent the whole sample is used to calculate the gradient based on the forward propagation of inputs. However, when the sample size becomes large, this is no longer efficient. Therefore it might be preferable to consider mini batches (randomly selected cases) for each iteration of forward- and backpropagation (Hardt et al., 2016 [11]). This method is called stochastic gradient descent.

The practical problem of gradient-based methods is that it might be difficult to find a global minimum. This problem can be induced by the existence of local minima and saddle points. The algorithm might stop early if the gradient of the error is close to zero, i.e. a marginal change in the weights does not have a significant impact on the error.

Several optimization algorithms have come up with solutions to these practical problems. A possible remedy is to consider an adaptive learning rate η for every parameter and at each time step. Another method is to introduce momentum for the learning rate in order to avoid being stuck in local minima. Current gradient-based optimization algorithms are ADADELTA (Zeiler, 2012 [25]) and ADAM (Ba, 2015 [2]).

2.3 The Bias-Variance Trade-off

A central issue of neural networks and machine learning in general is overfitting. One has to be very careful with the choice of tuning parameters, such as the number of hidden layers or the number of hidden nodes. If the architecture of the neural network is too complex, there is a tendency to not only fit the structure of the data but also the noise. As a consequence, the training error can be reduced to zero but the model typically generalizes poorly.

Predictive accuracy can be measured by the *expected prediction error* (Hastie et al., 2009 [12]):

$$\mathrm{Err} = \mathrm{E}\left[L\left\{Y, f(X)\right\}\right], \tag{2.8}$$

where $L(\cdot)$ is an arbitrary loss function. However, neural networks minimize only the training error which is an overly optimistic approximation of the actual expected prediction error. This optimism lies in the repeated use of the data for estimation and evaluation.

$$\overline{\mathrm{err}} = \frac{1}{N}\sum_{i=1}^{N} L\left\{Y_i, \widehat{f}(X_i)\right\} \tag{2.9}$$

Under the assumption of a quadratic loss function and by fixing the covariates ($X = x_0$), the expected prediction error can be decomposed in the following way:

$$\mathrm{Err}(x_0) = \sigma_\epsilon^2 + \mathrm{Bias}^2\left\{\widehat{f}(x_0)\right\} + \mathrm{Var}\left\{\widehat{\mathrm{f}}(\mathrm{x}_0)\right\}. \tag{2.10}$$

Predictive accuracy is thus determined by the bias as well as the variance of the model. The first term, σ_ϵ^2, is the irreducible error which is independent of any model.

Neural networks with a sufficiently complex structure are able to reduce the bias to zero at the expense of a high variance. A simple way to mitigate this problem is to introduce an additional penalty term for model complexity. Common choices are weight decay (L^2) and LASSO (L^1) penalization on the weights.

Hastie et al. (2005) [13] propose elastic net, a L^1/L^2 hybrid penalization. When considering an arbitrary loss function $L(\cdot)$, the optimization problem becomes:

$$\min_\theta L(\theta) + \lambda\left\{(1-\alpha)\|\theta\|_1 + \alpha\|\theta\|_2^2\right\}, \tag{2.11}$$

where θ is a vector of parameters. If $\alpha = 0$, elastic net is identical to LASSO, if $\alpha = 1$, it is identical to weight decay penalization. The advantage of elastic net is that it achieves the sparsity property of the LASSO and works well with highly correlated regressors. However, feature selection is not really possible in the context of neural networks, due to their multiple-layer structure. A particular input variable has multiple weights and a shrinkage of one weight to zero does not eliminate the whole variable but only one connection of the neural network.

The currently most prevalent regularization method for neural networks is dropout (Hinton et al., 2014 [16]). The idea is to randomly drop units with a probability p and adjust the weights for a thinned-out network. The final trained model can then be seen as an ensemble of less complex neural networks. Dropout tries to mitigate the risk of co-adaptation of weight parameters, which can be the cause for overfitting (Hinton et al., 2012 [17]).

2.4 Neural Network Quantile Regression

Neural networks have been applied extensively to mean regression and classification problems. Neural networks can also be used for approximating quantiles. This was first formalized by Taylor (2000) [23]. An application to time series data can be found in Xu et al. (2016) [18], who extended the CaViaR framework of Engle et al. (2004) [8] by using neural network quantile regression instead of standard quantile regression.

Conceptually, the approach is similar to linear quantile regression, as introduced by Koenker et al. (1978 [19], 1982 [20]). The goal is to find the best approximation for the conditional quantile of a random variable. Therefore the following loss function has to be minimized:

$$\min_{\theta} \sum_{t=1}^{T} \rho_\tau \left\{ y_t - \widehat{y}_t^\tau(\theta) \right\}, \tag{2.12}$$

where θ is a vector of coefficients, y_t are the observed values and $\widehat{y}_t^\tau(\theta)$ are the fitted τ-quantiles and ρ_τ is the tilted absolute error function defined as:

$$\rho_\tau(u) = \begin{cases} \tau u & \text{if } u \geq 0 \\ (\tau - 1)u & \text{if } u < 0 \end{cases}, \tag{2.13}$$

Since the backpropagation algorithm requires differentiability, Taylor (2000) proposes a slight modification of the loss function:

$$\rho_\tau(u) = \begin{cases} \tau h(u) & \text{if } u \geq 0 \\ (\tau - 1)h(u) & \text{if } u < 0 \end{cases}, \tag{2.14}$$

with $h(u)$ being the Huber norm, a hybrid $L^1/L2$ norm, defined as:

$$h(u) = \begin{cases} \frac{u^2}{2\epsilon} & \text{if } 0 \leq \|u\| \leq \epsilon \\ \|u\| - \frac{\epsilon}{2} & \text{if } \|u\| > \epsilon \end{cases}. \tag{2.15}$$

The functional form of a neural network quantile regression model with one hidden layer is given by

$$f(X_t, \mathbf{w}) = \sum_{m=1}^{M} w_m^o \; g\left(\sum_{k=1}^{K} w_{k,m}^h X_{k,t} + b_m^h\right) + b^o, \tag{2.16}$$

where $w_{k,m}^h$ is the weight from input k to hidden node m, w_m^h is the weight of hidden node m and b_m^h and b^o are the corresponding bias terms for the hidden nodes and the output node. $g(\cdot)$ is a nonlinear hidden layer activation function. The activation function for the output node is assumed to be linear.

The estimated conditional τ-quantile is then the fitted value of the neural network:

$$\widehat{Y}_t^\tau = f(x_t, \widehat{\mathbf{w}}), \tag{2.17}$$

where $\widehat{\mathbf{w}}$ is the solution of the backpropagation procedure and x_t is the vector of all inputs at time t.

Neural network quantile regression as defined above was implemented in R in the `QRNN` package (Cannon, 2011) [3]. The package also allows for the inclusion of a L^2 penalty term. Recent software packages do not require differentiability of the loss function and also enables the consideration of more complex neural networks.

3 Methodology

In this section we explain the details of our systemic risk analysis. Our methodology involves four steps. The first step is concerned with the estimation of VaR with linear quantile regression using a set of risk factors as explaining variables. The results are used in the next step to estimate the CoVaR for each financial institution using neural network quantile regression. Next we calculate marginal effects to model systemic risk spillover effects, resulting in a time-varying systemic risk network. In the final step we propose three systemic risk measures based on this systemic risk network.

3.1 Step 1: Estimation of VaR

VaR is defined as the maximum loss over a fixed time horizon at a certain level of confidence. Mathematically, it is the τ-quantile of the profit and loss distribution:

$$\mathrm{P}(X_{i,t} \leq \mathrm{VaR}^{\tau}_{i,t}) = \tau, \tag{3.1}$$

where $X_{i,t}$ is the return of firm i at time t and $\tau \in (0,1)$ is the quantile level.

The VaR of each firm i is estimated as the fitted value of a linear quantile regression procedure by regressing the returns on a set of macro state variables M_{t-1}.

$$X_{i,t} = \alpha_i + \gamma_i M_{t-1} + \epsilon_{i,t}, \tag{3.2}$$

where the conditional quantile of the error term $\epsilon^{\tau}_{i,t}|M_{t-1} = 0$. The VaR is the fitted value of the linear quantile regression problem:

$$\mathrm{VaR}^{\tau}_{i,t} = \widehat{\alpha}_i + \widehat{\gamma}_i M_{t-1}. \tag{3.3}$$

3.2 Step 2: Estimation of CoVaR with NNQR

CoVaR was introduced as a systemic extension for standard VaR (Adrian et al., 2016 [1]). Similar to VaR, it is a risk measure defined as a conditional quantile of the loss distribution. But deviating, CoVaR is contingent on a specific financial distress scenario. The motivation for using CoVaR is the identification of systemically important banks. For the distress scenario we assume that all other firms are at their VaR. By doing this we follow the reasoning of Hautsch et al. (2014) [14] and Härdle et al. (2016) [10].

$$\mathrm{P}(X_{j,t} \leq \mathrm{CoVaR}\,|X_{-j,t} = \mathrm{VaR}^{\tau}_{-j,t}) = \tau, \tag{3.4}$$

where $X_{-j,t}$ is a vector of returns of all firms except j and $\text{VaR}^{\tau}_{-j,t}$ is the corresponding vector of VaRs.

CoVaR can be estimated as a fitted conditional quantile, building on the results for the VaRs obtained in step 1. Chao et al. (2015) [4] and Härdle et al. (2016) [10] find evidence for nonlinearity in the dependence between pairs of financial institutions. Hence, linear quantile regression might not be an appropriate procedure to estimate the risk spillovers. We therefore propose the use of neural network quantile regression. The flexibility of the approach allows to detect possible nonlinear dependencies in the data.

The conditional quantile of bank j's returns is regressed on the returns of all other banks and using a neural network as defined in section 2.4:

$$
\begin{aligned}
X_{j,t} &= f(X_{-j,t}, \mathbf{w}) + \epsilon_{j,t}, && (3.5)\\
&= \sum_{m=1}^{M} w^{o}_{m}\ g\left(\sum_{k\neq j}^{K} w^{h}_{k,m} X_{k,t} + b^{h}_{m}\right) + b^{o} + \epsilon_{j,t}, && (3.6)
\end{aligned}
$$

with the conditional quantile of error term $\epsilon^{\tau}_{j,t}|X_{-j,t} = 0$.

To calculate the CoVaR of firm j, the fitted neural network has to be evaluated at the distress scenario:

$$
\text{CoVaR}^{\tau}_{j,t} = f(\text{VaR}^{\tau}_{-j,t}, \widehat{\mathbf{w}}), \tag{3.7}
$$

where $\widehat{\mathbf{w}}$ is the estimated vector of weights and bias terms. Nonlinearity is introduced by the use of the nonlinear activation function.

3.3 Step 3: Calculation of Risk Spillover Effects

Based on the weights estimated by the NNQR procedure, it is now possible to obtain risk spillover effects between each directed pair of banks. We propose to estimate the spillover effects by taking the first derivative of the conditional quantile of firm j's return with respect to the return of firm i.

$$
\frac{\partial X^{\tau}_{j|-j,t}}{\partial X_{i,t}} = \frac{\partial}{\partial X_{i,t}} \sum_{m=1}^{M} w^{o}_{m}\ g\left(\sum_{k\neq j}^{K} w^{h}_{k,m} X_{k,t} + b^{h}_{m}\right) + b^{o} + \epsilon_{j,t} \tag{3.8}
$$

In the case of a sigmoid tangent activation function we have

$$
\frac{\partial X^{\tau}_{j|-j,t}}{\partial X_{i,t}} = \sum_{m=1}^{M} w^{o}_{m} w^{h}_{i,m}\ g'\left(\sum_{k\neq j}^{K} w^{h}_{k,m} X_{k,t} + b^{h}_{m}\right) \tag{3.9}
$$

with

$$g'(z) = \frac{\partial \tanh(z/2)}{\partial z} \tag{3.10}$$

$$= \frac{2}{(\exp^{-z/2} + \exp^{z/2})^2}. \tag{3.11}$$

In the case of a ReLu activation function we have

$$\frac{\partial X^{\tau}_{j|-j,t}}{\partial X_{i,t}} = \sum_{m=1}^{M} w^o_m w^h_{i,m} \mathbf{1}\left(\sum_{k \neq j}^{K} w^h_{k,m} X_{k,t} + b^h_m > 0 \right). \tag{3.12}$$

Since we are interested in the lower tail dependency, we consider the marginal effect evaluated at the distress scenario as defined in the previous subsection:

$$\frac{\partial \, \mathrm{CoVaR}^{\tau}_{\mathrm{j,t}}}{\partial \, \mathrm{VaR}^{\tau}_{i,t}} = \sum_{m=1}^{M} w^o_m w^h_{i,m} \; g' \left(\sum_{k \neq j}^{K} w^h_{k,m} \, \mathrm{VaR}_{k,t} + b^h_m \right). \tag{3.13}$$

Calculating such a marginal effect for each directed pair of firms yields an off-diagonal adjacency matrix of risk spillover effects at time t:

$$A_t = \begin{pmatrix} 0 & a_{12,t} & \dots & a_{1K,t} \\ a_{21,t} & 0 & \dots & a_{2K,t} \\ \vdots & \dots & \ddots & \vdots \\ a_{K1,t} & a_{K2,t} & \dots & 0 \end{pmatrix}, \tag{3.14}$$

with elements defined as absolute values of marginal effects:

$$a_{ji,t} = \begin{cases} |\frac{\partial \, \mathrm{CoVaR}^{\tau}_{j,t}}{\partial \, \mathrm{VaR}^{\tau}_{i,t}}|, & \text{if } j \neq i \\ 0, & \text{if } j = i \end{cases}. \tag{3.15}$$

Note that the risk spillover effects are not symmetric in general, thus $a_{ji,t} \neq a_{ij,t}$. This adjacency matrix specifies a weighted directed graph modeling the systemic risk in the financial system.

3.4 Step 4: Network Analysis of Spillover Effects

To further analyze the systemic relevance of the financial institutions we can calculate several network measures proposed by Diebold et al. (2014) [7].

First, the total directional connectedness *to* firm j at time t is defined as the sum of absolute marginal effects of all other firms on j.

$$C_{j\leftarrow\cdot,t} = \sum_{i=1}^{K} a_{ji,t} \tag{3.16}$$

Analogously, one can define the total directional connectedness *from* firm i at time t as the sum of absolute marginal effects from i to all other firms.

$$C_{\cdot\leftarrow i,t} = \sum_{j=1}^{K} a_{ji,t} \tag{3.17}$$

Lastly, Diebold et al. define the total connectedness at time t as the sum of all absolute marginal effects.

$$C_t = \frac{1}{K} \sum_{i=1}^{K} \sum_{j=1}^{K} a_{ji,t} \tag{3.18}$$

The total connectedness is a measure for the interconnectedness on the level of the entire system, without differentiating between individual components of the network.

Building on the analysis of Diebold et al., we refine the approach by incorporating VaR and CoVaR in the measurement of the systemic relevance. In particular, we propose the *Systemic Fragility Index (SFI)* and the *Systemic Hazard Index (SHI)*:

$$SFI_{j,t} = \sum_{i=1}^{K} \left(1 + |\,\text{VaR}_{i,t}^{\tau}\,|\right) \cdot a_{ji,t} \tag{3.19}$$

$$SHI_{i,t} = \sum_{j=1}^{K} \left(1 + |\,\text{CoVaR}_{j,t}^{\tau}\,|\right) \cdot a_{ji,t} \tag{3.20}$$

The *SFI* is a systemic risk measure for the vulnerability of a financial institution. It increases if those adjacency weights pointing to j are large and also if the VaRs of firms i (i.e. the risk factors for j) increase.

The *SHI* is a risk measure for the exposure of the financial system to firm i. It depends on the out-going adjacency weights from i and also on the other firm's CoVaR.

As a third index we propose the *Systemic Network Risk Index (SNRI)*, a measure for the total systemic risk in the financial system which depends on the marginal effects, the outgoing VaRs and the incoming CoVaRs.

$$SNRI_t = \sum_{i=1}^{K}\sum_{j=1}^{K}(1 + |\,\text{VaR}_{i,t}^{\tau}\,|) \cdot (1 + |\,\text{CoVaR}_{j,t}^{\tau}\,|) \cdot a_{ji,t}. \tag{3.21}$$

Lastly, we define the adjusted adjacency matrix,

$$\widetilde{A}_t = \begin{pmatrix} 0 & \widetilde{a}_{12,t} & \dots & \widetilde{a}_{1K,t} \\ \widetilde{a}_{21,t} & 0 & \dots & \widetilde{a}_{2K,t} \\ \vdots & \dots & \ddots & \vdots \\ \widetilde{a}_{K1,t} & \widetilde{a}_{K2,t} & \dots & 0 \end{pmatrix}, \tag{3.22}$$

with elements defined as:

$$\widetilde{a}_{ji,t} = \begin{cases} a_{ji,t} \cdot \text{VaR}_{i,t}^{\tau} \cdot \text{CoVaR}_{j,t}^{\tau}, & \text{if } j \neq i \\ 0, & \text{if } j = i \end{cases}. \tag{3.23}$$

The adjusted adjacency matrix accounts for the level of outgoing VaRs and incoming CoVaRs and is an improved representation of risk spillover effects. Systemic spillover effects are thus determined by the marginal effects of the NNQR procedure as well as by the VaRs and CoVaRs.

4 Results

4.1 Data

The data contains stock returns for the *global systemically important banks (G-SIBs)* from the United States selected by the Financial Stability Board (FSB) in the time period between January 4, 2007 and May 31, 2018. The daily stock returns are obtained from Yahoo Finance.

Financial Institution	NYSE symbol
Wells Fargo & Company	WFC
JP Morgan Chase & co.	JPM
Bank of America Corporation	BAC
Citygroup	C
The Bank of New York Mellon Corporation	BK
State Street Corporation	STT
Goldman Sachs Group, Inc.	GS
Morgan Stanley	MS

Table 1: List of G-SIBs in the USA.

Additionally to these return data, we consider daily observations of the following set of macro state variables.

i) implied volatility index (VIX), from Yahoo Finance;

ii) the weekly S&P500 index returns, from Yahoo Finance;

iii) Moody's Seasoned Baa Corporate Bond Yield Relative to Yield on 10-Year Treasury Constant Maturity from Federal Reserve Bank of St. Louis;

iv) 10-Year Treasury Constant Maturity Minus 3-Month Treasury Constant Maturity from Federal Reserve Bank of St. Louis.

4.2 Model Selection

The tuning parameters for the neural network quantile regression procedure are selected in the following way. For each financial institution we regress the returns on the other firms' returns to estimate the 5% quantile. The data is separated into a training and a validation set repeatedly in a moving window approach. We consider an estimation window of 250 days and a subsequent validation window of 50 days. Start of each estimation window is the begin of the new year. The resulting performance indicators are then aggregated over all firms and all windows to select the best model.

As a first measure for model performance we propose the average tilted absolute error of prediction ($ATAE$) which can be compared to the MSE in mean regression:

$$ATAE = \frac{1}{T}\sum_{t=1}^{T} \rho_\tau(X_{j,t} - \widehat{X}_{j|-j,t}), \tag{4.1}$$

where $X_{j,t}$ is the observed and $\widehat{X}^\tau_{j|-j,t}$ the fitted value. A small value for the $ATAE$ is preferred.

The second measure is the R^1 criterion (Koenker et al., 1999 [21]), a coefficient of determination defined analogously to the R^2 measure in mean regression.

$$R^1 = 1 - \frac{\sum_{t=1}^{T} \rho_\tau(X_{j,t} - \widehat{X}^\tau_{j|-j,t})}{\sum_{t=1}^{T} \rho_\tau(X_{j,t} - \widehat{X}^\tau_{j})}, \tag{4.2}$$

where $\widehat{X}^\tau_j$ is the estimated unconditional τ-quantile. It should be noted that for the in-sample fit it has to hold that $R^1 \in [0,1]$. However, the out-of-sample fit for a particular unsuitable model can be worse than a constant unconditional quantile fit. In this case the R^1 can even be negative.

Lastly, we introduce the ratio of quantile exceedances ($RQEX$) as a measure of calibration. A well-calibrated model should have a ratio close to τ:

$$RQEX = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}_{X_{j,t} < \widehat{X}^\tau_j}. \tag{4.3}$$

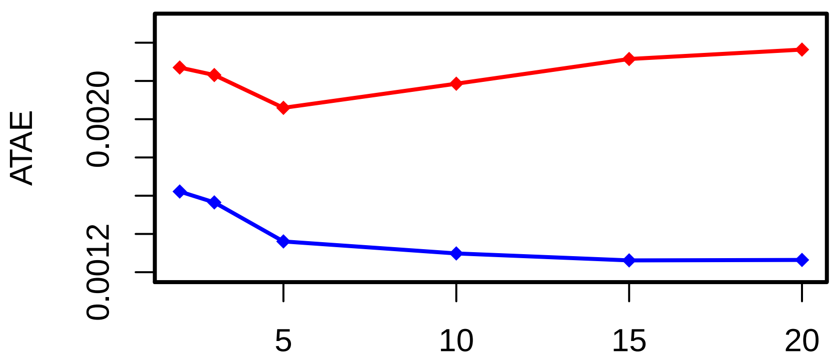

Figure 3: $ATAE$ contingent on the number of hidden nodes. In-sample fit (blue line) and out-of-sample fit (red line).

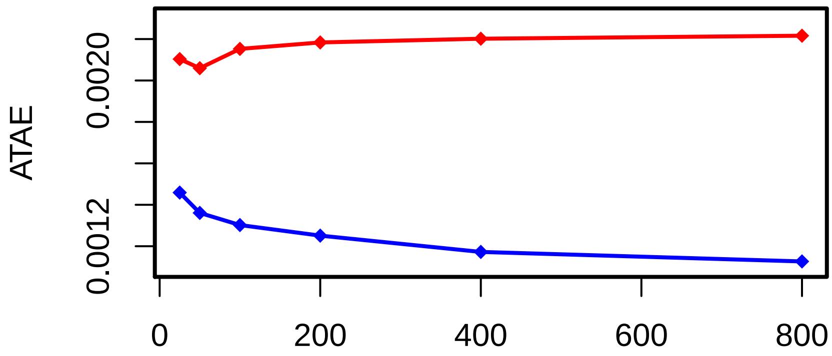

Figure 4: $ATAE$ contingent on the number of epochs. In-sample fit (blue line) and out-of-sample fit (red line).

Figures 3 and 4 visualize the problems of overfitting. A large number of hidden nodes and a large number of epochs can effectively reduce the training error (blue line). However, the test error (red line) decreases only to a certain point. After this point the test error starts to increase again.

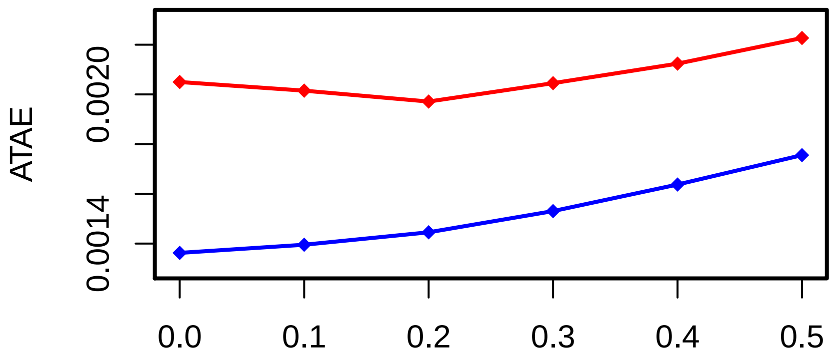

Figure 5: $ATAE$ contingent on the dropout rate p. In-sample fit (blue line) and out-of-sample fit (red line).

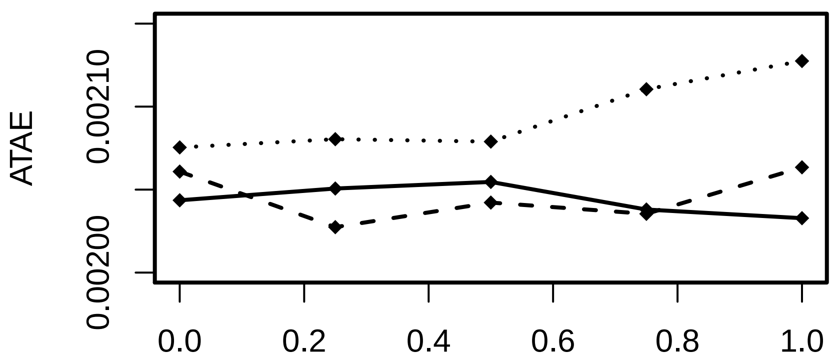

Figure 6: $ATAE$ contingent on elastic net parameter α. $\lambda = 0.0001$ (solid line), $\lambda = 0.001$ (dashed line) and $\lambda = 0.01$ (pointed line).

We considered two different regularization methods, dropout and elastic net. The results are shown in Figures 5 and 6. A small dropout rate (10-20%) can reduce the test error compared to the baseline setting of zero dropout. Elastic net penalization can also improve the fit, given that λ is not chosen to be too large. Both methods have, however, a negative impact on the in-sample fit, as they introduce a bias to the model.

For model selection we consider 10 different model specifications. The results can be found in Table 2 [1].

Model	$ATAE$	R^1	$RQEX$
ReLu, $H = 5$	0.002026	0.4668	0.0650
ReLu, $H = 5$, $\alpha = 0$, $\lambda = 0.001$	0.002026	0.4675	0.0661
ReLu, $H = 5$, $\alpha = 0.25$, $\lambda = 0.001$	0.002023	0.4682	0.0657
ReLu, $H = 5$, $p = 0.1$	0.001973	0.4728	0.0595
ReLu, $H = 5$, $\alpha = 0.25$, $\lambda = 0.001$, $p = 0.1$	0.001973	0.4740	0.0584
ReLu, $H = (5, 2)$, $p = 0.1$	0.002193	0.4554	0.0770
ReLu, $H = (3, 3)$, $p = 0.1$	0.002227	0.4412	0.0677
ReLu, $H = 10$, $p = 0.1$	0.002033	0.4788	0.0752
tanh, $H = 2$	0.002845	0.3030	0.1018
tanh, $H = 5$	0.002643	0.3691	0.1036

Table 2: Out-of-sample performance for different model specifications. We consider ReLu and tanh activation functions, H refers to the number and structure of hidden nodes, α and λ are the elastic net parameters and p is the input layer dropout rate.

The results in Table 2 suggest that complex models are dominated by less complex models. As a second observation, the ReLu (rectifier linear unit) activation function is superior to the tanh activation function. Both dropout and elastic net have a positive impact on the model performance.

The best model of the candidates is a neural network with 5 hidden nodes in a single hidden layer with a ReLu activation function. The model has also a dropout ratio of $p = 0.1$ and elastic net parameters $\alpha = 0.25$ and $\lambda = 0.001$. It ranks first in $ATAE$ (one of only two models that fall below 0.002) and second in R^1. Also the model's $RQEX$ (0.0584) is the closest to the quantile level of 0.05. We will use this model in the following estimation steps.

[1] We use the ADADELTA optimization algorithm with parameters $\rho = 0.99$ and $\epsilon = 1e - 08$. The number of epochs for all models is 50.

4.3 Estimation Results

4.3.1 VaR and CoVaR

As explained in section 3, the analysis is carried out in four steps. In the first two steps VaR and CoVaR are estimated for each firm, using linear quantile regression and neural network quantile regression, respectively. To account for potential non-stationarity, we employ a sliding window estimation framework for both measures. The window size is chosen to be 250 observations (implying one year of daily stock returns). We chose the quantile level $\tau = 5\%$. The fitted values for all banks are visualized in Figure 7.

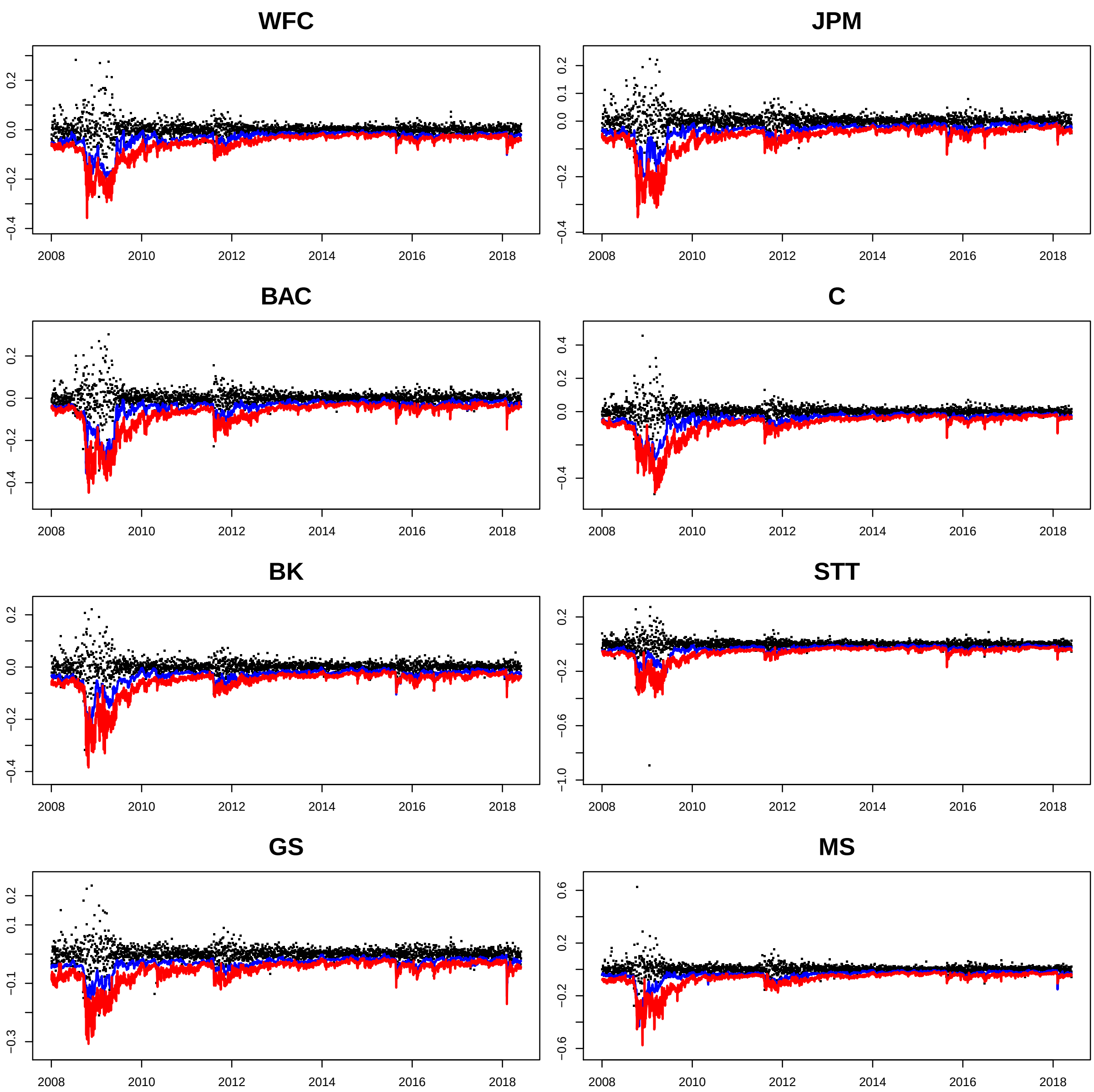

Figure 7: Plot of Returns (black dots), VaR (blue line) and CoVaR estimated by NNQR (red line).

The VaRs and CoVaRs of all banks follow a similar pattern. In the course of the financial crisis and the bankruptcy of Lehman Brothers and Bear Stearns, both measures explode, indicating an increase in systemic risk during this period. After a short stabilization period, the CoVaRs rise again in the second half of 2011. What follows is a relatively stable period with a few non-persistent spikes.

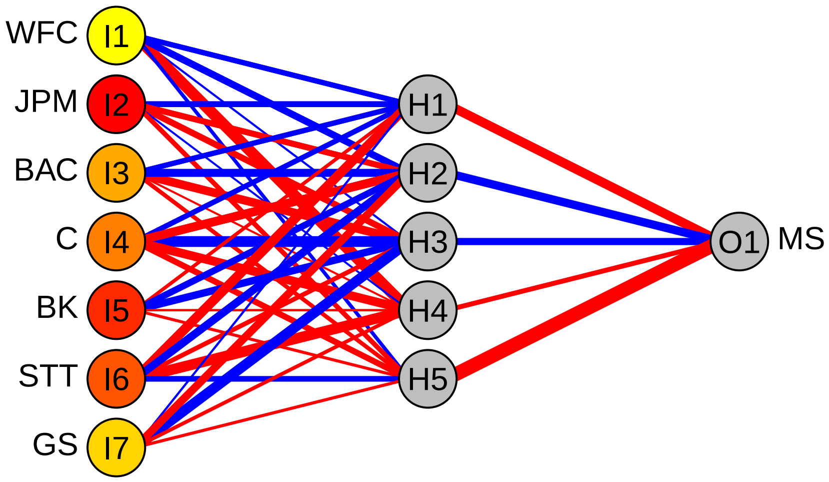

Figure 8: Fitted quantile regression neural network for Morgan Stanley on March 13, 2008. Red connections indicate negative weights, blue connections indicate positive weights. The color of the input nodes visualizes the variable importance rank calculated as the marginal effect of the respective firm on Morgan Stanley (yellow implies low importance, red implies high importance).

4.3.2 Risk Spillover Network

Based on the weights estimated in the NNQR procedure and the estimated VaRs and CoVaRs, we calculated the spillover of each pair of banks for each point in time of the estimation horizon. The result is a time-varying weighted adjusted adjacency matrix (as defined in equation 3.22). Figure 9 visualizes a simple time average of these matrices.

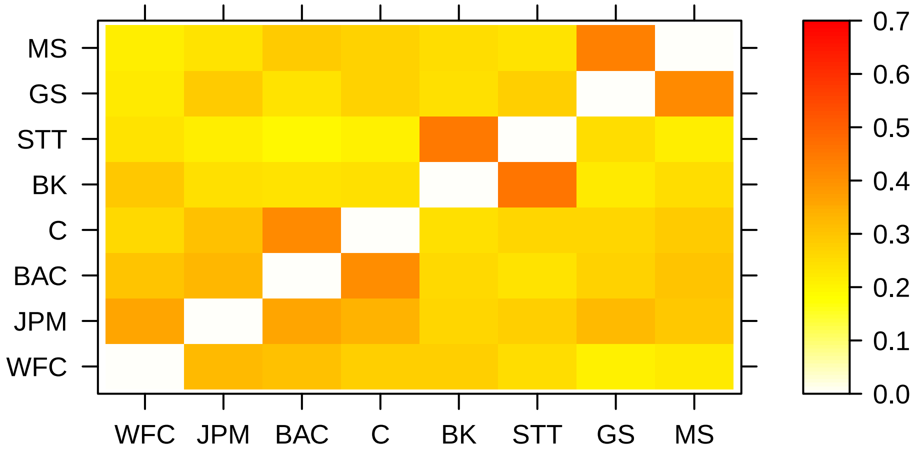

Figure 9: Level plot of the risk spillover effects, averaged over the whole estimation period.

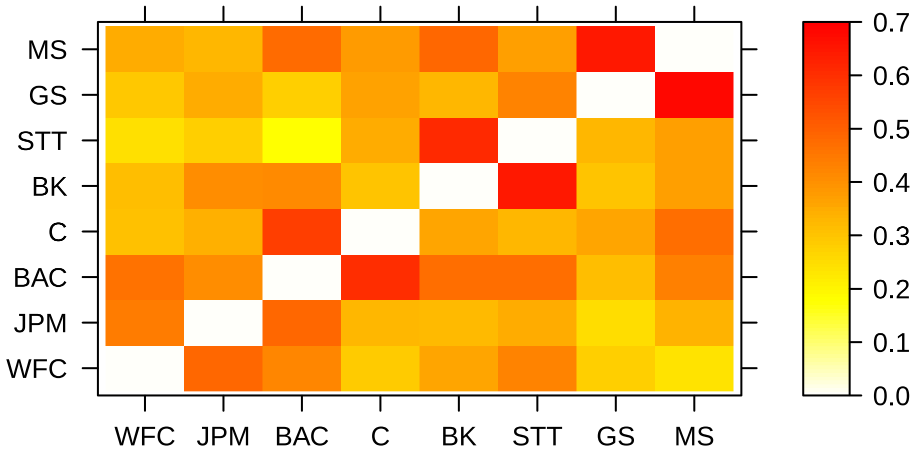

Figure 10: Level plot of the risk spillover effects, averaged over post-Lehman period (September 15, 2008 - December 14, 2008).

When restricting the included observations to the period up to three months after the Lehman bankruptcy, the time average looks differently. Figure 10 shows that the spillover effects were significantly larger during this period of financial distress. This result is in line with economic theory, as financial institutions were in fact very interconnected due to large derivative positions causing mutual counterparty risk.

Another important observation that can be made from these two plots is that the spillover effects have a tendency to be symmetric. If one bank has a large impact on another bank, the converse is also likely. This symmetry pattern becomes even more clear when looking at the network representations of the spillover effects in Figures 11 and 12. The largest edges (30%) of the network, as visualized in Figure 12, occur mostly in pairs. The symmetry is not caused by the model setup, which is asymmetric in its nature, but is rather implied by the data.

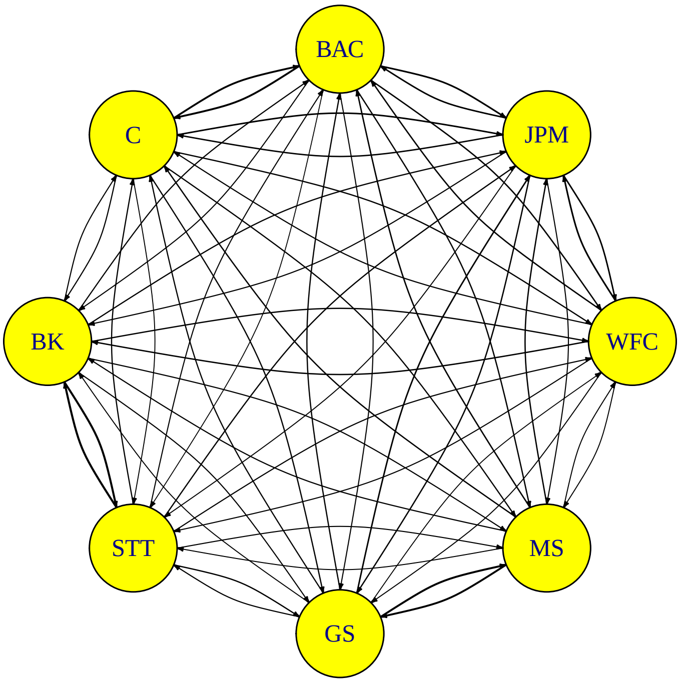

Figure 11: Systemic Risk network of spillover effects, averaged over the whole estimation period.

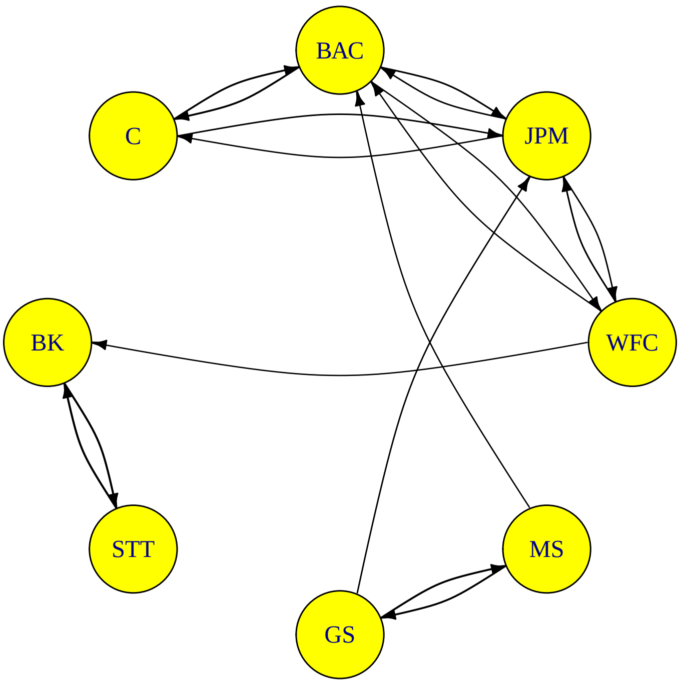

Figure 12: Systemic Risk network of spillover effects, averaged over the whole estimation period. Only the 30% largest edges are displayed.

4.3.3 Network Risk Measures

Finally, we estimated the systemic network measures using the results of the previous steps. The *Systemic Network Risk Index* is a measure for the overall systemic risk in the financial system. The time series plot can be found in Figure 13. From 2008 until the start of 2018 there have been one large and two smaller spikes. The first and largest spike represents the height of the financial crises. Two smaller ones follow in 2011/2012 and 2015/2016, each.

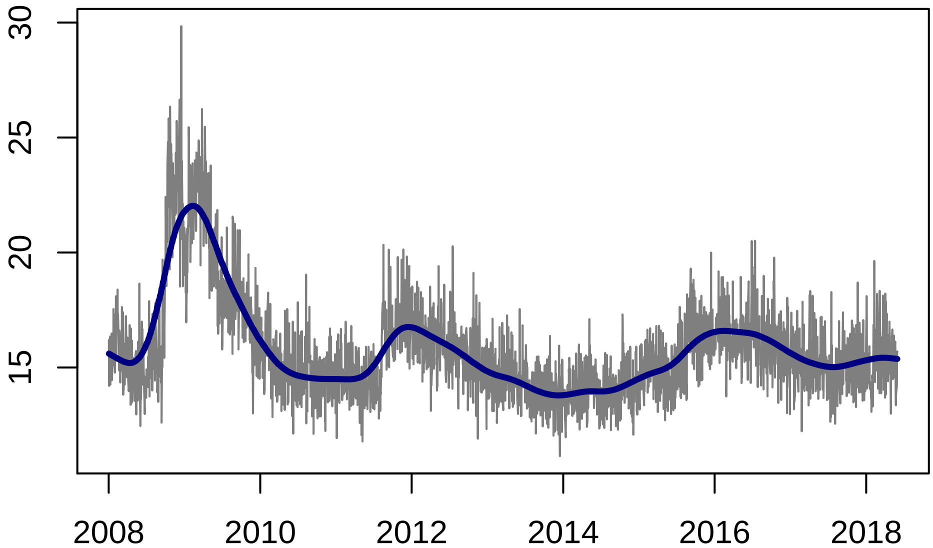

Figure 13: Systemic Network Risk Index (grey line) and cubic spline interpolation with sparsity parameter equal to 0.8 (blue line).

Rank	Bank	SFI
1	STT	2.433
2	BK	2.362
3	BAC	2.225
4	MS	2.233
5	JPM	2.134
6	C	2.125
7	GS	2.069
8	WCF	2.019

Table 3: Firms ranked according to the systemic fragility index (averaged over post-Lehman period).

Rank	Bank	SHI
1	BAC	2.730
2	GS	2.431
3	MS	2.372
4	BK	2.342
5	C	2.289
6	JPM	2.221
7	WCF	2.211
8	STT	2.041

Table 4: Firms ranked according to the systemic hazard index. (averaged over post-Lehman period).

While the $SNRI$ does not differentiate between different banks, we then identified the most systemically relevant firms during the financial crises. Hereby we considered the most vulnerable banks identified by the SFI in Table 3 as well as the most dangerous banks identified by the SHI in Table 4. The results in the tables represent averages over the three month period after the Lehman bankruptcy.

The most fragile banks according to our methodology are the *State Street Corporation*, the *The Bank of New York Mellon Corporation* and the *Bank of America Corporation.* The firms which impose the largest systemic risk to the financial system are again the *Bank of America Corporation*, *Goldman Sachs* and *Morgan Stanley.* While *Goldman Sachs* ranks very high in the SHI, it is nearly at the bottom of the SFI, indicating that their exposure to the financial system is weaker than the other way around. The opposite is the case for the *State Street Corporation* which has the largest SFI and the lowest SHI of all financial institutions that we have considered.

5 Conclusion

Even if the global economy seems to have recovered from the financial crisis, systemic risk is still a relevant topic. Whereas there are no immediate systemic threats to the financial system today, latent risks are still present.

This master thesis proposes a novel approach to estimate the conditional Value at Risk (CoVaR) of financial institutions based on neural network quantile regression. We estimate a network of systemic risk spillover effects and propose three network-based risk measures, the *Systemic Fragility Index* to rank the firms with the largest exposure to the financial system, the *Systemic Hazard Index* which ranks the firms according to the risks they impose to the financial system and the *Systemic Network Risk Index* which is a measure for the overall systemic risk.

The methodology is applied to the global systemically important banks from the United States in the period from 2007 until 2018. The results are in line with previous findings in the literature. We observe the Systemic Network Risk Index increasing sharply during the financial crisis after which it stabilizes.

This master thesis is an important contribution to the vast literature about systemic risk. Neural networks have been utilized almost exclusively as a device for prediction. An accomplishment of this thesis is to find a way to interpret the underlying neural network by estimating risk spillover effects out of the fitted neural networks.

We leave it open for future research to investigate possible benefits of connecting the estimation of CoVaR in the cross-sectional and the time series dimension. Our current methodology treats the single estimation problems separately from each other. Recent advances in transfer learning and multitask learning suggest that this is promising research path to increase efficiency.

References

[1] Adrian, T. and Brunnermeier, M.K., (2016). CoVaR. *American Economic Review* vol. 106, no. 7, (pp. 1705-41).

[2] Ba, J. and Kingma, P., (2015). ADAM: A Method for Stochastic Optimization. Published as a conference paper at ICLR 2015.

[3] Cannon, A.J. (2011). Quantile regression neural networks: Implementation in R and application to precipitation downscaling. *Computers & Geosciences* 37, 1277-1284.

[4] Chao, S.-K., Härdle, W.K., Wang, W., 2015. Quantile regression in risk calibration. Handb. Financ. Econom. Stat. 1467-1489.

[5] Chrislb, 2005. Figure retrieved from https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png on June 15, 2018.

[6] Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Math. Control Signals Systems* (1989) 2:303-314.

[7] Diebold, F.X. and Yilmaz, K. (2014). On the network topology of variance decompositions: Measuring the connectedness of financial firms. *Journal of Econometrics* 182, 119-134.

[8] Engle, R.F. and Manganelli, S. (2004). CAViaR, *Journal of Business & Economic Statistics*, 22:4, 367-381, DOI:10.1198/073500104000000370.

[9] Glorot, X., Bordes, A. and Bengio., Y. Deep sparse rectifier neural networks (2011). In Proc. 14th International Conference on Artificial Intelligence and Statistics, 315-323.

[10] Härdle, W.K., Wang, W. and Yu, L. (2016). TENET: Tail-Event driven NETwork risk. *Journal of Econometrics*, Volume 192, Issue 2, Pages 499-513.

[11] Hardt, M., Recht, B. and Singer, Y. Train faster, generalize better: Stability of stochastic gradient descent. *arXiv*, 1509.01240, 2015.

[12] Hastie, T., Tibshirani, A. and Friedman, J. (2009). The Elements of Statistical Learning (2. Edition). Stanford, Springer.

[13] Hastie, T. and Zou, H. (2005). Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67, Part 2, pp. 301-320.

[14] Hautsch, N., Schaumburg, J. and Schienle, M. (2014). Financial Network Systemic Risk Contributions. *Review of Finance*, Volume 19, Issue 2, 1 March 2015, Pages 685-738.

[15] Hinton, G.E., Rumelhart, D.E. and Williams, R.J. (1986). Learning representations by back-propagating errors. *Nature* (1986) Band 323, S. 533-536.

[16] Hinton, G.E., Krizhevsky, A., Salakhutdinov, R., Srivastava, N. and Sutskever, I. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15, 1929-1958.

[17] Hinton, G.E., Krizhevsky, A., Salakhutdinov, R., Srivastava, N. and Sutskever, I. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:* 1207.0580.

[18] Jiang, C., Liu, X., Xu, Q., and Yu, K. (2016). Quantile autoregression neural network model with applications to evaluating value at risk. *Applied Soft Computing* 49, 1-12.

[19] Koenker, R.W. and Bassett, G.W. (1978). Regression Quantiles, *Econometrica*, 46, 33-50.

[20] Koenker, R.W. and Bassett, G.W. (1982). Robust Tests for Heteroscedasticity Based on Regression Quantiles. *Econometrica* 50, 43-62.

[21] Koenker, R.W. and Machado J.A.F. (1999). Goodness of fit and related inference processes for quantile regression, *J. Amer. Statist. Assoc.* 94(448), pp. 1296-1310.

[22] Kuan, C.-M. and White, H. (1994). Artificial neural networks: an econometric perspective. *Econometric Reviews*, 13:1, 1-91, DOI: 10.1080/07474939408800273.

[23] Taylor, J.W. (2000). A Quantile Regression Neural Network Approach to Estimating the Conditional Density of Multiperiod Returns. *Journal of Forecasting*, Vol. 19, pp. 299-311.

[24] Werbos, P. (1974). Beyond regression: new tools for prediction and analysis in the behavioral sciences. PhD thesis, Harvard University, Cambridge, Nov. 1974.

[25] Zeiler, M. D. (2012). ADADELTA: An adaptive learning rate method. *arXiv*: 1212.5701., 2012.

Declaration of Authorship

I hereby confirm that I have authored this Master's thesis independently and without use of others than the indicated sources. All passages which are literally or in general matter taken out of publications or other sources are marked as such.

Berlin, July 9, 2018

Georg Keilbar