IWH
Halle Institute for Economic Research
Member of the Leibniz Association

Discussion Papers

No. 6
March 2024

Forecasting Economic Activity Using a Neural Network in Uncertain Times: Monte Carlo Evidence and Application to the German GDP

Oliver Holtemöller, Boris Kozyrev

Authors

Oliver Holtemöller
Halle Institute for Economic Research (IWH) –
Member of the Leibniz Association,
Department of Macroeconomics, and
Martin Luther University Halle-Wittenberg
E-mail: oliver.holtemoeller@iwh-halle.de
Tel +49 345 7753 800

Boris Kozyrev
Corresponding author
Halle Institute for Economic Research (IWH) –
Member of the Leibniz Association,
Department of Macroeconomics
E-mail: boris.kozyrev@iwh-halle.de
Tel +49 345 7753 865

The responsibility for discussion papers lies solely with the individual authors. The views expressed herein do not necessarily represent those of IWH. The papers represent preliminary work and are circulated to encourage discussion with the authors. Citation of the discussion papers should account for their provisional character; a revised version may be available directly from the authors.

Comments and suggestions on the methods and results presented are welcome.

IWH Discussion Papers are indexed in RePEc-EconPapers and in ECONIS.

Editor

Halle Institute for Economic Research (IWH) –
Member of the Leibniz Association

Address: Kleine Maerkerstrasse 8
D-06108 Halle (Saale), Germany
Postal Address: P.O. Box 11 03 61
D-06017 Halle (Saale), Germany

Tel +49 345 7753 60
Fax +49 345 7753 820

www.iwh-halle.de

ISSN 2194-2188

Forecasting Economic Activity Using a Neural Network in Uncertain Times: Monte Carlo Evidence and Application to the German GDP

Abstract

In this study, we analyzed the forecasting and nowcasting performance of a generalized regression neural network (GRNN). We provide evidence from Monte Carlo simulations for the relative forecast performance of GRNN depending on the data-generating process. We show that GRNN outperforms an autoregressive benchmark model in many practically relevant cases. Then, we applied GRNN to forecast quarterly German GDP growth by extending univariate GRNN to multivariate and mixed-frequency settings. We could distinguish between “normal” times and situations where the time-series behavior is very different from “normal” times such as during the COVID-19 recession and recovery. GRNN was superior in terms of root mean forecast errors compared to an autoregressive model and to more sophisticated approaches such as dynamic factor models if applied appropriately.

Keywords: forecasting, neural network, nowcasting, time series models

JEL classification: C22, C45, C53

1 Introduction

A timely assessment of the current economic development and the short-term economic outlook is essential for many decisions in companies, governments, public institutions and private households. Accordingly, many approaches for nowcasting and forecasting economic activity exist. An important indicator of economic activity is gross domestic product (GDP).

However, standard forecasting models may yield unreliable estimates during recessions or crises. The most recent example is the COVID-19 crisis. The pandemic was an unprecedented shock never observed before in modern history (Barbaglia et al., 2022). Standard models have failed to estimate its impact on the world economy in real time. We propose a nowcasting method based on a simple neural network (NN) called a Generalized Regression Neural Network (GRNN). GRNN has a significant advantage over more sophisticated NN specifications: it is relatively intuitive. Decision makers who use GRNN results can understand how forecasts are derived.

GRNN is a neural-network implementation of the non-parametric Nadaraya-Watson Gaussian Kernel Regression estimator (Nadaraya, 1964; Watson, 1964). Therefore, it is assumed that advantages from both NN and non-parametric approaches will be shared. For instance, no assumptions of a specific data distribution or a specific functional relationship between predictors and dependent variables are required. Moreover, non-parametric methods are less sensitive to outliers than parametric approaches. Martínez et al. (2022) proposed a new method of feature selection to produce forecasts using GRNN, confirming that the obtained predictions can compete with different – often less tractable – NN architectures.

In this study, we applied the GRNN framework to economic settings. We systematically investigated the GRNN performance depending on the data-generating process using Monte Carlo simulations. We argue that a GRNN-based approach has a higher forecasting power than standard parametric frameworks. We conducted Monte Carlo simulations for various data-generating processes (DGPs), in which predictions obtained from the GRNN model are compared to an autoregressive process of order 1, AR(1), as a benchmark. In addition, we compared GRNN forecasting performance to automatically fitted autoregressive moving-average (ARMA) models. First, we simulated stationary linear univariate time series, such as AR(1), AR(4), ARMA(3,3). We demonstrate that GRNN surpasses the benchmark in forecasting performance in some practically relevant scenarios. Moreover, upon extending the simulation procedure to nonlinear settings, we observed that GRNN may be superior to the selected benchmark and estimated ARMA models.

Finally, we performed a one-step-ahead pseudo-out-of-sample forecast to predict German GDP and show how to integrate indicators, such as industrial production, into the approach. Our empirical results suggest that GRNN can be helpful for short-term forecasting. If applied appropriately, our method can be particularly beneficial during uncertain times such as the COVID-19 crisis.

The structure of the paper is as follows: Section 2 explains the GRNN approach. In section 3, we present the Monte Carlo simulations. Section 4 contains the application of the approach to forecasting German GDP growth. Conclusions are provided in section 5.

2 Generalized Regression Neural Network (GRNN)

2.1 General Approach

A generalized regression neural network (GRNN) is a variation of a radial basis NN, first proposed by Specht et al. (1991). GRNN can be graphically represented using nodes and layers, see, for example, Sumiyati and Warsito (2020). It is a non-parametric model that can be applied to classification or (time-series) regressions. This model assumes that a prediction can be obtained as a weighted average of all previous values based on their proximity to its last behavior, which allows a very intuitive interpretation. GRNN is a version of the Nadaraya-Watson Gaussian Kernel Regression estimator (see Appendix A) defined in terms of a NN (Ahmed et al., 2010).

GRNN has several advantages over other algorithms. First, GRNN is non-parametric. No assumptions regarding the functional form are required. Unlike the ARIMA approach, a global model in which a chosen relationship is extrapolated to all observations, GRNN is a local estimator (Gheyas and Smith, 2009). Moreover, only two parameters must be defined: the number of lags (k) and a smoothing parameter (σ).

More formally, suppose both a training set consisting of n training patterns (vectors) $\{x_1, x_2, \ldots, x_n\}$ and corresponding targets $\{y_1, y_2, \ldots, y_n\}$ are given. The output for an input pattern x is estimated based on the closeness of x to the training patterns x_i. A corresponding weight w_i could be expressed as follows:

$$w_i = \frac{exp(-\frac{||x-x_i||^2}{2\sigma^2})}{\sum_{l=1}^{n} exp(-\frac{||x-x_l||^2}{2\sigma^2})}, \tag{1}$$

where $||\cdot||$ is the Euclidian distance, and σ is a smoothing parameter. The forecast is a weighted sum of training target outputs:

$$\hat{y} = \sum_{i=1}^{n} w_i y_i, \tag{2}$$

where y_i is the target output for training data x_i.

This approach can also be applied to time series data. In a univariate case, previously observed values may be used to determine future behavior of a given time series. First, a training set of k-tuples of sequential historical values is created. The corresponding training target is the following observation. Then, the last k observations are used as an input pattern. To illustrate this idea, Figure 1 depicts the dynamics of the German GDP's quarter-over-quarter (q-o-q) growth rate between the first quarter of 1991 and the third quarter of 2003. In this example, we have 50 observations and $k = 4$, indicating that we have a training set containing 46 k-tuples:

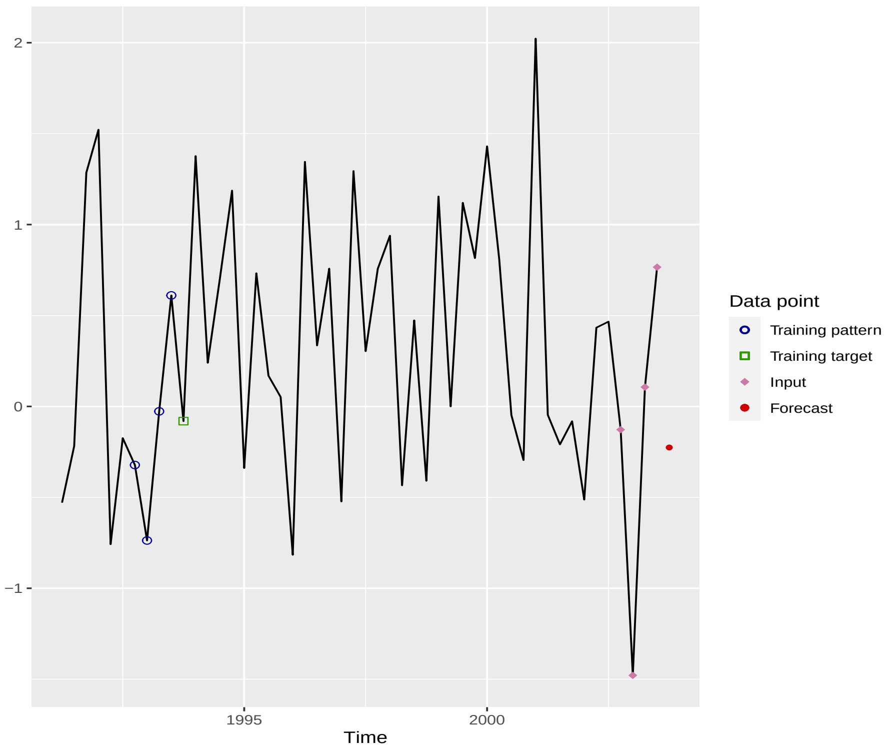

Figure 1: GRNN time series framework

$S = \{(x_1, x_2, x_3, x_4), (x_2, x_3, x_4, x_5), ..., (x_{46}, x_{47}, x_{48}, x_{49})\}$. For example, let us take the first entity of S, namely $S_1 = (x_1, x_2, x_3, x_4)$. In this case, its corresponding target would be the next observed value x_5. The same holds for every other element in S_k. An input pattern consits of the last k observations, i.e., $x = (x_{47}, x_{48}, x_{49}, x_{50})$. To produce a forecast, we calculated the Euclidian distance for each element in S_k and weights were assigned to every training target value according to (1). Moreover, a weighted average was taken as in (2). Figure 1 shows the training pattern, its target value, the input target, and a one-step-ahead forecast. This training pattern was chosen because it was the closest to the last observed values (input pattern) according to the Euclidian distance. This implies that the highlighted target value has the largest weight, w_i.

Several methods for determining k and σ have been proposed, including K-folded cross-validation (Yan, 2012). We relied on Martínez et al. (2022), who suggested a fast algorithm to produce highly accurate forecasts. The number of lags could be determined using a heuristic approach:

1. If a time series is seasonal, then $k = s$, where s is the length of the seasonal period.

2. If a time series is not seasonal, $k = p$, where p is the number of significant lags of the partial autocorrelation function (PACF).

3. If both conditions are not met, $k = 5$.

σ is estimated by minimizing the forecasting error on a validation set, which is formed by the last h observations of a given time series, where h is the forecasting horizon.

To predict future values of non-stationary or trend-stationary time series, we modified the basic approach. GRNN time series forecasting is based on detecting patterns in the past. This means that this model cannot predict an observation that is out of range of previous observations. The following additive data transformation is proposed to capture trend-stationarity:

1. Both a training pattern vector and the corresponding target value(s) are modified by subtracting the mean of the former. Take the previous example, illustrated in Figure 1. Each observation in $S_1 = (x_1, x_2, x_3, x_4)$ is demeaned by $\bar{x}_1 = \frac{\sum_{i=1}^{4} x_i}{4}$. Thus, we can obtain $\tilde{S}_1 = (x_1 - \bar{x}_1, x_2 - \bar{x}_1, x_3 - \bar{x}_1, x_4 - \bar{x}_1)$. The initial training target value x_5 is similarly modified as $\tilde{x}_5 = x_5 - \bar{x}_1$. The input pattern is also demeaned to compute the Euclidian distance, i.e., $\tilde{x} = (x_{47} - \bar{x}_1, x_{48} - \bar{x}_1, x_{49} - \bar{x}_1, x_{50} - \bar{x}_1)$. This assures that the distance is unaffected by the transformation. Then, a weight w_5 is assigned to $\tilde{x}_5$. This procedure applies to each entity in S_k, each corresponding target value, and the input pattern.

2. A prediction is obtained using modified data. In our case, the prediction is a weighted sum of transformed target values, namely $\hat{\tilde{x}}_{t+1} = \sum_{t=5}^{50} w_t \tilde{x}_t$

3. The mean of the input vector is added to obtain the actual forecast. In other words, the final prediction is $\hat{x}_{t+1} = \sum_{t=5}^{50} w_t \tilde{x}_t + \bar{x}$, where $\bar{x} = \frac{\sum_{i=47}^{50} x_i}{4}$ is the mean of input vector x.

If a time series is characterized by an exponential trend, instead of demeaning, we divide by the mean of each k-tuple in S_k (multiplicative transformation). The transformed weighted average is multiplied by the mean value of input vector x to retrieve a final forecast.

Finally, we use an iterative approach throughout the entire study: If one wants to forecast h-step-ahead, where $h \geq 2$, first, a one-step-ahead forecast must be carried out. Then, the forecast is treated as an observation to produce another one-step-ahead forecast. In other words, to produce a h-step-ahead forecast, a one-step-ahead forecast is iteratively repeated h times.

2.2 Extension to the Multivariate Case and Mixed Frequencies

Statistical offices usually report GDP at a quarterly frequency. Many economic indicators containing information for estimating current and future GDP are available at higher frequencies. Therefore, several approaches have been developed that can combine various frequencies, such as Dynamic Factor Models (DFM) (Bańbura and Rünstler, 2011), Mixed-Data Sampling (MIDAS), and Mixed-frequency Vector Autoregressive Models (MF-VAR) (Kuzin et al., 2011). We need to adjust the initial approach to exploit the possibility of using GRNN to forecast or nowcast GDP.

Although it seems possible to naturally extend the GRNN approach to multivariate settings, as proposed in Martínez et al. (2022), we suggest an alternative method for adjusting GRNN for forecasting. First, we aggregated monthly data to quarterly frequency to match the GDP reporting schedule. Each aggregated time series is then divided by the corresponding GDP value, creating a new dataset relative to GDP. In the next step, GRNN is applied to separately predict the ratio of each time series relative to GDP, considering that these ratios may be non-stationary by using data transformations (multiplicative and additive). Finally, a set of individual forecasts is obtained by updating each predicted ratio with observed and forecast values for a given time series, and a median value is calculated to produce the final estimation.

3 Monte Carlo Simulations

We studied the forecast precision of GRNN time-series predictions depending on various DGPs using Monte Carlo simulations. We assumed that each simulated series y is a monthly time series, and we aimed to predict the next 12 months. AR(1) was selected as a benchmark, i.e., we compare GRNN forecast precision to its AR(1) counterpart when the true DGP is unknown.

3.1 Forecasting Algorithm

We considered N samples of n observations of a target series $y_{i,t}$, $i = 1 \dots N$, $t = 1 \dots n$, generated from a specific DGP. Each sample was divided into two separate observation groups, such that $n = I + P$. $I = n - 12$ in-sample values were exploited to estimate each model, whereas $P = 12$ pseudo-out-of-sample observations are used to evaluate forecast precision. The models' forecasting performance is evaluated by using the mean absolute forecast error ($MAFE$) and the root mean squared forecast error ($RMSFE$). $MAFE$ for forecast horizon h can be expressed as follows:

$$MAFE_h = \frac{\sum_{i=1}^{N} |y_{i,t+h} - \hat{y}_{i,t+h|t}|}{N}, \tag{3}$$

where h is the forecast horizon, $\hat{y}_{i,t+h|t}$ is a prediction of $y_{i,t+h}$ based on available information at time t, and $y_{i,t+h}$ is the actual value at time $t+h$. $RMSFE$ for forecast horizon h is defined as:

$$RMSFE_h = \sqrt{\frac{\sum_{i=1}^{N} (y_{i,t+h} - \hat{y}_{i,t+h|t})^2}{N}}. \tag{4}$$

Additionally, we define relative $MAFE$ and relative $RMSFE$, denoted $rMAFE$ and $rRMSFE$, which aid the assessment of the GRNN forecasting performance relative to the AR(1) benchmark:

$$\begin{aligned} rMAFE_h &= \frac{\sum_{i=1}^{N} |y_{i,t+h} - \hat{y}^{GRNN}_{i,t+h|t}|}{\sum_{i=1}^{N} |y_{i,t+h} - \hat{y}^{AR(1)}_{i,t+h|t}|} = \frac{MAFE^{GRNN}_h}{MAFE^{AR(1)}_h} \\ rRMSFE_h &= \frac{\sqrt{\sum_{i=1}^{N} (y_{i,t+h} - \hat{y}^{GRNN}_{i,t+h|t})^2}}{\sqrt{\sum_{i=1}^{N} (y_{i,t+h} - \hat{y}^{AR(1)}_{i,t+h|t})^2}} = \frac{RMSFE^{GRNN}_h}{RMSFE^{AR(1)}_h}, \end{aligned} \tag{5}$$

We chose $N = 10000$ and considered sample lengths $n \in \{112, 212, 412, 1012\}$. GRNN should perform better after ample observations are made. However, macroeconomic data are typically limited. Therefore we analyzed small, medium and large sample sizes.

As DGPs we considered $AR(1)$, $AR(4)$, $ARMA(3,3)$, $VAR(2)$, and $TAR(2)$. We start from the $AR(1)$, our benchmark. We expected the AR(1) to outperform GRNN if AR(1) is the true DGP. To analyze lag misspecification, we added $AR(4)$ as another GDP to our Monte Carlo exercise. Additionally, $ARMA(3,3)$ was included to study the importance of adding a moving-average component. A bivariate Vector Autoregressive Process (VAR) of order 2, denoted $VAR(2)$, was used to observe how missing variable(s) influence the relative forecasting performance. Finally, using a threshold autoregressive process (TAR), we introduced nonlinearity. We will explain the choice of parameters in the corresponding subsections.

Overall, we used the following algorithm:

1. Simulate $N = 10000$ samples of $n \in \{62, 112, 212, 412, 1012\}$ observations using different DGPs $\in \{AR(1), AR(4), ARMA(3,3), VAR(2), TAR(2)\}$.

2. Use $I = n - 12$ observations to estimate pre-selected models, namely AR(1) and GRNN with no transformation;

3. The last $P = 12$ observations are used to conduct the pseudo-out-of-sample forecast.

4. Compute $MAFE_h$ and $RMSFE_h \quad \forall h = 1, ..., 12$

5. Compute $rRMAFE_h$ and $rRMSFE_h \quad \forall h = 1, ..., 12$

3.2 Simulations Results

In this subsection, we describe our main findings from the Monte Carlo simulations. First, each DGP is discussed individually, including model specification and relative forecasting performance. We found out that $rRMAFE$ and $rRMSFE$ yielded almost identical results. Therefore, we report only the $rRMSFE$ values here. The forecasting performance within each specification was analyzed with respect to n and h. A potential trade-off between the computational time and forecasting precision is also discussed. As an additional comparison between GRNN and parametric approaches, we studied how fitting an $ARMA$ model based on information criteria performs.

AR(1)

The first DGP selected for the Monte Carlo simulation was AR(1). Even though this specification is straightforward, the obtained forecasts are typically precise for many different time series (Marcellino et al., 2006). More formally, AR(1) could be expressed as follows:

$$y_t = \alpha y_{t-1} + \epsilon_t, \tag{6}$$

where α is an AR(1) coefficient, and ϵ_t is white noise. The persistence of an AR(1) process depends on α. For example, if $|\alpha|$ is close to 0, the observed time series may resemble white noise. Meanwhile,

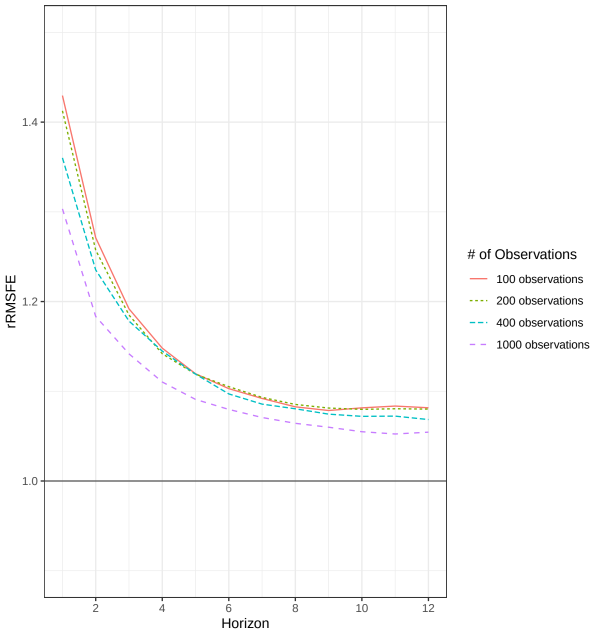

Figure 2: Case 1.1: AR(1), $\alpha = 0.9$, no transformation

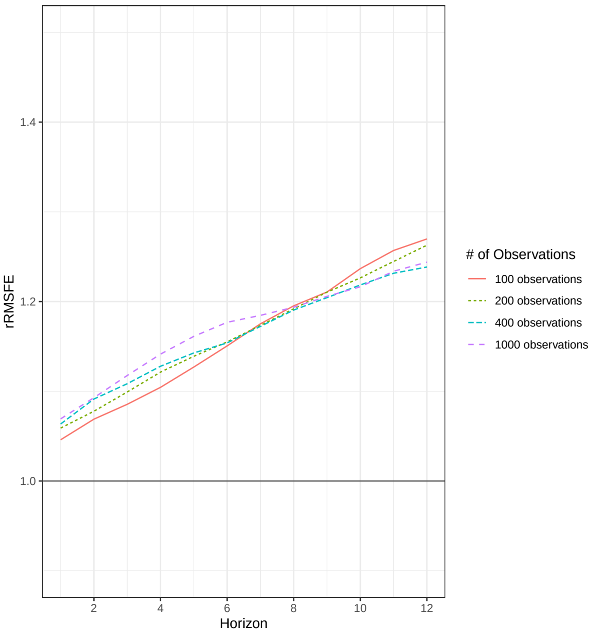

Figure 3: Case 1.1: AR(1), $\alpha = 0.9$, additive transformation

higher absolute values of α indicate a more significant contribution of a previous value of a time series compared to an error term. Therefore, to capture this phenomenon, we choose values of $\alpha \in \{0.9, 0.5, 0.2\}$. The error terms in every simulation were i.i.d. with $\epsilon_t \sim N(0, 1)$.

There are different methods to estimate AR(1) processes, such as ordinary least squares (OLS), maximum likelihood (ML) estimation, the conditional sum of squares (CSS), and Yule-Walker equations based on the method of moments. In this study, we estimated the time series using CSS. As expected, the simulation results show that for any given number of observations n, and for any forecasting horizon h, AR(1) is superior to GRNN in terms of forecast precision.[1]

Case 1.1: $\alpha = 0.9$. In Figure 2, $rRMSFE$ for $AR(1)$ with $\alpha = 0.9$ is presented. In general, a slow decay indicating an inverse proportionality between the forecast horizon and $rRMSFE$ was observed. For short-term forecasting, the number of observations plays a more significant role, i.e., the more data is available, the lower $rRMSFE$. However, this difference is disappearing for longer horizons. Recall that for any stationary $ARMA$ process, when the forecasting horizon tends to infinity, its point forecast tends to the mean. In other words, we expect $rRMSFE$ to tend to one, implying that GRNN will adapt its weights accordingly to produce a mean forecast. However, the asymptotic properties of the Nadaraya-Watson univariate time series estimator are difficult to postulate because they depend on different assumptions. These so-called mixing conditions are challenging to test in practice (Heiler, 1999). Moreover, macroeconomic time series are often too short to rely on asymptotic results. Therefore, it is important to study the forecast properties empirically. In Figure 3, $rRMSFE$

[1] Accordingly, we did not compare GRNN to forecasts based on the estimated true DGP for the remaining simulations.

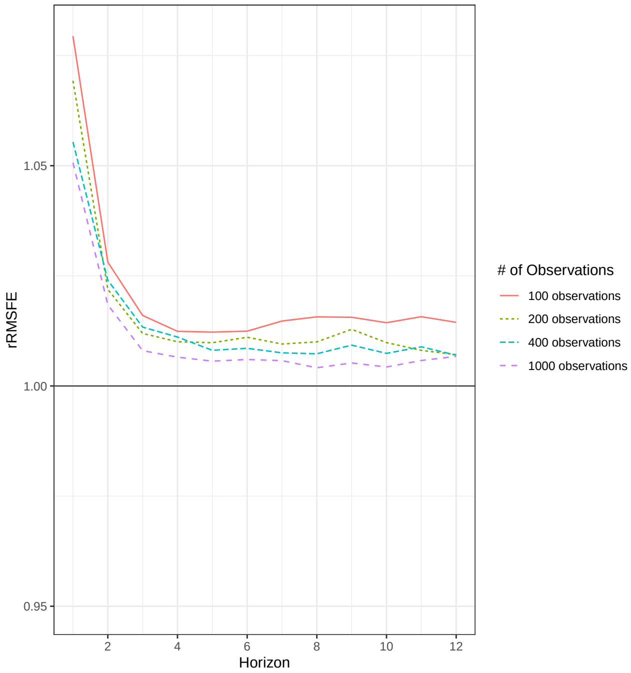

Figure 4: $rRMSFE$ for AR(1), $\alpha = 0.5$

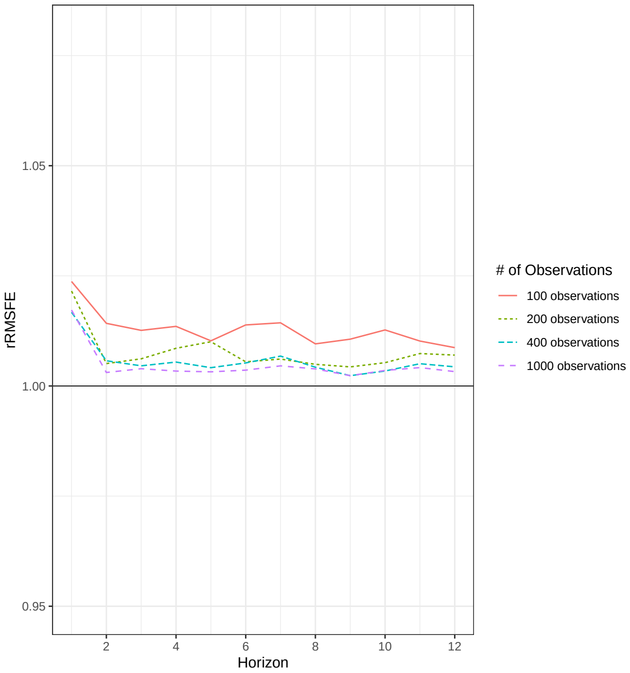

Figure 5: $rRMSFE$ for AR(1), $\alpha = 0.2$

for $AR(1)$ with $\alpha = 0.9$ with an additive data transformation is presented. In this case, additive data transformation is redundant because we have a stationary time series. However, due to high persistence, we found it beneficial to compare the forecasting performance of GRNN with additive transformation to GRNN with no transformation. For short-term horizons, GRNN with additive transformation outperforms its counterpart.[2] However, the $rRMSFE$ increases in h unlike in the case without transformation. Moreover, the number of observations n plays an important role. The less data we observed, the steeper the slope of $rRMSFE$. If the AR(1) process is highly persistent, the effect of applying data transformation on forecasting performance depends on h.

Case 1.2: $\alpha = 0.5$. In Figure 4, $rRMSFE$ for AR(1) with $\alpha = 0.5$ is shown. For this case, the values of $rRMSFE$ are lower compared to the previous case with a large α. Thus, on average, GRNN performs better if the autocorrelation value is farther from one. A similar slow decay of $rRMSFE$ was observed as in Figure 2.[3] Furthermore, $rRMSFE$ decreases if the number of observations n increases.

Case 1.3: $\alpha = 0.2$. This case is similar to case 1.2, see Figure 5. However, regardless of sample size n, GRNN performs almost as accurately as AR(1). When the persistence of an AR(1) process is relatively low, the forecast qualities of GRNN and AR(1) are comparable and the role of n becomes less critical.

[2] This was the only DGP used in our Monte Carlo simulations for which this held.

[3] Data transformation is redundant – $rRMSFE$ of GRNN with no data modification is lower for every h compared to the same model with transformation.

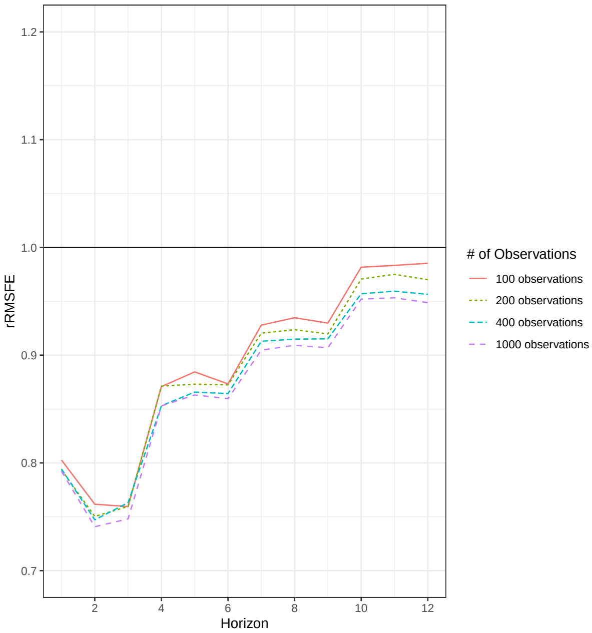

Figure 6: Case 2.1: $AR(4)$

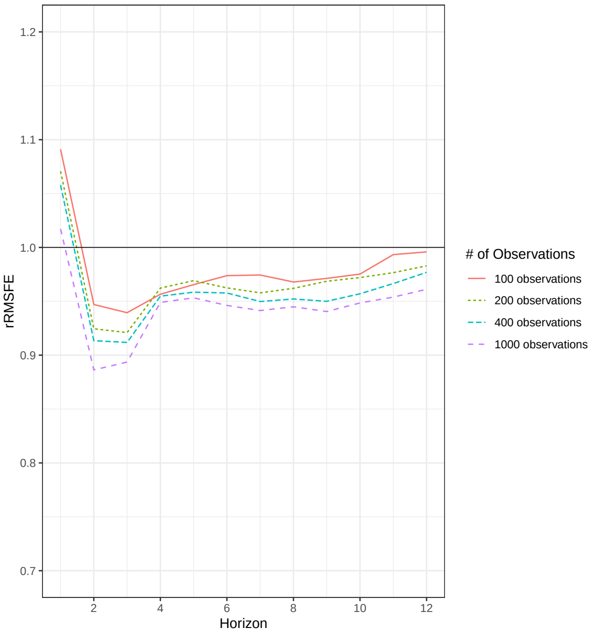

Figure 7: Case 2.2: $AR(4)$

AR(4)

One possible way to naturally extend the $AR(1)$ model is to increase the number of lags, thus shifting towards $AR(p)$, where $p > 1$. Using an $AR(p)$, we aimed to study the potential lag misspecification. As a baseline specification, we selected $AR(4)$. Because we focused on a stationary univariate time series, we had to define an AR(4) that satisfies the stationarity condition: the roots of the $AR(p)$ polynomial must be outside the unit circle. We selected two separate cases.

Case 2.1: AR(4) First, we considered the following $AR(4)$:

$$y_t = 0.2y_{t-1} - 0.2y_{t-2} + 0.5y_{t-3} - 0.6y_{t-4} + \epsilon_t, \tag{7}$$

where ϵ_t is white noise. This specification allowed us to examine a lag misspecification issue per se and consider the importance of higher-order lags while producing a forecast. As shown, the relative weights of y_{t-3} and y_{t-4} are larger than those of y_{t-1} and y_{t-2}. We chose these parameters on purpose. Supposedly, $AR(1)$ fails to detect this relationship, whereas a more flexible GRNN should adjust its weights accordingly. Imagine, however, a different scenario when the $AR(1)$ coefficient is dominant, forcing the entire DGP to mimic an AR(1). Under these settings, the Monte Carlo results may be indistinguishable from the previous case, see case 2.2.

In Figure 6, $rRMSFE$ for the baseline $AR(4)$, equation (7), is illustrated. As expected, GRNN outperformed $AR(1)$ for any given time horizon. The non-parametric approach can capture a more complicated DGP, regardless of the amount of observations. On average, $rRMSFE$ is less than 0.8 for one-step-ahead forecasting, indicating that GRNN predicts 20% more accurately. Similar to the $AR(1)$ case, $rRMSFE$ tends to one as the forecasting horizon h increases, implying that for a longterm prediction, both $AR(1)$ and GRNN have the same forecasting power because both predict the unconditional mean for longer forecast horizons. The speed of performance convergence depends on the sample size. When more observations are available, GRNN is more precise than $AR(1)$. However, even if the number of observations is low (50 or 100), GRNN performs better than a wrong parametric time series model.

Case 2.2: $AR(4)$ Now, we inspected the case in which the relative weights of y_{t-3} and y_{t-4} were significantly lower than those of y_{t-1} and y_{t-2}. To be more specific, let us define the following model:

$$y_t = 0.8y_{t-1} - 0.3y_{t-2} + 0.4y_{t-4} + \epsilon_t, \tag{8}$$

where ϵ_t is white noise.

In Figure 7, $rRMSFE$ based on the DGP defined in equation (8) is shown. The behavior of $rRMSFE$ is notably different from the previous case. For $h = 1$, $AR(1)$ outperforms GRNN for any number of observations. However, when $h \geq 2$, $rRMSFE$ falls below one again. Thus, even when the previous observation y_{t-1} contributes the most, GRNN still outperforms $AR(1)$ for some longer forecasting horizons. Remarkably, n influences the relative forecasting performance. When more data are observed, the better GRNN can capture an $AR(4)$ process with a dominating $AR(1)$ component.

Case 3.1: $ARMA(3,3)$**.**

We extended the $AR(p)$ model by adding moving-average components. The $ARMA(p,q)$ model became popular after Box and Jenkins proposed a reasonable way to determine autoregressive and moving average orders. This approach has been predominant in parametric time-series analysis for decades. However, as argued in Hannan and Deistler (2012), this approach performs well only for low-order polynomials for p and q (three or less). Considering this, we found it fruitful to examine the $ARMA(3,3)$ case. $ARMA(3,3)$ is highly volatile, even though it is stationary upon selecting parameters. A handful of significant autoregressive terms alongside the moving-average components make estimating and conducting a prolific forecast analysis cumbersome. This specification has been applied in the literature, to examine future numbers of monthly active worldwide Facebook and Twitter users (Al-Haija et al., 2019), for example.

As a benchmark, the following $ARMA(3,3)$ was used:

$$y_t = 0.8y_{t-1} - 0.3y_{t-2} - 0.5y_{t-3} - 0.4\epsilon_{t-1} + 0.2\epsilon_{t-2} + 0.1\epsilon_{t-3} + \epsilon_t, \tag{9}$$

where ϵ_t is a white noise. In Figure 8, $rRMSFE$ for a baseline ARMA(3,3) is illustrated. GRNN outperforms $AR(1)$ as expected, yielding lower $rRMSFE$ values for all values of n and h. Even with few observations, GRNN still produces better predictions than the benchmark. Based on our Monte Carlo simulation results, one may expect that GRNN copes better than $AR(1)$ when a more sophisticated $ARMA(p,q)$ is introduced.

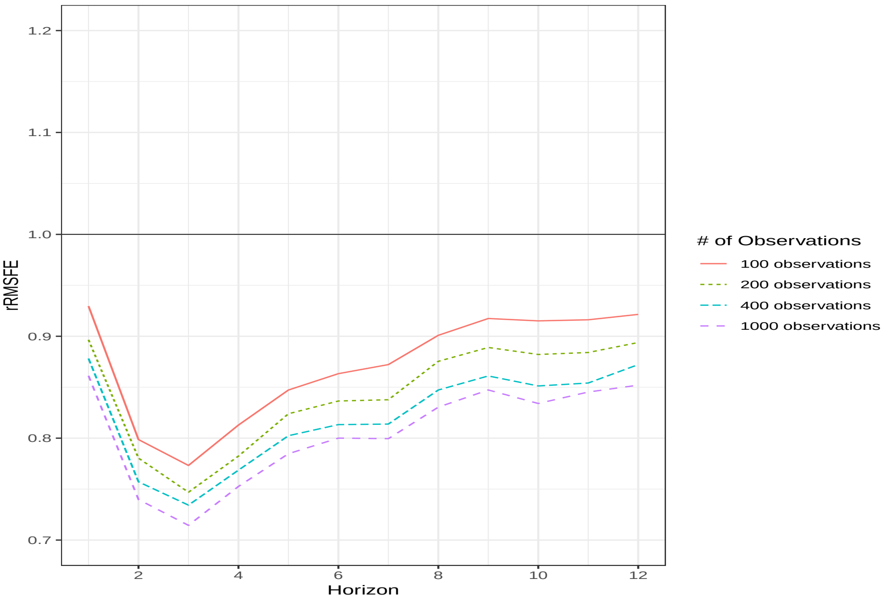

Figure 8: Case 3.1: $ARMA(3,3)$

Case 4.1: $VAR(2)$

Vector Autoregressive Models of order p, denoted as $VAR(p)$ are extensively used in macroeconomic forecasting. $VAR(p)$ specifications imply that the set of current observations of given indicators can be explained by corresponding past values of the same variables involved in the analysis (Lütkepohl, 2013).

For our Monte Carlo simulations, we choose the following bivariate specification of a $VAR(2)$:

$$y_{1,t} = 0.2y_{1,t-1} + 0.1y_{2,t-1} - 0.4y_{1,t-2} + 0.3y_{2,t-2} + \epsilon_{1,t} \tag{10}$$

$$y_{2,t} = -0.3y_{1,t-1} + 0.4y_{2,t-1} + 0.2y_{1,t-2} - 0.3y_{2,t-2} + \epsilon_{2,t}, \tag{11}$$

where $\epsilon_{1,t}$ and $\epsilon_{2,t}$ are i.i.d. with $\epsilon_{k,t} \sim N(0,1)$, $k = 1,2$.

In Figure 9, the $rRMSFE$ values for $VAR(2)$ are presented. GRNN is superior to $AR(1)$ for short-term forecasting when the true DGP is a $VAR(2)$. This indicates that GRNN, even with a relatively small sample size, is able to cope better with the missing variable(s) challenge better than an $AR(1)$.

Case 5.1: $SETAR(2,2)$

The threshold autoregressive model (TAR) is another extension of the $AR(p)$ model. In the TAR framework, a regime-switching behavior is introduced, which makes TAR models piecewise linear. More formally, thresholds split one-dimensional Euclidean space into several subspaces (regimes), with a separate $AR(1)$ in each of those regimes (Gibson and Nur, 2011). This division implies that TAR models are nonlinear but stay linear locally.

The simplest class of TAR models is the self exciting threshold autoregressive model of orders p

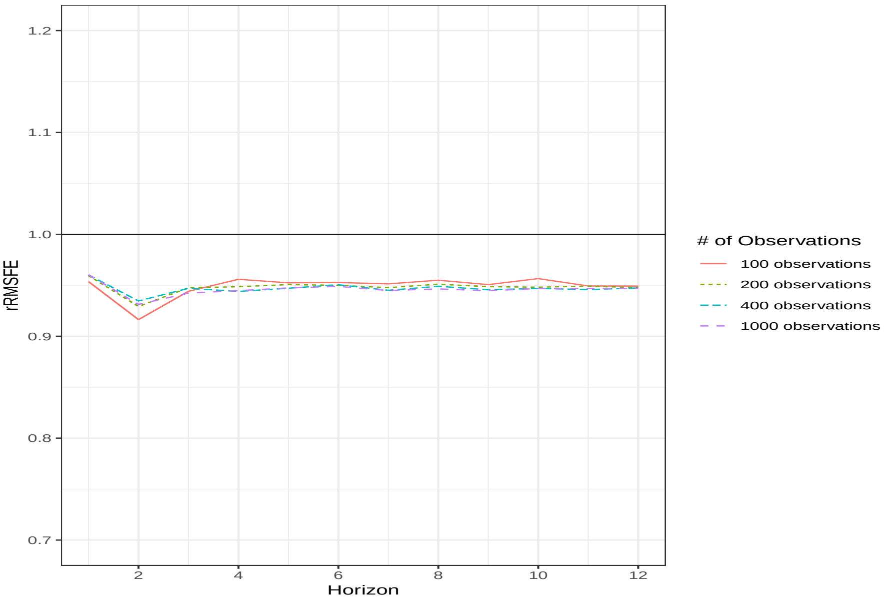

Figure 9: Case 4.1: $VAR(2)$

and q, denoted $SETAR(p, q)$, where p is the number of regimes, and q is the order of the autoregressive part. SETAR implies that the regime depends on lagged values of the dependent variable.

For our Monte Carlo simulations, we chose a $SETAR(2, 2)$ specification proposed in Li and Tong (2016):

$$y_t = \begin{cases} 1 - 0.3y_{t-1} + 0.5y_{t-2} + \epsilon_t, & \text{if } y_{t-2} \leq 0.2 \\ -1 + 0.6y_{t-1} - 0.3y_{t-2} + \epsilon_t, & \text{if } y_{t-2} > 0.2, \end{cases} \tag{12}$$

where ϵ_t is i.i.d. with $\epsilon_t \sim N(0, 1)$.

In Figure 10, the $rRMSFE$ values for $SETAR(2, 2)$ are presented. When nonlinearity is introduced, GRNN is superior to $AR(1)$ for short-term forecasting. Moreover, the relative forecasting performance of GRNN for $h \leq 2$ is declines with respect to n. When the number of observations is low, GRNN produces a smaller value of $rRMSFE$. This pattern is similar to the $VAR(2)$ case. Thus, GRNN can capture a nonlinearity even with a small sample by adjusting its weighting.

Case 6.1: Logistic Map

Additionally, we applied the same Monte Carlo simulation to another class of nonlinear univariate time series, namely a logistic map. Formally, the logistic map could be represented as:

$$y_t = ry_{t-1}(1 - y_{t-1}) \tag{13}$$

The behavior of this nonlinear DGP critically depends on r. We chose $r = 4$, as under this condition, observed values eventually fall within the interval $[0, 1]$. In Figure 11, the $rRMSFE$ values are compared to $AR(1)$. As in the previous case of $SETAR(2, 2)$, GRNN tends to outperform $AR(1)$ for short-term horizons once nonlinearity is introduced.

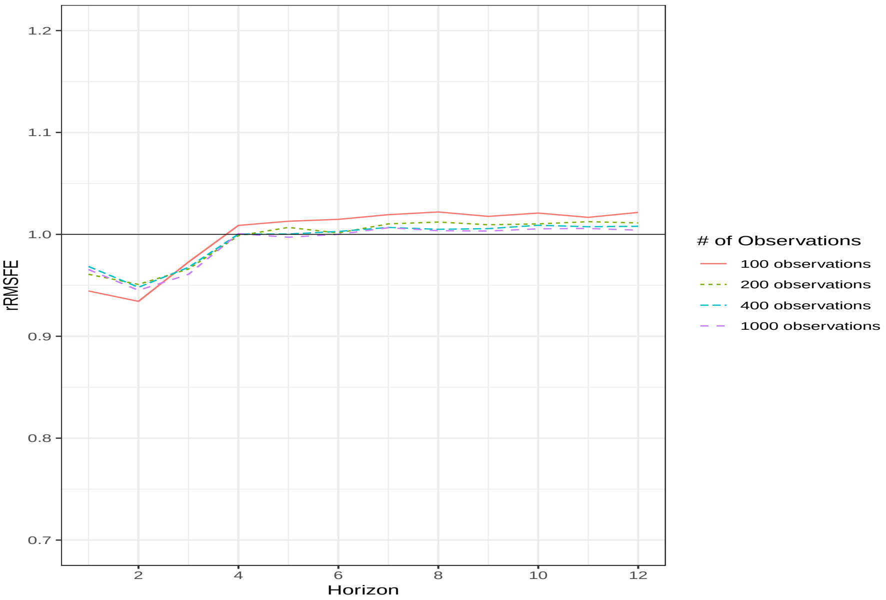

Figure 10: Case 5.1: $SETAR(2, 2)$

Discussion of Simulation Results

In this subsection, some broader results are addressed. First, we compare the forecasting performance across different DGPs. Figure 12 presents the values of $rRMSFE$ for selected DGPs with fixed $n = 200$. Almost all $rRMSFE$ curves are either close to one or fall below this threshold. For $n = 200$, on average, GRNN produces better or at least comparable predictions to $AR(1)$ for every forecasting horizon h except if the true DGP is an $AR(1)$. This observation is important because it signals that GRNN – a less restrictive algorithm – may serve as a benchmark instead of $AR(1)$ in many empirical studies.

It is worth noting, that there might be a trade-off between computational time and forecast precision. For example, the neural-network specification used in Richardson et al. (2018) takes approximately 31 minutes to produce a forecast. However, time-series GRNN is significantly faster than other NN specifications. The average time to fit and predict with GRNN ranges from 0.4-0.6 seconds. For comparison, implementing the same procedure using an ARMA model takes 0.1-0.3 seconds. In other words, an appropriate approach for finding σ and the number of lags allows us to estimate time-series GRNN almost as fast as classic parametric frameworks.

3.3 Parametric vs Nonparametric Approach

Suppose a true DGP is unknown, and one carries out a parametric time series analysis. In that case, various techniques are used to fit an ARMA model (among others, different information criteria and inspection of autocovariance and partial autocovariance). If the process is indeed an $AR(4)$ or $ARMA(3, 3)$, it is unlikely that the estimated model ends up being $AR(1)$. We address the issue in this subsection.

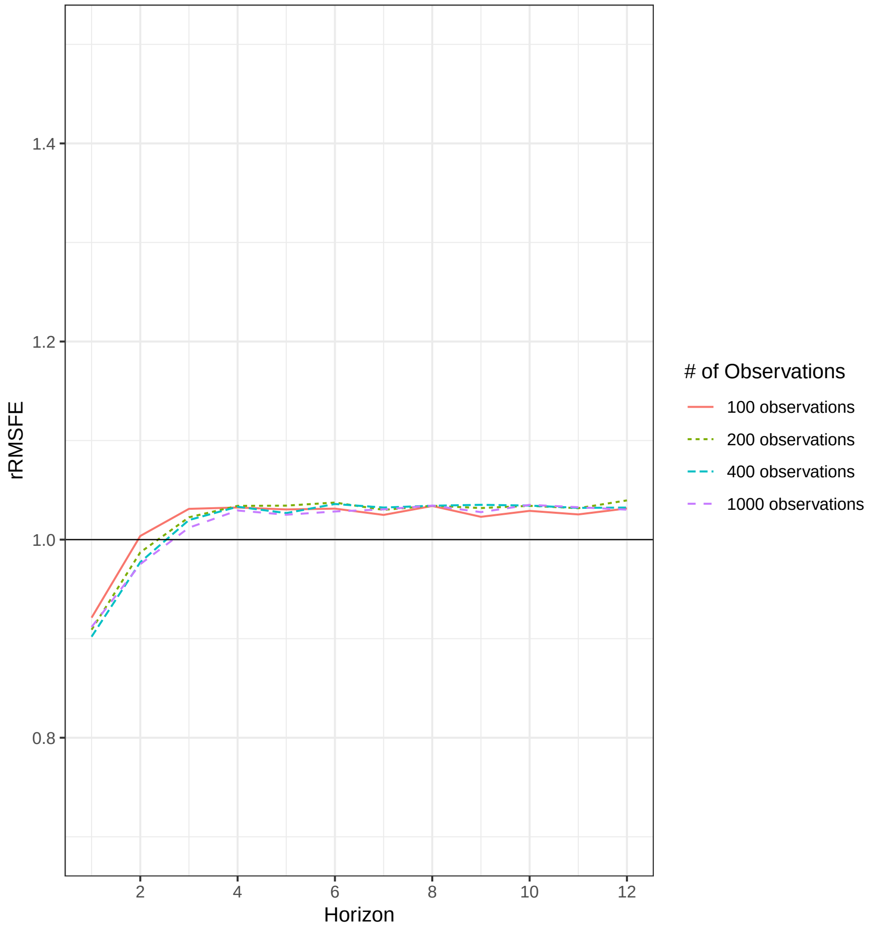

Figure 11: Case 6.1: Logistic Map

First, comparing different approaches to $AR(1)$ is not as unfair as it may seem. $AR(1)$ is extensively used as a general benchmark for various forecast exercises. We show that, under different settings, GRNN – without assuming any functional form – may serve as a more flexible approach for comparison. Even if a sample size is relatively small, GRNN is a valid approach, outperforming the more common $AR(1)$ model.

We intentionally did not collate a non-parametric approach to a true DGP model. Once the true DGP is available, there is no need to use other, even sophisticated, approaches because the true model always yields better forecasts.

However, the true underlying process could rarely be detected. Thus, a common approach is to apply model selection criteria. We applied the ARMA fitting procedure to the same samples and compare forecast precision with GRNN using the same metric, $rRMSFE$, as defined in (4). The results show that ARMA, in most cases, dominates GRNN (Figure 13). However, we highlight two potential problems when dealing with automatically fitted ARMA models. Because we simulated data from well-defined DGPs and knew the exact functional forms, it is possible to empirically test how well generated data could be fitted to a parametric setting using existing frameworks. For our purposes, we used an information criteria approach, i.e., the selection was based on Akaike Information Criteria

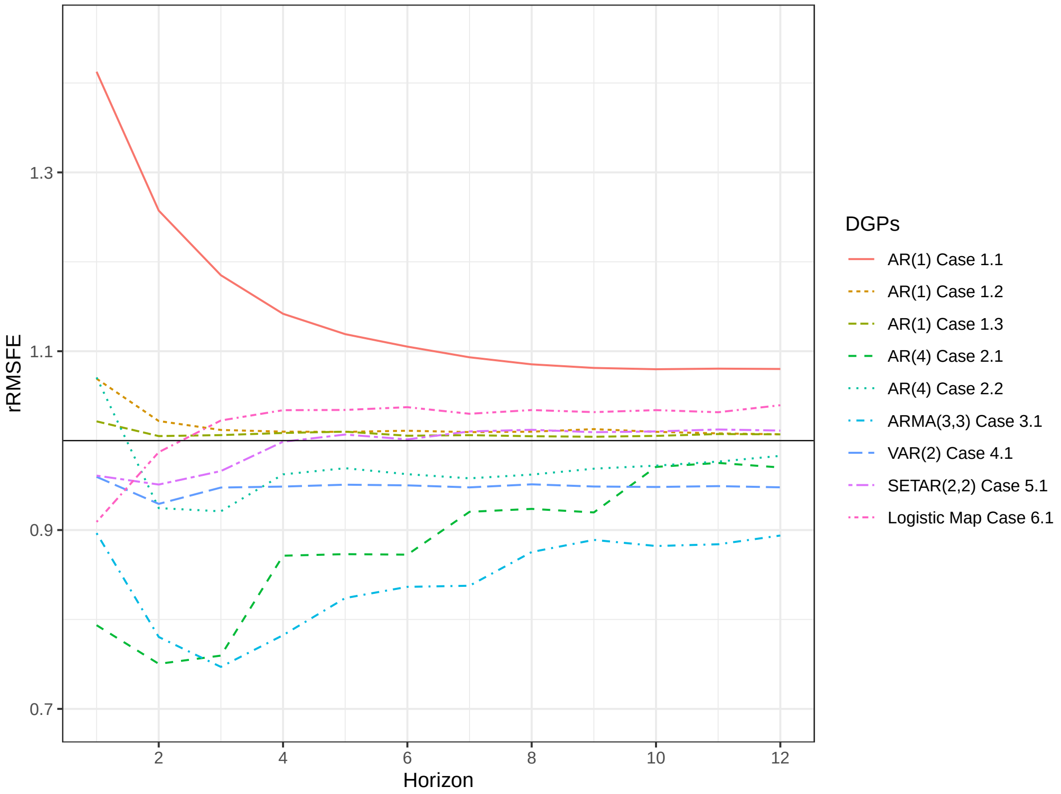

Figure 12: Simulation results for $n = 200$ for different DGPs

DGPs	$n = 100$	$n = 200$	$n = 400$	$n = 1000$
$AR(1)$,case 1.1	0.7364	0.5966	0.5302	0.4602
$AR(1)$, case 1.2	0.5176	0.4789	0.4959	0.5039
$AR(1)$, case 1.3	0.4899	0.3051	0.2869	0.3758
$AR(4)$, case 2.1	0.5365	0.4896	0.4902	0.4804

Table 1: Share of correctly estimated samples using ARMA fitting

(AIC) values (max 5 lags). The same samples of $AR(1)$ and $AR(4)$ (case 2.1), on which we performed our Monte Carlo simulations, were chosen. Table 1 presents shares of correctly estimated models given the true DGPs. Even though a parametric approach seems to be rather restrictive for almost all DGPs, $rRMSFE$ of the fitted ARMA model is superior to GRNN estimation. Table 1 shows that parametric ARMA estimations often cannot detect the true model. In the best scenario of $AR(1)$ with high persistence, approximately 75% of simulated samples were correctly detected. All other DGPs had a lower match. Real-world data is barely distributed according to well-defined parametric models. In a given situation, an incorrectly-specified ARMA model may yield unreliable forecasts. Thus, we conclude that GRNN – as a less restrictive model – may indeed compete with parametric frameworks when dealing with univariate time series.

Second, ARMA fitting cannot correctly capture nonlinear time series, at least for short horizons,

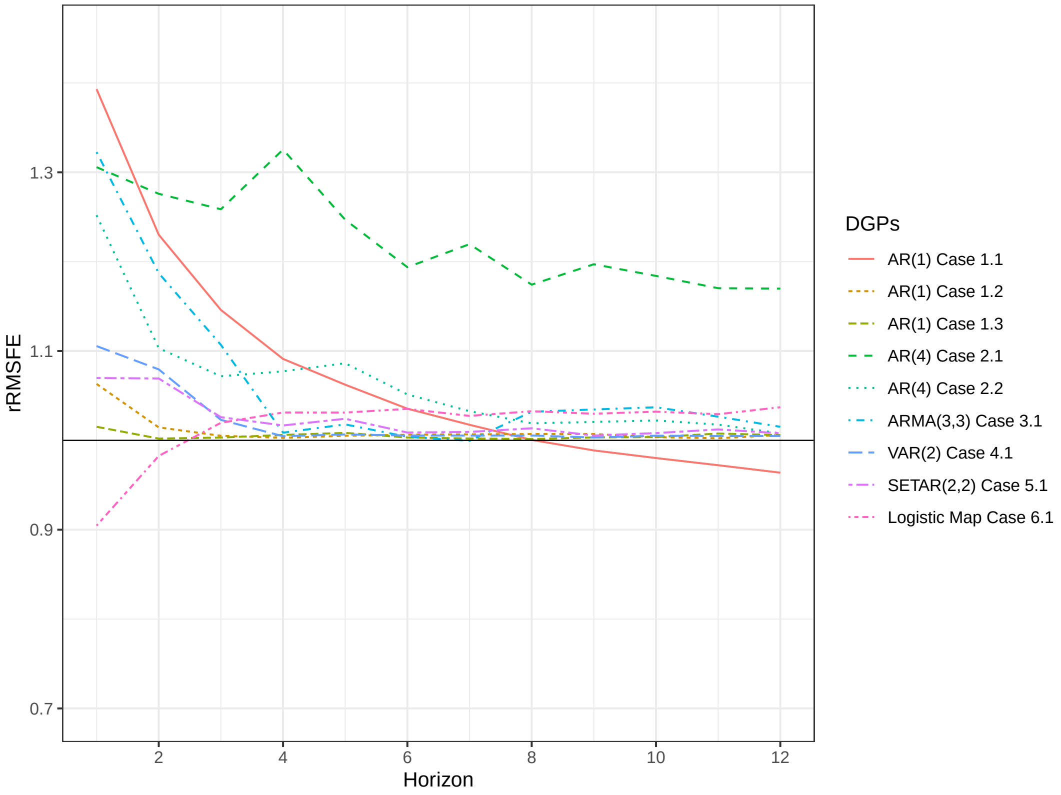

Figure 13: Simulation results for $n = 200$ for different DGPs compared to fitted ARMA

as case 6.1 (logistic map) illustrates. Moreover, a similar pattern of $rRMSFE$ values was observed in all figures. This can be attributed to the fact that ARMA fitting procedure often assumes that the DGP follows $AR(1)$, thus failing to detect existing nonlinearity. However, GRNN can capture the nontrivial behaviour of given logistic map samples, providing superior short-term forecasts.

4 Nowcasting the German GDP

4.1 Empirical Strategy

To obtain a GDP nowcast (both level value and growth rate) for a given quarter we propose an approach based on GRNN time-series prediction. Thus far, we have demonstrated the potential of GRNN-based models to predict univariate time series. However, to estimate the current GDP value we should consider numerous different time series, some of which may be observed at higher frequencies than quarterly. We propose a specific extension of the GRNN framework, enabling the use of univariate GRNN time-series predictions to nowcast the GDP and exploit monthly indicators.

Contemporaneous values of some crucial macroeconomic variables are not observed within a quarter. In the case of Germany, the first official estimates of the overall economic development are available only when GDP figures are released, that is, 30 days after the end of a quarter.[4] However,

[4] https://www.destatis.de/EN/Service/EXDAT/Datensaetze/early-indicator-economic-development.html

it is possible to estimate the GDP using higher frequency variables (monthly, daily) that are observed and published within a quarter (Giannone et al., 2008).

As a preliminary step of our analysis, actual (level) values of monthly time series $y_{t,n}$, $n = 1, \ldots, N$, are aggregated to match the quarterly frequency. We denote aggregated time series $y^Q_{t,n}$. Let us consider $S_{y_t} = \{y^Q_{t,1}, y^Q_{t,2}, ..., y^Q_{t,N}\}$ a set of all aggregated time series $y^Q_{t,n}$ selected to nowcast GDP for any given quarter t. We define $\tilde{y}^Q_{t,n}$ as actual observations of time series $y^Q_{t,n} \in S_{y_t}$ divided by the corresponding value of GDP_t:

$$\tilde{y}^Q_{t,n} = \frac{y^Q_{t,n}}{GDP_t} \quad y^Q_{t,n} \in S_{y_t} \tag{14}$$

We chose this specification for two main reasons. First, taking a ratio relates to a log transformation. Additionally, the actual GDP values, unlike some indicators, are far from zero. Therefore we need not to be concerned about putting them into the denominator. Let us now consider a set of all aggregated time series $S_{\tilde{y}} = \{\tilde{y}^Q_{t,1}, \tilde{y}^Q_{t,2}, ..., \tilde{y}^Q_{t,N}\}$ transformed according to (14). Each entity of the set depicts the dynamics of an observed time series relative to GDP.

When we obtain $S_{\tilde{y}}$, we can separately train each ratio $\tilde{y}^Q_{t,n} \in S_{\tilde{y}}$ on all available information using GRNN with transformation (additive and multiplicative) and without transformation and conduct a one-step-ahead forecast. As we argue further, it seems necessary to apply transformations because some $\tilde{y}^Q_{t,n}$ are in fact trend-stationary or non-stationary. Thus, we have three different point nowcasts for every $\tilde{y}^Q_{t,n}$. We denote obtained nowcast $\hat{\tilde{y}}^Q_{t,n}$ and a set of all "bivariate" nowcasts as $S_{\hat{\tilde{y}}} = \{\hat{\tilde{y}}^Q_{t,1}, \hat{\tilde{y}}^Q_{t,2} .... \hat{\tilde{y}}^Q_{t,N}\}$. Finally, after predicting $\hat{\tilde{y}}^Q_{t,n} \in S_{\hat{\tilde{y}}}$ we retrieved the GDP nowcasts using the following relation:

$$\widehat{GDP}_{t,n} = \frac{y^Q_{t,n}}{\hat{\tilde{y}}^Q_{t,n}} \tag{15}$$

We obtained N different "bivariate" GDP nowcasts for a given quarter t. We collected all the nowcasts in a set $S_{GDP} = \{\widehat{GDP}_{t,1}, \widehat{GDP}_{t,2} ... \widehat{GDP}_{t,N}\}$. As a final prediction we use the median of this set as in Stock and Watson (1999), for example. The median is more robust than the mean because of potential outliers. It is also possible to estimate a trimmed mean – the value one can estimate after eliminating a certain percentage of the dataset's smallest and largest values. However, to produce our nowcast, we used the median because it allowed us to use all available data without any imposed restrictions. It is also possible to obtain the same set as S_{GDP} but with growth rates by modifying each entity accordingly.

4.2 Data

Our data set consisted of 125 German macroeconomic indicators, and the sample covers the period from first quarter of 1991 to the second quarter of 2022. Each time series was seasonally adjusted.

Our proposed method allows us to explicitly capture a ragged edge phenomenon using only those time series for which we have at least one observation within a quarter t and aggregating it to quarterly frequency. We considered two different scenarios based on data availability:

1. *Real-time nowcasting*: We used the data available one month into the quarter. Missing monthly values were forecasted using either GRNN with transformations or ARMA models.

2. *Full data backcasting*: We used the complete data for a given quarter, assuming access to all monthly and quarterly data. Under these settings, each model should perform better due to the availability of more information.

To compared the forecasting power of each model, we selected 15 quarters (Q1 2020 to Q3 2023) for real-time nowcasting and 22 quarters (Q1 2017 to Q2 2022) for complete data backcasting, and conduct a one-step-ahead pseudo-out-of-sample forecast. For example, we used each indicator to produce a nowcast for Q2 2022, including all available information within Q2 2022. Based on this information, the current state of GDP was evaluated. The performance of each model was then calculated by comparing an estimation to the actual value of GDP using (3) and (4). This procedure was repeated for each pre-selected quarter. The periods chosen for the empirical analysis included the COVID-19 crisis because it is useful to study the performance of the models under highly volatile conditions.

4.3 Benchmark Selection

We compare the GRNN predictions to nowcasts produced by two German research institutes. The ifo Institute publishes their GDP nowcasts, known as ifoCAST, updating them twice a month based on available data Lehmann et al. (2020). The core of ifoCAST is a dynamic factor model with mixed frequencies. Another institute is the IWH, whose nowcasts (IWH Flash) are considered advantageous for nowcasting German economic activity (Claudio et al., 2020; Heinisch and Scheufele, 2018). Unlike the ifo, the IWH publishes its nowcasts just one month into a quarter. To ensure a fair comparison, we used only the data available one month into the quarter for this exercise.

We select the following benchmarks to compare the full backcasting performance, for which no predictions are available from both institutions. As a baseline model, $AR(1)$ for quarterly GDP was chosen as in Richardson et al. (2018). In addition, a more simplified model averaging framework based on linear regression was estimated. This approach is similar to our proposed method. However, instead of using GRNN, we ran N bivariate OLS regressions to obtain a similar set of GDP predictions. After collecting all the predictions, a simple mean (which is superior in terms of forecast accuracy to taking a median) was taken to produce a final forecast. This procedure loosely mimics the methodology of IWH Flash. Additionally, we estimated a simple dynamic factor model (DFM), in which the optimal number of factors to incorporate in the model was chosen according to Bai and Ng (2002). The lag order of the factor-VAR was selected based on AIC.

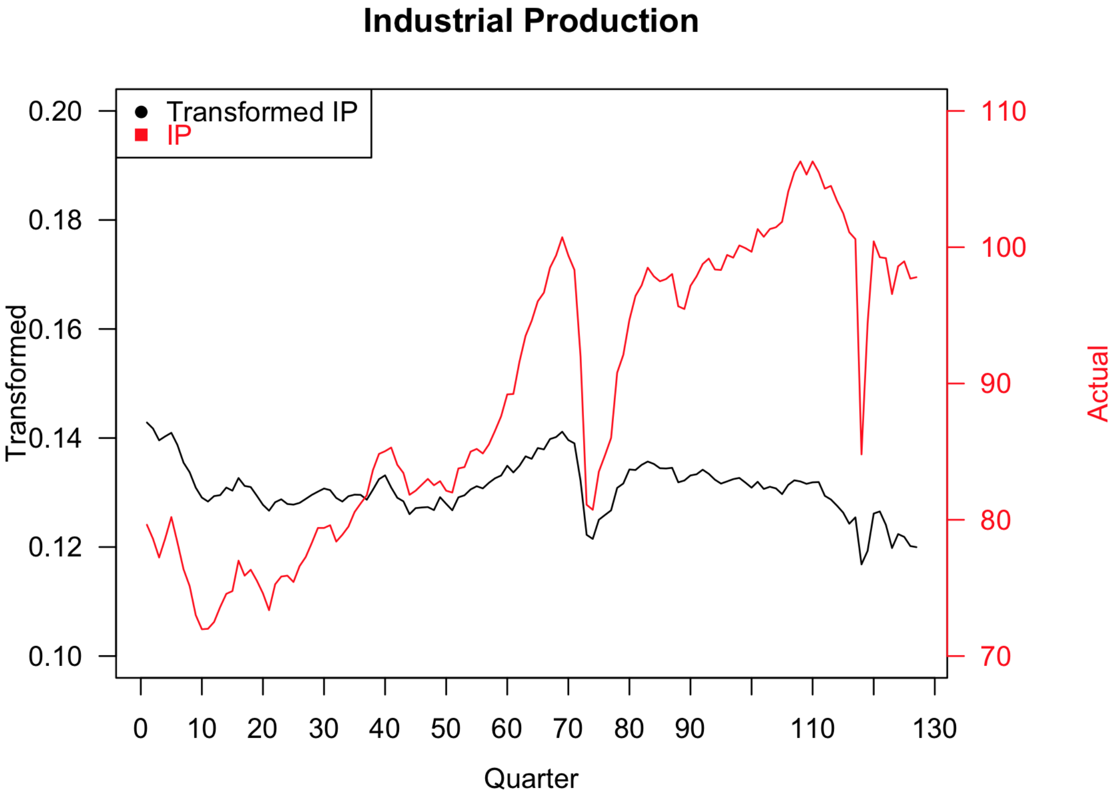

Figure 14: Actual and scaled IP values (Q1 1991 to Q2 2022)

4.4 Scaled Data for the GRNN

GDP exhibits a positive trend over time. Various other time series are characterized either by positive or negative trends. However, a handful of indicators are relatively stable over time. This implies that for some $y_{t,n}^{Q}$ its counterpart $\tilde{y}_{t,n}^{Q}$ may be stationary if it mimics the GDP dynamics. However, most $\tilde{y}_{t,n}^{Q}$ are non-stationary. To highlight a potential difference in dynamics of $\tilde{y}_{t,n}^{Q}$, we discuss two crucial indicators, namely, the industrial production index (IP) and the ifo Business Climate Index.

IP measures the value added in the manufacturing sector and represents the output value less the values of intermediate consumption and consumption of fixed capital. IP is an important indicator used to assess the current state of economic activity (Eraslan and Götz, 2021). In Figure 14, the actual (red line) and transformed IP (black line, scaled by GDP) values are depicted.

The ifo Business Climate Index is one of the most important early indicators for economic development in Germany. This indicator has been extensively used in the literature to predict GDP and IP (Lehmann, 2022). In Figure 15, the actual (red line) and scaled (black line) values of the ifo Business Climate Indexes are depicted. By construction, the actual values of the ifo Business Climate Index, unlike IP, are bounded. Thus, oscillations rather than a clear trend are observed. Correspondingly, this forces the scaled time series to decay slowly over time, making it non-stationary. Nonetheless, similarities in dynamics between the two lines are observed, indicating a positive association between

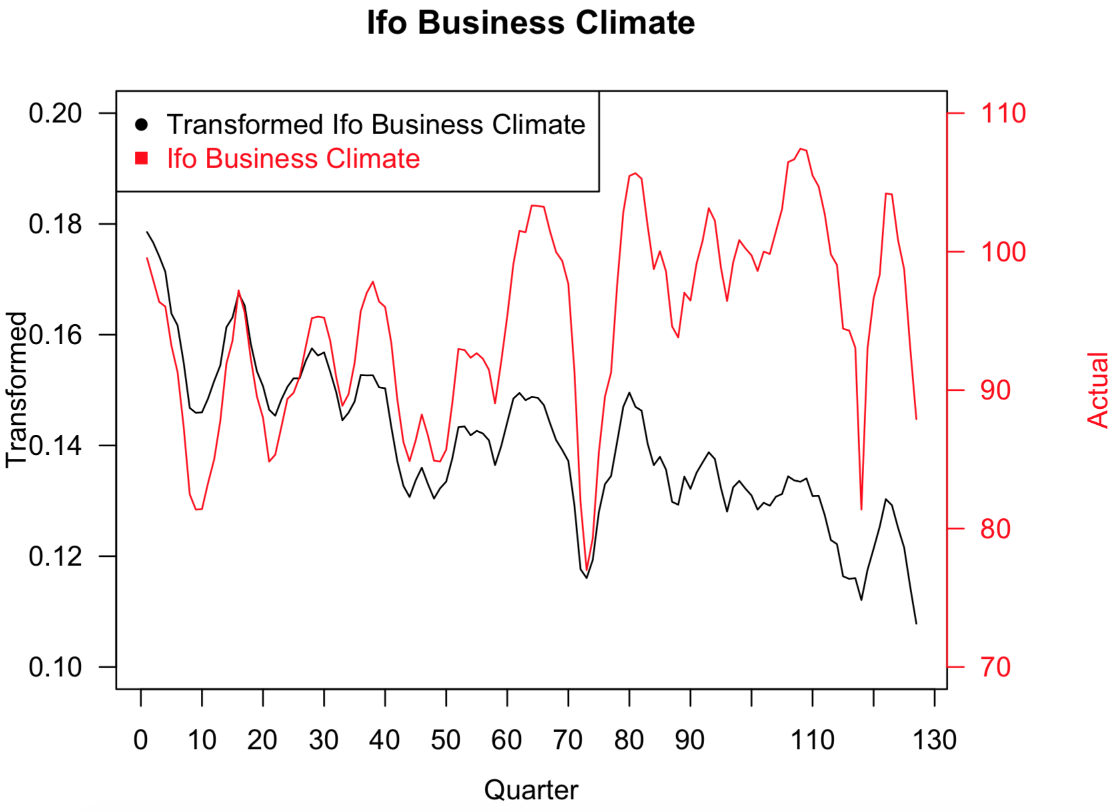

Figure 15: Actual and scaled ifo Business Climate Index (Q1 1991 to Q2 2022)

GDP and the ifo Business Climate Index.

4.5 Results

Real-Time Nowcasting

First, we analyzed the nowcasting performance. Table 2 depicts both $MAFE$ and $RMSFE$ for all pre-specified approaches. Additionally, we computed the combined value (GRNN combined), in which we aggregated all GRNN estimates (with additive, multiplicative and no transformations) into one set and took the median of all the predictions. In this scenario, we computed missing monthly values by using GRNN with the multiplicative transformation.[5]

If we include the COVID-19 crisis (Q1 2020 to Q3 2020), both benchmarks, ifoCAST and IWH Flash, outperform GRNN (columns 1-2). However, IWH Flash incorporates expert knowledge. Meanwhile, the estimations for all GRNN specifications and ifoCAST are solely based on the models' outcomes. This is why the IWH Flash performs considerably better than other selected models in the real-time setting. However, if we discard the COVID-19 data (columns 3-4), GRNN is superior to existing nowcasting approaches for German GDP, yielding lower results for $RMSFE$ and $MAFE$.

[5] Similar results are obtained if missing values are forecasted using ARMA or GRNN with additive transformation.

	Q1 2020 - Q3 2023		Q4 2020 - Q3 2023	
Model	$MAFE$	$RMSFE$	$MAFE$	$RMSFE$
GRNN additive	1.48	2.63	0.55	0.70
GRNN no transformation	1.90	2.96	0.96	1.20
GRNN multiplicative	1.39	2.62	0.42	0.60
GRNN combined	1.39	2.62	0.42	0.59
ifoCAST	1.44	1.94	0.99	1.31
IWH Flash	1.04	1.62	0.60	1.01

Table 2: Real-time nowcasting performance with monthly data forecast by GRNN with the multiplicative transformation

Model	$MAFE$	$rMAFE$	p-value	$RMSFE$	$rRMSFE$	p-value
GRNN additive	0.96	0.56	0.033	1.82	0.51	0.084
GRNN no transformation	0.83	0.48	0.070	1.21	0.34	0.087
GRNN multiplicative	1.00	0.59	0.031	1.92	0.54	0.087
GRNN combined	0.94	0.55	0.032	1.75	0.49	0.084
OLS model averaging	1.27	0.74	0.064	2.55	0.71	0.123
DFM	1.57	0.91	0.205	3.06	0.86	0.144
AR(1)	1.71	1	–	3.57	1	–

Table 3: Backcasting performance (Q1 2017 - Q2 2022)

Full Data Backcasting

Table 3 presents $MAFE$ and $RMSFE$, respectively, as well as $rMAFE$ and $rRMSFE$. Each GRNN-based model outperforms the other models in terms of $MAFE$ and $RMSFE$. On average, GRNN approaches can reduce the forecasting error by almost 50% compared to the $AR(1)$ model. The model averaging approach and DFM yield better nowcasts than the benchmark over the entire period. Additionally, we conducted the one-sided Diebold-Mariano (DM) test with the Harvey-Leybourne-Newbold correction for small samples to check whether two forecasts have the same accuracy.[6] We had to use a corrected version of the DM test since our sample size is not large enough to implement the classic version. Corresponding p-values are reported in Table 3. The DM test results using absolute errors demonstrate that we can reject the null hypothesis when comparing each GRNN-based model and the model averaging approach to $AR(1)$ at 10%. Similar results were obtained using a quadratic loss function instead of $MAFE$.

[6] The alternative hypothesis is that the first forecast is more accurate than the second forecast, namely $AR(1)$.

Model	$rMAFE$	$rRMSFE$
GRNN additive	0.53	0.50
GRNN no transformation	0.30	0.29
GRNN multiplicative	0.58	0.53
GRNN combined	0.52	0.48
Model averaging	0.69	0.71
DFM	0.82	0.84
AR(1)	1	1

Table 4: Backcasting performance (Q1 2020 - Q1 2021)

Below, we will discuss how the models performed during the COVID-19 crisis. Table 4 shows the corresponding $rMAFE$ and $rRMSFE$ values for first quarter of 2020 to first quarter of 2021. Generally, for most models, the results are similar to Table 3, except for GRNN without data transformation. This model performs significantly better from Q1 2020 to Q1 2021, most accurately predicting the current state of economic activity. However, if we look at the quarters before the COVID-19 pandemic, i.e., Q1 2017 to Q4 2019, and after this period, namely Q2 2021 to Q2 2022, $rMAFE$ and $rRMSFE$ for GRNN with no transformation yield 1.2 and 1.3, respectively. Recall that a classic GRNN model cannot capture any trend by construction. Therefore during the expansion, this specification cannot produce reliable estimations, unlike, for example, GRNN with additive transformation.[7] However, GRNN with unmodified data may perform substantially better during the crisis, relying only on previously observed data.

Surprisingly, there is no additional gain from using GRNN with a stationarity test in advance. This can be explained as follows: GRNN with multiplicative and additive transformations on average yield closely-related estimations for one-step-ahead forecasting. Thus, under the proposed framework of obtaining different "bivariate" nowcasts and taking a median of the set, time series differentiation due to non-stationarity is redundant.

Overall, an approach based on GRNN with additive transformation should be more reliable than its counterpart (multiplicative) in the case of Germany. However, if one expects a rather drastic drop in economic activity, then GRNN without transformation will better capture this phenomenon.

5 Conclusions

We demonstrate the advantages of using a GRNN-based approach for macroeconomic nowcasting. First, we documented the properties of GRNN forecasts in a Monte Carlo investigation. We simulated

[7] $rMAFE$ and $rRMSFE$ for Q1 2017 to Q4 2019 are 0.7 and 0.8, respectively. Meanwhile, in the post-Covid quarters, both metrics yield 0.6.

different univariate DGPs and compared the forecasting performance of GRNN and $AR(1)$. In many cases, the GRNN approach outperformed the benchmark. Additionally, we showed that ARMA fitting approaches, even though they cannot properly identify the true DGP in many cases, yield better predictions if the true DGP is linear. However, unlike GRNN, standard parametric approaches based on ARMA cannot detect nonlinearity. We argue that it is better to use a non-parametric approach, which is more flexible, rather than imposing restrictive and often incorrect assumptions about underlying distributions.

We propose a new method of using a univariate time-series GRNN model to nowcast German GDP. First, specific data scaling needs to be implemented, i.e., each indicator is divided by GDP. Then, these ratios are used to perform one-step-ahead forecasting using GRNN. Afterward a set of GDP nowcasts is obtained using actual aggregated observations within a given quarter. In real-time settings, GRNN can compete with existing approaches. Finally, we show that this algorithm has a high forecasting power, outperforming traditional nowcasting models ($AR(1)$, DFM, model averaging), especially during the COVID-19 crisis once a sufficient amount of data is observed within a given quarter.

References

Ahmed, N. K., Atiya, A. F., Gayar, N. E. and El-Shishiny, H.: 2010, An Empirical comparison of machine learning models for time series forecasting, *Econometric reviews* **29**(5-6), 594–621.

Al-Haija, Q. A., Mao, Q. and Al Nasr, K.: 2019, Forecasting the number of monthly active Facebook and Twitter worldwide users using ARMA model.

Bai, J. and Ng, S.: 2002, Determining the number of factors in approximate factor models, *Econometrica* **70**(1), 191–221.

Bańbura, M. and Rünstler, G.: 2011, A look into the factor model black box: publication lags and the role of hard and soft data in forecasting GDP, *International Journal of Forecasting* **27**(2), 333–346.

Barbaglia, L., Frattarolo, L., Onorante, L., Pericoli, F. M., Ratto, M. and Pezzoli, L. T.: 2022, Testing big data in a big crisis: Nowcasting under COVID-19, *International Journal of Forecasting* .

Claudio, J. C., Heinisch, K. and Holtemöller, O.: 2020, Nowcasting East German GDP growth: a MIDAS approach, *Empirical Economics* **58**(1), 29–54.

Demir, S. and Toktamiş, Ö.: 2010, On the adaptive Nadaraya-Watson kernel regression estimators, *Hacettepe Journal of Mathematics and Statistics* **39**(3), 429–437.

Eraslan, S. and Götz, T.: 2021, An unconventional weekly economic activity index for Germany, *Economics Letters* **204**, 109881.

Gheyas, I. A. and Smith, L. S.: 2009, A neural network approach to time series forecasting, *Proceedings of the World Congress on Engineering*, Vol. 2, pp. 1–3.

Giannone, D., Reichlin, L. and Small, D.: 2008, Nowcasting: The real-time informational content of macroeconomic data, *Journal of monetary economics* **55**(4), 665–676.

Gibson, D. and Nur, D.: 2011, Threshold autoregressive models in finance: a comparative approach.

Hannan, E. J. and Deistler, M.: 2012, *The statistical theory of linear systems*, SIAM.

Hansen, B. E.: 2009, Lecture notes on nonparametrics, *Lecture notes* .

Härdle, W.: 1990, *Applied nonparametric regression*, number 19, Cambridge university press.

Heiler, S.: 1999, A survey on nonparametric time series analysis, *Technical report*, CoFE Discussion Paper.

Heinisch, K. and Scheufele, R.: 2018, Bottom-up or direct? forecasting german gdp in a data-rich environment, *Empirical Economics* **54**, 705–745.

Kuzin, V., Marcellino, M. and Schumacher, C.: 2011, MIDAS vs. mixed-frequency VAR: Nowcasting GDP in the euro area, *International Journal of Forecasting* **27**(2), 529–542.

Lehmann, R.: 2022, The forecasting power of the ifo business survey, *Journal of Business Cycle Research* pp. 1–52.

Lehmann, R., Reif, M. and Wollmershäuser, T.: 2020, ifoCAST: The New Forecast Standard of the ifo Institute, *ifo Schnelldienst* **73**(11), 31–39.

Li, D. and Tong, H.: 2016, Nested sub-sample search algorithm for estimation of threshold models, *Statistica Sinica* pp. 1543–1554.

Lütkepohl, H.: 2013, Vector autoregressive models, *Handbook of research methods and applications in empirical macroeconomics*, Edward Elgar Publishing, pp. 139–164.

Marcellino, M., Stock, J. H. and Watson, M. W.: 2006, A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series, *Journal of econometrics* **135**(1-2), 499–526.

Martínez, F., Charte, F., Frías, M. P. and Martínez-Rodríguez, A. M.: 2022, Strategies for time series forecasting with generalized regression neural networks, *Neurocomputing* **491**, 509–521.

Martínez, F., Frías, M. P., Pérez-Godoy, M. D. and Rivera, A. J.: 2022, Time series forecasting by generalized regression neural networks trained with multiple series, *IEEE Access* **10**, 3275–3283.

Nadaraya, E. A.: 1964, On estimating regression, *Theory of Probability & Its Applications* **9**(1), 141–142.

Richardson, A., Mulder, T. et al.: 2018, Nowcasting New Zealand GDP using machine learning algorithms.

Sheather, S. J.: 2004, Density estimation, *Statistical science* pp. 588–597.

Silverman, B. W.: 2018, *Density estimation for statistics and data analysis*, Routledge.

Specht, D. F. et al.: 1991, A general regression neural network, *IEEE transactions on neural networks* **2**(6), 568–576.

Stock, J. H. and Watson, M. W.: 1999, Forecasting inflation, *Journal of monetary economics* **44**(2), 293–335.

Sumiyati, S. and Warsito, B.: 2020, Comparison of Suspended Particulate Matter Prediction Based on Linear and Non-Linear Models, *IOP Conference Series: Earth and Environmental Science*, Vol. 448, IOP Publishing, p. 012029.

Wang, H. and Suter, D.: 2004, Robust adaptive-scale parametric model estimation for computer vision, *IEEE transactions on pattern analysis and machine intelligence* **26**(11), 1459–1474.

Watson, G. S.: 1964, Smooth regression analysis, *Sankhyā: The Indian Journal of Statistics, Series A* pp. 359–372.

Yan, W.: 2012, Toward automatic time-series forecasting using neural networks, *IEEE transactions on neural networks and learning systems* **23**(7), 1028–1039.

Zambom, A. Z. and Ronaldo, D.: 2013, A review of kernel density estimation with applications to econometrics, *International Econometric Review* **5**(1), 20–42.

Appendices

A Nadaraya-Watson Gaussian Kernel Regression Estimator

Let us assume that for given pairs $(X_i, Y_i)_{i=1}^N$, we have the following model:

$$Y_i = g(X_i) + \epsilon_t, \tag{16}$$

where function $g(\cdot)$ is unknown.

Additionally, we assume that the error terms ϵ_t are i.i.d random variables with mean 0 and some variance σ^2. In this case, $g(\cdot)$ is a conditional mean $E(y|x)$ or alternatively (Demir and Toktamiş, 2010):

$$g(x) = E(y|x) = \int \frac{yf(x,y)}{f(x)} dy, \tag{17}$$

where $f(x,y)$ is the joint density function of (X, Y), whereas $f(x)$ is the marginal density.

Applying the kernel density estimato, a non-parametric method, which estimates a probability density function of a given random variable based on kernels, for both $f(x,y)$ and $f(x)$, one can obtain the Nadaraya-Watson kernel estimator of the regression function $g(\cdot)$:

$$\begin{aligned} \hat{g}(x) = \hat{E}(y|x) &= \frac{\sum_{i=1}^N Y_i K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})} = \sum_{i=1}^N \frac{K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})} Y_i = \sum_{i=1}^N W_i(x) Y_i \\ W_i(x) &= \frac{K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})}, \end{aligned} \tag{18}$$

where $K(\cdot)$ is a (symmetric) non-negative integrable function called kernel, h is a bandwidth parameter, W_i is a set of weights, which sum up to 1.

A kernel function, denoted as $K(\cdot)$, is an arbitrary function that satisfies the following general conditions (inter alia, see Silverman (2018)):

- $\int_{-\infty}^{\infty} K(u) du = 1$
- $K(u) = K(-u)$ – in most cases symmetry is assumed
- $\int_{-\infty}^{\infty} uK(u) du = 0$ – first moment equals to 0
- $\int_{-\infty}^{\infty} u^2 K(u) du \neq 0$ – second moment differs from 0

Although plenty of functions satisfy the properties mentioned above, choosing a specific kernel should not significantly influence the precision of the density estimation. Still, the most commonly used kernels in empirical studies are Epanechnikov and Gaussian (Härdle, 1990; Hansen, 2009). In our study, we selected the Gaussian kernel, which assures the smoothness of a density function and the existence of derivatives of all orders (Zambom and Ronaldo, 2013). More formally, the Gaussian kernel could be expressed as follows:

$$K_G(u) = \frac{1}{\sqrt{2}} \exp\left(-\frac{u^2}{2}\right) \tag{19}$$

Many scholars have argued that the Nadaraya-Watson kernel estimator relies heavily on a selection of bandwidth h, which controls the smoothness of the probability density function estimation (Wang and Suter, 2004; Sheather, 2004). Several procedures have been proposed to find an optimal value of h. These approaches may be divided into two separate classes: quality-of-fit (cross-validation) and plug-in methods (minimizing the mean integrated square error ($MISE$) between the real density and its kernel-based approximation).

Halle Institute for Economic Research –
Member of the Leibniz Association

Kleine Maerkerstrasse 8
D-06108 Halle (Saale), Germany

Postal Adress: P.O. Box 11 03 61
D-06017 Halle (Saale), Germany

Tel +49 345 7753 60
Fax +49 345 7753 820

www.iwh-halle.de

ISSN 2194-2188

Leibniz
Association

The IWH is funded by the federal government and the German federal states.