Holtemöller, Oliver; Kozyrev, Boris

Conference Paper

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

Beiträge zur Jahrestagung des Vereins für Socialpolitik 2023: Growth and the "sociale Frage"

Provided in Cooperation with:
Verein für Socialpolitik / German Economic Association

Suggested Citation: Holtemöller, Oliver; Kozyrev, Boris (2023) : Forecasting Economic Activity with a Neural Network in Uncertain Times: Monte Carlo Evidence and Application to German GDP, Beiträge zur Jahrestagung des Vereins für Socialpolitik 2023: Growth and the "sociale Frage", ZBW - Leibniz Information Centre for Economics, Kiel, Hamburg

This Version is available at:
https://hdl.handle.net/10419/277688

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

Oliver Holtemöller* Boris Kozyrev†

February 28, 2023

Abstract

In this paper, we analyze the forecasting and nowcasting performance of a generalized regression neural network (GRNN). First, we provide evidence from Monte Carlo simulations for the relative forecast performance of GRNN depending on the true but unknown data-generating process. We show that GRNN outperforms autoregressive-moving average models in many practically relevant cases. Second, we apply GRNN to forecast quarterly German GDP growth. We distinguish between "normal" times and situations in which the time-series behavior is very different from "normal" times such as during the COVID recession and recovery. It turns out that GRNN is superior in terms of root mean forecast errors both to an autoregressive model and to more sophisticated approaches like dynamic factor models if applied appropriately.

Keywords: Forecasting, neural network, nowcasting, time series models.

JEL: C22, C45, C53.

*Martin Luther University Halle-Wittenberg and Halle Institute for Economic Research (IWH) oliver.holtemoeller@iwh-halle.de

†Halle Institute for Economic Research (IWH), boris.kozyrev@iwh-halle.de, corresponding author

1 Introduction

Different authorities and decision-makers need to assess the current (sometimes past) state of economic activity. Over the last couple of decades, efforts have been put into developing a framework for forecasting the present or recent past using main macroeconomic indicators sampled at different frequencies (Aastveit et al., 2014). As a result, several approaches have been established, inter alia Dynamic Factor Models (DFM) (Bańbura and Rünstler, 2011), Mixed-Data Sampling (MIDAS), and Mixed-frequency Vector Autoregressive Models (MF-VAR) (Kuzin et al., 2011).

However, during recessions or crises, these forecasting models may yield unreliable estimates. In other words, the prediction quality of such approaches significantly deteriorates during unstable periods. The most recent example is the COVID-19 crisis. As argued in Barbaglia et al. (2022), this pandemic and the corresponding crisis was an unprecedented shock never observed before in modern history. It was nearly impossible to estimate its potential impact on the world economy in real time. The recent pandemic makes researchers look for a more reliable and trustworthy approach, which is valid even during severe downfalls.

In this article, we propose a method of nowcasting based on a simple Neural Network called Generalized Regression Neural Network (GRNN). A supplementary data transformation is also introduced. This model has been selected due to two main reasons. First, this model, unlike other sophisticated NN specifications, is rather intuitive. This fact may serve as an additional reason for decision makers to use GRNN. GRNN is the Nadaraya-Watson Gaussian Kernel Regression estimator defined in terms of a neural network. In other words, this model is assumed to share advantages from both NN and non-parametric approaches. Second, Martínez et al. (2022) proposed a new automatic method to produce a forecast using GRNN, proving that the obtained predictions can compete with different – often less tractable – NN architectures.

We argue a GRNN-based approach has a high forecasting power compared to standard parametric frameworks. First, we conduct Monte Carlo simulations, in which predictions obtained from the GRNN model are compared to the selected AR(1) benchmark given different Data Generating Processes (DGPs). An additional check of fitting ARMA using simulated samples is provided. Finally, we perform a one-step-ahead pseudo-out-of-sample-forecast to predict the actual value of German GDP and its corresponding growth rate. The empirical results indicate that our method performs well, especially during the COVID-19 crisis.

2 Generalized Regression Neural Network (GRNN)

A generalized regression neural network, denoted GRNN, is a variation of a radial basis neural network, first proposed by Specht et al. (1991). GRNN is a nonparametric model which can be applied to classification or (time-series) regressions. This model relies on the fact that a prediction for a given data point x_i can be obtained as a weighted average of all previous values based on their proximity to x_i, 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 neural network (Ahmed et al., 2010). Like other neural networks, GRNN can be graphically represented using nodes and layers, see for example Sumiyati and Warsito (2020).

GRNN has several advantages over other algorithms. First, due to its similarity to the Nadaraya-Watson estimator, GRNN is non-parametric. No additional assumptions regarding the functional form are needed. Unlike the ARIMA approach, which is a global approximator – a chosen relationship is extrapolated to all

observations – GRNN is a local estimator (Gheyas and Smith, 2009). Moreover, to define GRNN using cross-sectional data, only one parameter is to be defined, namely a smoothing parameter σ. This parameter is important because it controls the length of the smoothing region.

More formally, suppose both a training set consisting of n training patterns $\{x_1, x_2, \ldots, x_n\}$ 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:

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

The final estimation 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 a future behaviour of a given time series. To be more specific, it is possible to form a training set consisting of k-tuples of historical values. The corresponding training target is the next observation. Then, the last k observations are used as an input pattern. To illustrate this idea, Figure 1 depicts the dynamics of German GDP growth rate between Q1 1991 – Q3 2003.

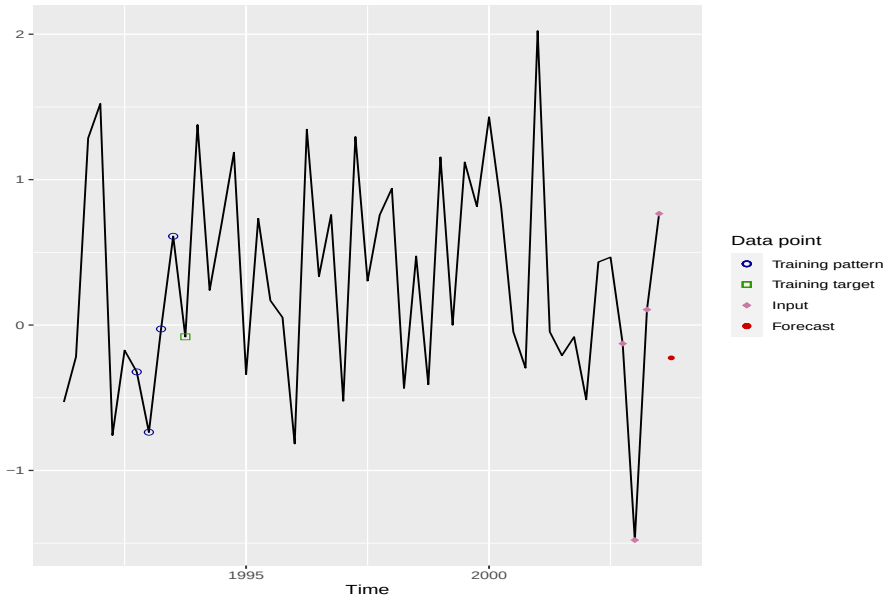


Figure 1: GRNN time series framework

In this example, we have 50 observations and $k = 4$, indicating that we have a training set containing 46 k-tuples: $S_k = \{(x_1, x_2, x_3, x_4), (x_2, x_3, x_4, x_5), ..., (x_{46}, x_{47}, x_{48}, x_{49})\}$. For example, let us take the first entity of S_k, 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. Meanwhile, an input pattern is the last k observations, i.e., $x = (x_{47}, x_{48}, x_{49}, x_{50})$. To produce a forecast Euclidian distance is calculated for each element in S_k and x and weights are assigned to every training target value according to (1) and a weighted average is taken as in (2). In Figure 1 one training pattern, its target value, the input target, and a one-step-ahead forecast are depicted.

This training pattern is chosen since it is the closest to the last observed values (input pattern) according to Euclidian distance. It implies that the highlighted target value gets the biggest weight w_i.

To produce a forecast we need to define two parameters k – the number of lags and σ. A handful of methods to determine both k and σ have been proposed, including K-folded cross-validation (Yan, 2012). Concerning this selection, we rely on Martínez et al. (2022), who suggested a relatively fast algorithm to produce highly accurate forecasts. The number of lags can be determined using a heuristic approach:

1. If a time series is seasonal, then $k = s$, where s is a length of the seasonal period.
2. If a time series is not seasonal, $k = p$, where p is a number of significant lags of the partial autocorrelation function (PACF).
3. If both conditions are not met, $k = 5$.

σ is estimated by exploiting an optimization tool, that minimizes a 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. For more details, see Martínez et al. (2022).

However, some modifications should be implemented, in order to predict future values of non-stationary or trend-stationary time series. Eventually, GRNN time series forecasting is based on detecting some patterns in the past to produce a final estimation. It means that this model is unable to predict an observation which is out of range of previous observations. To capture a trend-stationarity, a following additive data transformation is proposed:

1. Both a training pattern vector and the corresponding target value(s) are modified by subtracting the mean of the former. Let us take the previous example, illustrated in Figure 1. Each observation in $S_1 = (x_1, x_2, x_3, x_4)$ is demeaned by $\bar{x}_1 = \frac{\sum_{i=1}^{k=4} x_i}{4}$. Thus, we 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 $\tilde{x}_5 = x_5 - \bar{x}_1$. In order to compute the Euclidian distance the input pattern is also demeaned, i.e. $\tilde{x} = (x_{47} - \bar{x}_1, x_{48} - \bar{x}_1, x_{49} - \bar{x}_1, x_{50} - \bar{x}_1)$. It assures that the distance is unaffected by the transformation. This procedure applies to each entity in S_k, each corresponding target value, and the input pattern.
2. Obtain a prediction using modified data. In our case, a prediction is a weighted sum of transformed target values, namely $\hat{\tilde{x}}_{t+1} = \sum_{t=5}^{50} \tilde{x}_t$
3. To get the actual forecast, the mean of the input vector is added. In other words, the final prediction is $\hat{x}_{t+1} = \sum_{t=5}^{50} \tilde{x}_t + \bar{x}$, where $\bar{x} = \frac{\sum_{i=47}^{k=50} x_i}{4}$ is the mean of input vector x.

However, various economic time series are characterized by a non-linear (exponential) trend. The same logic as in the previous case may be extended to capture an additive trend. To be more specific, instead of demeaning, it is possible to divide by an aforementioned mean of each k-tuple in S_k. To retrieve a final forecast, the transformed weighted average is multiplied – rather than added – by the mean value of input vector x.

Finally, while forecasting, we use an "iterative" approach throughout the whole study. It implies if one wants to forecast h-step-ahead, where $h \geq 2$, first, a one-step-ahead forecast is carried out. Then, the forecast is treated as if it is an observation to produce another one-step-ahead forecast. In other words, in order to produce a h-step-ahead forecast, a one-step-ahead is iteratively repeated h times.

3 Monte Carlo Simulations

In this section, we study the forecast precision of GRNN time series predictions depending on different generating processes (DGPs) using Monte Carlo simulations. Without loss of generality, we assume that each simulated y_t is a monthly time series, and we are aiming to predict the next 12 months. AR(1) is selected as a benchmark throughout Monte Carlo investigation. In other words, we compare GRNN forecast precision to its AR(1) counterpart when true DGP is unknown. AR(1) has been extensively used in the literature as a benchmark (Faust and Wright, 2013; Franses, 2020). Last but not least, as demonstrated earlier, additive and multiplicative data transformations can be applied to capture various time series trends. In our simulations, we choose GRNN with no transformation and with additive transformation.

3.1 Forecasting Algorithm

We consider N samples of n observations of a target series y_t drawn from pre-selected distributions. Each sample is divided up into two separate groups of observations, denoted I and P, such that $n = I + P$. $I = n - 12$ in-sample values are exploited to estimate each model, whereas $P = 12$ pseudo-out-of-sample observations are later used to evaluate forecast precision. Forecasting performance of the models is evaluated by using a set of two measures, i.e., Mean Absolute Forecast Error ($MAFE$) and Root Mean Squared Forecast Error ($RMSFE$).

$MAFE$ for forecast horizon h can be expressed as follows:

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

where h is a forecast horizon, $\hat{Y}_{t+h|t}$ is a prediction of Y_{t+h} based on available information at time t, Y_{t+h} is an actual value at time $t + h$, N is a number of samples.

Whereas $RMSFE$ for forecast horizon h can be defined as:

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

where h is a forecast horizon, $\hat{Y}_{t+h|t}$ is a prediction of Y_{t+h} based on available information at time t, Y_{t+h} is an actual value at $t + h$, N is a number of samples.

Additionally, we define relative $MAFE$ and relative $RMSFE$, later denoted $rMAFE$ and $rRMSFE$, which help us to assess to what extent GRNN is superior (otherwise inferior) to a benchmark:

$$\begin{aligned} rMAFE_h &= \frac{\sum_{i=1}^{N} |Y_{t+h} - \hat{Y}_{t+h|t}^{GRNN}|}{\sum_{i=1}^{N} |Y_{t+h} - \hat{Y}_{t+h|t}^{AR(1)}|} = \frac{MAFE_h^{GRNN}}{MAFE_h^{AR(1)}} \\ rRMSFE_h &= \frac{\sqrt{\sum_{i=1}^{N} (Y_{t+h} - \hat{Y}_{t+h|t}^{GRNN})^2}}{\sqrt{\sum_{i=1}^{N} (Y_{t+h} - \hat{Y}_{t+h|t}^{AR(1)})^2}} = \frac{RMSFE_h^{GRNN}}{RMSFE_h^{AR(1)}}, \end{aligned} \tag{5}$$

We choose $N = 10000$. Concerning the number of observations, GRNN is supposed to perform better once an ample amount of observations are available. Meanwhile, macroeconomic data is usually limited, which is why to argue that GRNN can be applied to real-world economic problems, one should also check how this model performs on small datasets. Taking both aforementioned facts into consideration, in our work $n \in S_n = \{62, 112, 212, 412, 1012\}$. Recall, $n = I + P$, where $P = 12$, the number of observations used to estimate models $I \in S_I = \{50, 100, 200, 400, 1000\}$.[1]

[1] It is assumed that to conduct a time series analysis at least 50 observations are required (Warner, 1998).

Finally, different DGPs are proposed: $AR(1), AR(4), ARMA(3,3), VAR(2), TAR(2)$. Not only may these DGPs help us to evaluate a relative GRNN performance per se, but also cover different challenges arising from parametric time-series analysis. We start from the simplest $AR(1)$ model. In order to analyze the lag misspecification, $AR(4)$ is added to Monte Carlo exercise. Additionally, $ARMA(3,3)$ is included to study the importance of infinite moving averages and infinite autoregressive representations. $VAR(2)$ can be used to observe how missing variable(s) influence the relative forecasting performance. Finally, with the help of TAR, we introduce non-linearity.

The last step is to select specific parameters for each model (coefficients and distribution of error terms). There exist no prolific procedures to optimally select such parameters because this choice depends on the specific research question. One possible solution is presented in Krone et al. (2017). They choose a range of autocorrelation values, ϕ such that $\phi \in [-0.9; 0.9]$, with steps of 0.1. However, in the paper, researchers examine the difference between various $AR(1)$ estimators applied to short time series. Although it may seem to be a good empirical strategy, we primarily focus on illustrating the usefulness of GRNN under different settings rather than deeply investigating how $rMAFE$ and $rRMSFE$ depend on specific parameters. We will explain the choice of parameters in the corresponding subsections.

To sum it up, we suggest the following algorithm:

1. Simulate $N = 10000$ samples of $n \in S_n = \{62, 112, 212, 412, 1012\}$ observations using different DGPs $\in S_{DGP} = \{AR(1), AR(4), ARMA(3,3), VAR(2), TAR(2)\}$.
2. Use $I = n - 12$ observations to estimate pre-selected models, namely AR(1), GRNN with additive transformation, GRNN with no transformation;
3. The last $P = 12$ observations are used to conduct 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 find out that the $rRMAFE$ and $rRMSFE$ yield almost identical results. That is why, without loss of generality, we report and analyze only $rRMSFE$ values. At this stage, the forecasting performance within each specification is analyzed with respect to n and h. Besides, GRNN estimations with no transforamtion are presented, since in almost every occasion, additive data transformation is redundant.

However, we would also like to compare DGPs with each other. Given a DGP and corresponding samples, the $rRMSFE$ behaviour is unique. Subsequently, to draw a more general conclusion, we plot $rRMSFE$ for each DGP for visual comparison. A potential trade-off between the computational time and forecasting precision is also discussed. As an additional comparison between GRNN and parametric approaches, we study how precisely the existing approach of fitting an $ARMA$ model based on information criteria performs on the simulated data.

3.2.1 AR(1)

The first DGP selected for Monte Carlo simulation is AR(1). Even though this specification seems to be straightforward, the obtained forecasts are typically rather 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, 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, higher absolute values of AR(1) indicate the 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 $\alpha \in S_\alpha = \{0.9, 0.5, 0.2\}$. The error terms in every simulation are iid with $\epsilon_t \sim N(0, 1)$.

There are different methods to estimate AR(1), among others Ordinary Least Squares (OLS), Maximum Likelihood (ML) estimation, Conditional Sum of Squares (CSS), Yule-Walker Equations based on the method of moments, etc. In this study, we estimate the time series using CSS. The simulation results yield that for any given number of observations, n, as well as for any time horizon, h, AR(1) is superior to GRNN in terms of forecast precision. It may seem obvious because once we know true DGP, its forecasting performance should be better than any other model, including GRNN.[2] However, our primary goal is to discover whether there are some noticeable patterns in $rMSFE/rMAF$E behavior.

Case 1

In Figure 2, $rRMSFE$ for $AR(1)$ with $\alpha = 0.9$ is presented. Under this scenario, $AR(1)$ with high persistence may be treated as a periphery between a stationary autoregressive process and a random walk when $\alpha = 1$ is assumed. In general, a slow decay indicating an inverse proportionality between the forecast horizon and $rRMSFE$ is observed. For short-term forecasting we notice that the number of observations plays a more significant role, i.e., the more data is available, the lower $rRMSFE$. However, this difference is inclined to disappear for further horizons. Recall 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 hard to postulate since they depend on different assumptions, so-called mixing conditions, which are hard to test in practice (Heiler, 1999). Even though asymptotic properties are cumbersome we are prone to study short-term forecast quality empirically.

In Figure 3, $rRMSFE$ for $AR(1)$ with $\alpha = 0.9$ with additive data transformation is presented. As argued previously, this transformation allows for capturing a linear trend. Although this AR(1) is borderline stationary, we find it beneficial to apply data modification. For short-term horizons, GRNN with this transformation outperforms its counterpart.[3] However, the $rRMSFE$ linearly increasing with respect to h unlike the previous setting. Moreover, the number of observations n plays an important role as well. Namely, the fewer data you observe, the steeper the angle of $rRMSFE$. In other words, when a high persistence of α takes place, the necessity of applying data transformation depends on h.

Case 2

[2]Because of this fact, we do not compare GRNN with true DGP for the rest of the simulations. Nevertheless, we address a potential issue of parametric forecast.

[3]This is the only DGP used in Monte Carlo simulations, for which this fact holds.

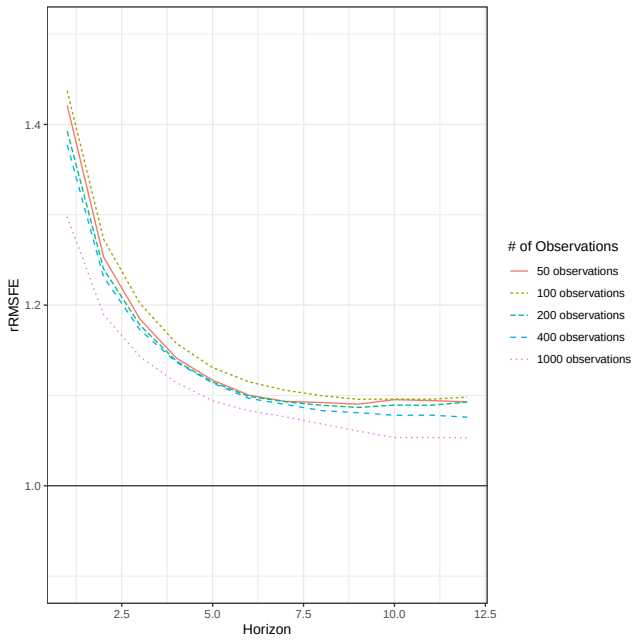


Figure 2: $rRMSFE$ without transformation

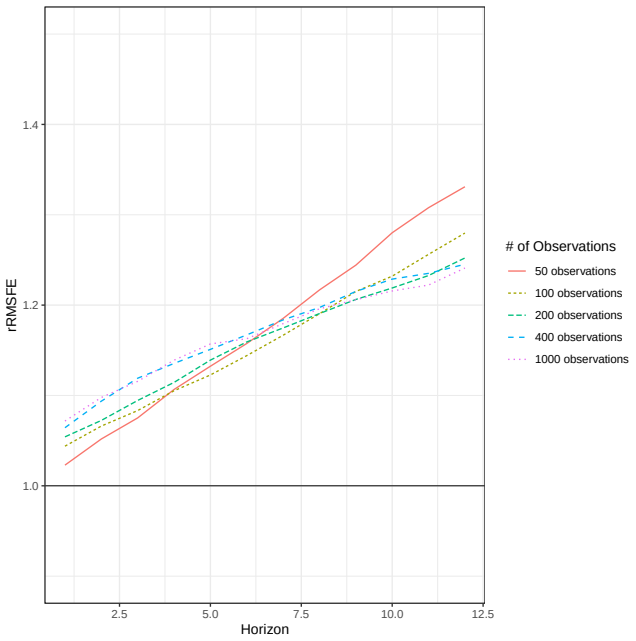


Figure 3: $rRMSFE$ with additive transformation

In Figure 4, rRMSFE for AR(1) with $\alpha = 0.5$ is shown. For this case, the values of $rRMSFE$ are lower compared to a previous instance. Thus, on average, GRNN performs better once the autocorrelation value is further from one. The similar slow decay of $rRMSFE$ is observed as in Figure 2. Furthermore, data transformation is already redundant – $rRMSFE$ of GRNN with no data modification is lower for every h compared to the same model with transformation. As for the number of observations, it is clear that almost all $rRMSFE$ curves are parallel to each other. This fact signals that the forecast quality indeed increases with respect to n.

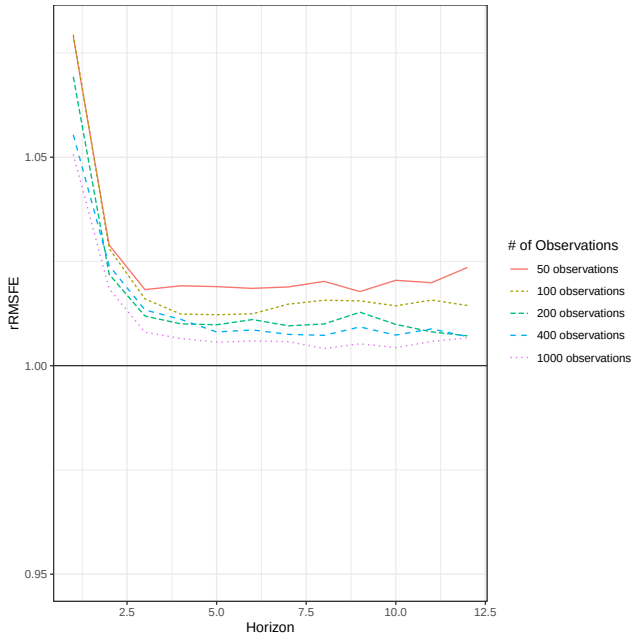


Figure 4: $rRMSFE$ for AR(1), $\alpha = 0.5$

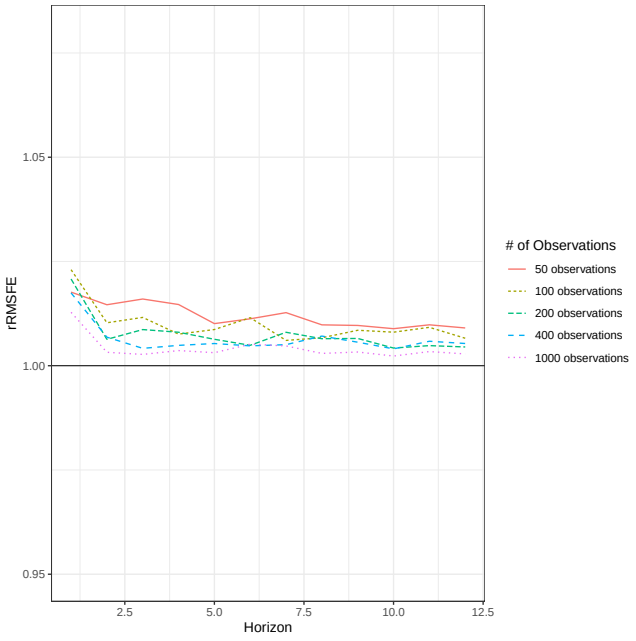


Figure 5: $rRMSFE$ for AR(1), $\alpha = 0.2$

Case 3

The same statements hold for the case when $\alpha = 0.2$, as depicted in Figure 5. Moreover, regardless of values of n, GRNN performs almost as accurately as true DGP. It signals that when the persistence of AR(1) is rather low, the forecast qualities of GRNN and AR(1) are comparable while the role of n becomes less important.

3.2.2 AR(4)

One possible way to naturally extend AR(1) model is to increase the number of lags, thus shifting towards AR(p), where $p > 1$. With the help of AR(p), we aim to study the potential lag misspecification, i.e., what would have happened if we falsely assumed the correct amount of significant lags. As a baseline specification, we select AR(4) for a specific reason. Various macroeconomic indicators, such as GDP and inflation rate, are published quarterly. This fact makes researchers treat AR(4) as a useful initial model for their analysis of such time series if they desire to start with non-periodic models (Rudd and Whelan, 2007).

Since we focus on stationary univariate time series, we are to define AR(4), which satisfies the stationarity condition – the roots of the AR(p) polynomial must be outside the unit circle. For our further analysis, we select two separate cases.

Case 1

First, we have a look at the following equation:

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

where ϵ_t is white noise.

Not only does this specification allow us to examine a lag misspecification issue per se, but also to consider the importance of further observations while producing a forecast. As we can see, relative weights of y_{t-3} and y_{t-4} are indeed higher than those of y_{t-1} and y_{t-2}. We choose these parameters on purpose. Supposedly AR(1) is assumed to fail to detect this relationship, whereas more flexible GRNN should adjust its weights accordingly. Imagine, however, a different scenario, when the AR(1) coefficient is dominant, forcing the whole DGP to mimic AR(1). Under these settings, the Monte Carlo results may yield indistinguishable estimations from the previous case, thus making it nearly impossible to inspect the omitted lag problem properly. This case is observed in case 2.

In Figure 6, rRMSFE for a baseline AR(4) is illustrated. For any given time horizon except one occasion ($n = 50$, $h = 10$), GRNN outperforms AR(1) as expected. The nonparametric approach can capture more complicated DGP, regardless of the initial amount of observations. On average, for one-step-ahead forecasting, rRMSFE is less than 0.8, indicating that GRNN predicts 20% more accurately. What can be evident is that similarly to AR(1), rRMSFE tends to one, implying that for a long-term prediction, both AR(1) and GRNN have the same forecasting power. Again, it deals with the fact that AR(4) forecast tends to its mean, which by construction equals zero, as in AR(1) case.

The speed of such convergence differs based on the sample size. For almost all forecasts, one can observe that the more observations are available, the more precise a relative forecast performance. However, even if the number of observations is rather low (50 or 100), it does not prevent GRNN from performing significantly better than a wrong parametric time series model. In a sense, GRNN is able to at least partially cope with a low signal-to-noise ratio challenge.

Case 2

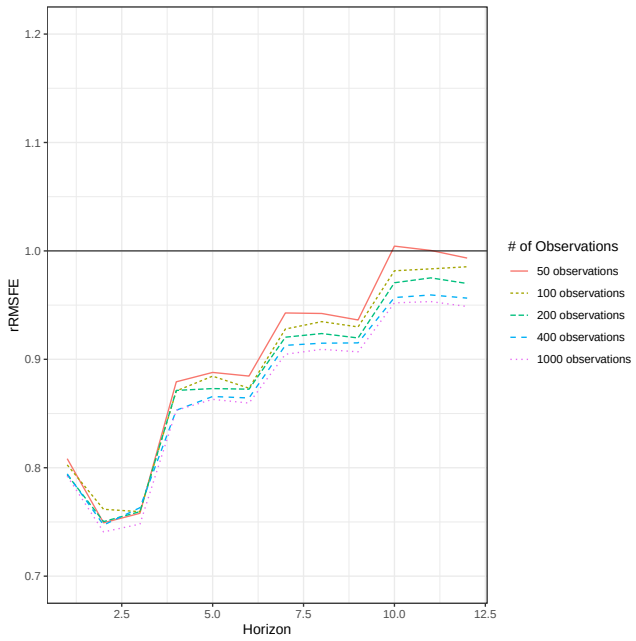


Figure 6: $rRMSFE$, case 1

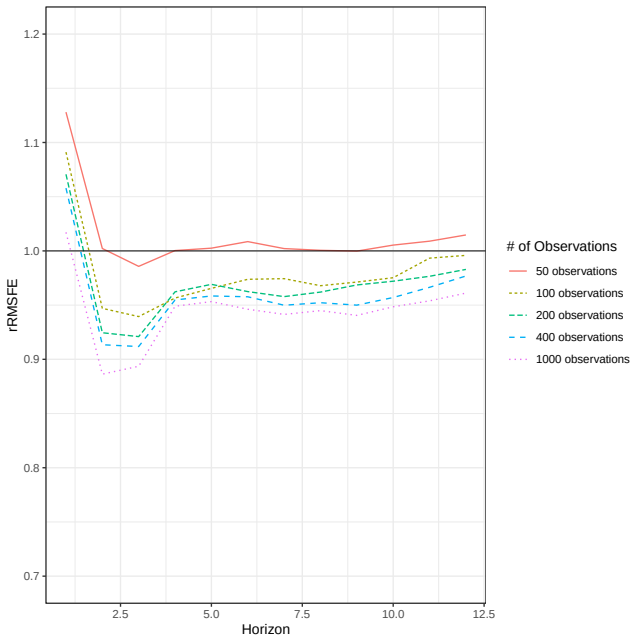


Figure 7: $rRMSFE$, case 2

Alternatively, we inspect the case when relative weights of y_{t-3} and y_{t-4} are significantly lower than those of y_{t-1} and y_{t-2}. To be more concrete, let us define the following model:

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

where ϵ_t is white noise.

In Figure 7, based on DGP, defined in (8) is shown. The behavior of $rRMSFE$ is notably different from the previous case. For $h = 1$, AR(1) outperforms GRNN given any amount of observations. However, once $h \geq 2$, we notice that $rRMSFE$ falls below one again, except when $n = 50$. So, 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. There is a sudden jump between $rRMSFE$ when $n = 50$ and the rest of the curves. In other words, the more data is observed, the more likely that GRNN can capture AR(4) process with a dominating AR(1) component.

3.2.3 ARMA(3,3)

Now, we extend AR(p) by adding moving average components, thus transforming the latter to $ARMA(p, q)$ model. This model became popular after Box and Jenkins had proposed an efficient way to find optimal values for both 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 (3 or less). Taking this into consideration, we find it fruitful to examine an extreme case, i.e., ARMA(3,3).

However, this model is of particular interest also for different reasons. To begin with, ARMA(3,3) – even though stationary upon selecting parameters – is highly volatile. A handful of significant autoregressive terms alongside moving average components make it cumbersome to estimate and conduct a prolific forecast analysis. Besides, this exact specification has been applied in academic literature, for example, to examine future numbers of the monthly active Facebook and Twitter worldwide users (Al-Haija et al., 2019). Due to a recent interest in

using unstructured data to produce macroeconomic forecasts, inspecting ARMA(3,3) may shed some light on underlying processes behind social network activity.

As a benchmark, the following ARMA(3,3) is presented:

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

where ϵ_t is a white noise.

In Figure 8, $rRMSFE$ for a baseline ARMA(3,3) is illustrated. GRNN outshines AR(1) as expected, yielding lower $rRMSFE$ values for almost all n and h. As in case 2 with AR(4) DGP, when $n = 50$ quality of forecasts is lower. Still, even with small amount of observations, GRNN still produces better predictions compared to the benchmark. Based on Monte Carlo simulation results, one may expect that GRNN copes notably better than $AR(1)$ when a more sophisticated $ARMA(p, q)$ is introduced.

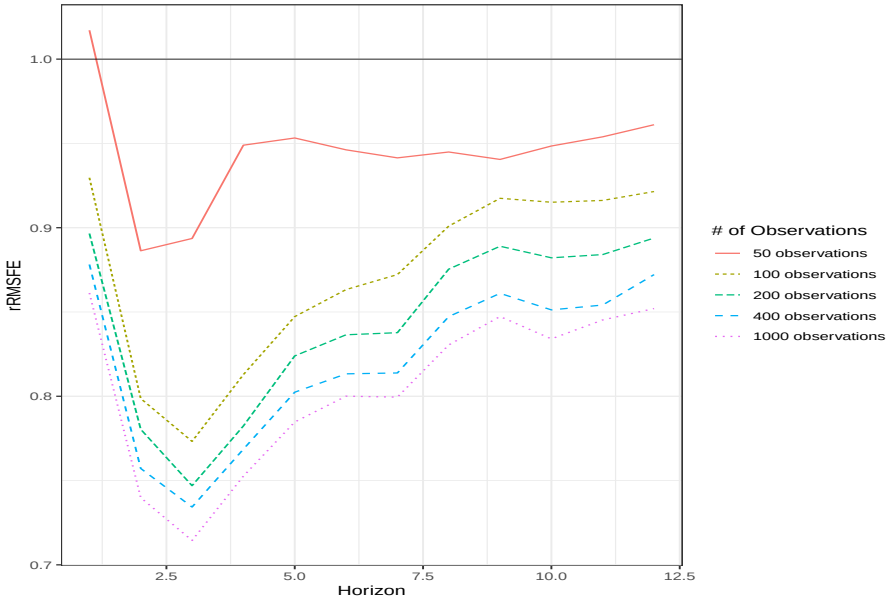


Figure 8: $rRMSFE$ for ARMA(3,3)

3.2.4 VAR(2)

Vector Autoregressive Models of order p, denoted $VAR(p)$ are extensively used in macroeconomic forecasting. Eventually, $VAR(p)$ 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 specification of $VAR(2)$:

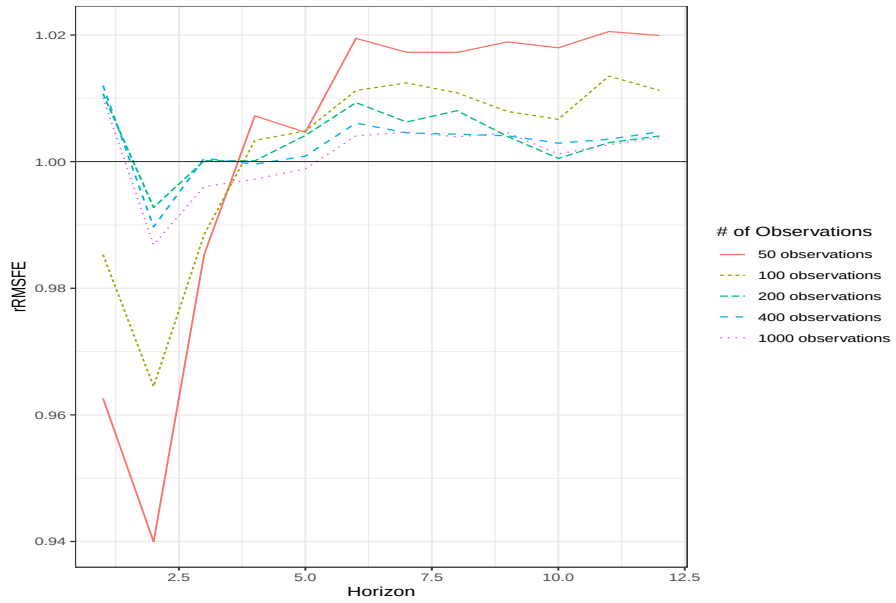
$$\begin{aligned} y_{1,t} &= 0.2y_{1,t-1} + 0.1y_{2,t-1} - 0.4y_{1,t-2} + 0.3y_{2,t-2} + \epsilon_{1,t} \\ y_{2,t} &= -0.3y_{1,t-1} + 0.4y_{2,t-1} + 0.2y_{1,t-2} - 0.3y_{2,t-2} + \epsilon_{2,t}, \end{aligned} \tag{10}$$

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

In Figure 9, $rRMSFE$ for $VAR(2)$ is presented. GRNN is superior to AR(1) for short-term forecasting when true DGP is $VAR(2)$. Surprisingly, when the number of observations $n = 50$, $rRMSFE$ for $h \leq 2$ is notably below all the other curves. It indicates that GRNN even with a relatively small sample size is able to cope with the missing variable(s) challenge significantly better than AR(1). We suspect that if true DGP is

either $VAR(p)$, where $p \geq 3$ or VAR(2) with more corresponding equations, the relative forecasting performance would increase even further.

Figure 9: $rRMSFE$ for VAR(2)



3.2.5 TAR(2)

The threshold autoregressive model (TAR) is another extension of simple AR(p). In the TAR framework, a regime-switching behavior is introduced, that 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 forces TAR to be nonlinear but staying locally linear. A considerable amount of articles both in economics and econometrics have been written to exploit the TAR model (Hansen, 2011).

The simplest class of TAR models is Self Exciting Threshold Autoregressive of order p and q, denoted $SETAR(p, q)$, where p is the number of thresholds, q is the order of the autoregressive part.[4] SETAR implies that a switching variable depends on lagged values of the dependent variable.

For Monte Carlo simulations, we choose a SETAR specification, closely related to this, proposed in Li and Tong (2016). To be more specific, SETAR(2,2) is as follows:

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

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

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

[4]Since q may vary across regimes, a model can sometimes be denoted as SETAR(p).

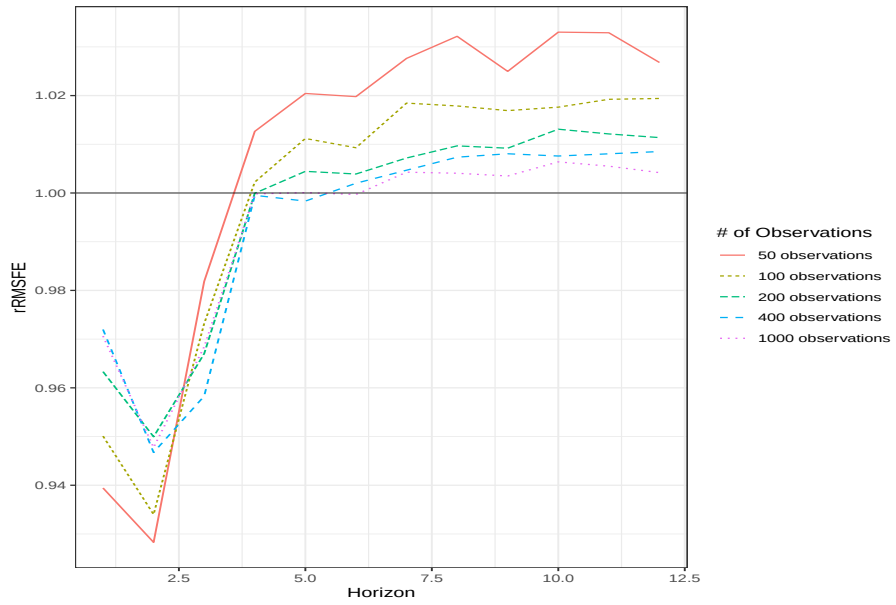


Figure 10: rRMSFE for TAR model

3.3 General remarks

In this subsection, some broader results are addressed. First, even though we discuss each case separately, it is necessary to compare the forecasting performance across different DGPs. In Figure 11, the values of $rRMSFE$ for pre-selected DGPs with fixed $n = 200$ are presented.

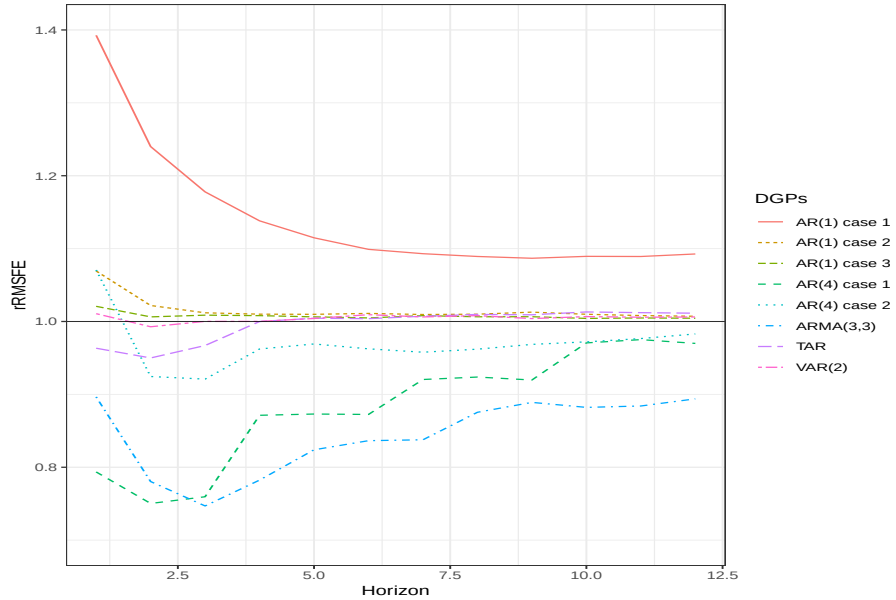


Figure 11: Simulation results for n=200

Almost all $rRMSFE$ curves are either indistinguishable from one or fall below this threshold, as argued previously. In other words, this plot indicates that when $n = 200$, on average GRNN, produces better or at least comparable predictions to AR(1) for every forecasting horizon h. This observation is important per se since it signals that GRNN – a less restrictive algorithm – may serve as a benchmark instead of AR(1) in many empirical studies. The GRNN's ability to capture non-stationarity unlike a more traditional $AR(1)$ strengthens

this intuition.

It is worth noting, NN usually demands additional costs to estimate a model, causing a trade-off between computational time and forecast precision. For example, the NN specification used in Richardson et al. (2018) takes approximately 31 minutes to produce a corresponding result. However, time-series GRNN is significantly faster. On average, the time taken to fit and predict with GRNN ranges from 0.4–0.6 seconds. For comparison, implementing the same procedure with ARMA takes 0.12–0.3 seconds. In other words, a new approach of finding σ and the number of lags allows us to estimate time series GRNN almost as fast as classic parameter frameworks.

3.4 Parametric vs Nonparametric Approach

One potential drawback of a previously established comparison is as follows. If a true DGP is unknown and one carries out a parametric time series analysis; there are various techniques to fit ARMA model (among others, different information criteria and acf/pacf values). If the process is indeed 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.

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

Intentionally, we do not collate a non-parametric approach to a true DGP model. Once true DGP is available, there is no need to use any other even sophisticated approaches since the true model always yields better estimations. On the other hand, the underlying process could almost never be detected, thus a common approach is either to impose some assumptions/restrictions or to force data to fit a parametric model, relying on the selection procedure.

Since we simulated data from well-defined DGPs and we know the exact functional forms, it is possible to test empirically, how well generated data could be fitted to a parametric setting using existing frameworks. For our purposes, we use information criteria approach, i.e., the selection is based on Akaike Information Criteria (AIC) values. The same samples of $AR(1)$ and $AR(4)$, on which we perform our Monte Carlo simulations, are chosen. In Table 1, shares of correctly estimated models are presented.

DGPs	$n = 50$	$n = 100$	$n = 200$	$n = 400$	$n = 1000$
$AR(1)$,case 1	0.1561	0.1972	0.2263	0.2766	0.3564
$AR(1)$, case 2	0.4053	0.5038	0.4770	0.4959	0.5018
$AR(1)$, case 3	0.1094	0.1876	0.2295	0.2863	0.3792
$AR(4)$, case 1	0.3824	0.5364	0.4924	0.4903	0.4815

Table 1: Share of correctly estimated models

Given a sample of AR(1), we assume this model has been correctly estimated if an underlying data distribution has been found precisely. For this exercise, we do not take into account the estimation of α, which in many instances may significantly differ from the true value. Even though, for almost all DGPs, rRMSFE of

fitted ARMA model is close to GRNN estimation (sometimes even better), a parametric approach seems to be rather restrictive. Table 1 shows that parametric ARMA estimations often cannot define a true model. In a best scenario of AR(4), only a half of simulated samples is detected properly. However, real world data is barely distributed according to well-defined parametric models. This drives us to a conclusion, that GRNN – as a less restrictive model – may indeed compete with parametric frameworks while dealing with univariate time series.

4 Nowcasting German GDP

4.1 Empirical Strategy

In order to obtain GDP nowcast (both level value and growth rate) for a given quarter we propose an approach based on GRNN time series prediction. So far, we have demonstrated the potential of GRNN-based models to predict univariate time series. However, to estimate the current value of GDP one should take into account numerous different time series observed at higher frequencies than quarterly. This fact, in principle, makes researchers shift towards a multivariate setting. To circumvent the challenges of the multivariate case, we propose a specific transformation, enabling the use of a univariate GRNN model to nowcast GDP, exploiting monthly observations.

Contemporaneous values of some crucial macroeconomic variables are not observed within a quarter. For the case of Germany, the first official estimations of the overall economic development are available only when GDP figures are released, that is 30 days after the end of a quarter.[5] However, it is possible to evaluate unobserved GDP value using higher frequency variables (monthly, daily) which are observed and published within a quarter. In other words, given a set of all relevant available time series the projection of GDP may be computed (Giannone et al., 2008).

As a preliminary step of our analysis, actual values of monthly time series $y_{t,n}$, $n = 1, \ldots, N$, used for producing nowcast are aggregated to match a quarterly frequency of GDP. 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, in other words:

$$\tilde{y}^Q_{t,n} = \frac{y^Q_{t,n}}{GDP_t} \quad \forall y^Q_{t,n} \in S_{y_t} \tag{11}$$

We suppose it is possible to apply similar transformations, such as taking a difference/sum or reciprocal. We choose this specification for two main reasons. First, taking a ratio relates, in some sense, relates to a well-known log transformation. Additionally, the actual values of GDP, unlike some indicators, are far from zero. Therefore we do not need 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 (11). Each entity of the set depicts the dynamics of an observed time series relative to GDP. Though, the economic interpretation of $S_{\tilde{y}}$ entities could seem to be challenging due to unit differences (variables can be expressed in Euros, percentages, or no units). Therefore, one should treat $S_{\tilde{y}}$ as a manually derived set, which, as shown later, is of particular interest for predicting the current value of GDP.

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

Once we obtain $S_{\tilde{y}}$, we separately train each ratio $\tilde{y}^Q_{t,n} \in S_{\tilde{y}}$ on all available information using GRNN with modifications (additive and multiplicative) and without data transformations 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 end up with 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 $S_{\hat{\tilde{y}}} = \{\hat{\tilde{y}}^Q_{t,1}, \hat{\tilde{y}}^Q_{t,2}....\hat{\tilde{y}}^Q_{t,N}\}$. Notice, that a time index is t rather than $t+1$ since we produce nowcast, that is our principal target is a current value of GDP for a quarter t.

Finally, after predicting $\hat{\tilde{y}}^Q_{t,n} \in S_{\hat{\tilde{y}}}$ it is possible to retrieve GDP nowcasts by using the following relation:

$$\widehat{GDP}_{t,n} = \frac{y^Q_{t,n}}{\hat{\tilde{y}}^Q_{t,n}} \tag{12}$$

We obtain N different "bivariate" GDP nowcasts for a given quarter t. We collect 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 will use a median value of this set. Under this framework, median value seems to be more precise than an arguably more usual mean value because of the potential outliers. It is also possible to estimate trimmed mean – the value, one can estimate after eliminating a certain percentage of the dataset's smallest and greatest values. However, to produce our nowcast, we stick to the median because it allows to use all available data without any imposed restrictions.

Additionally, we propose a more subtle approach, later denoted as GRNN ADF. Rather than applying a specific GRNN model to the whole dataset, it may seem important to distinguish between different $\tilde{y}^Q_{t,n} \in S_{\tilde{y}}$. Supposedly, if a given $\tilde{y}^Q_{t,n}$ is stationary, then GRNN without data transformation on average shall perform better. Alternatively, if a linear trend is observed, GRNN with additive data modification is expected to prevail. Thus, the following algorithm based on the Augmented Dickey-Fuller test (ADF) is introduced:

- If we cannot reject the H_0 hypothesis – a unit root is present –, but can reject a combined H_0 – a unit root is presented and there is no linear trend – we use additive data transformation.
- If we reject both H_0 and combined H_0, we assume that no transformation is needed.
- If we fail to reject a combined H_0 regardless of initial H_0, we apply multiplicative transformation.

4.2 Model Selection

To compare the nowcasting performance the following model are selected. As a rather simple baseline model AR(1) for quarterly GDP is chosen as in Richardson et al. (2018). Also, a more simplified model averaging framework based on linear regression is estimated. To be more specific, this approach is similar to our proposed method. However, instead of using GRNN, we run N bivariate OLS regressions to obtain a similar set of GDP predictions. After collecting all the predictions, a simple mean is taken to produce a final forecast. This method is proven to be an advantageous tool to nowcast German economic activity (Claudio et al., 2020; Heinisch and Scheufele, 2018).[6] Additionally, we estimate DFM, in which the optimal number of factors to incorporate in the model is chosen according to Bai and Ng (2002), whereas the lag order of the factor-VAR selection is based on AIC.

[6] https://www.iwh-halle.de/en/research/data-and-analysis/macroeconomic-reports/iwh-flash-indicator/

4.3 Data

Our data set consists of 125 German macroeconomic indicators, and the sample covers the period Q1 1991 – Q2 2022. Prior to the analysis, time series possessing missing values are eliminated. Each time series is seasonally adjusted and either quarterly or monthly aggregated to match quarterly GDP frequency. Each indicator is left as it is (for GRNN-based approaches) or transformed to satisfy a stationarity property (for all the other models).

Our proposed method allows us to explicitly capture a ragged edge phenomenon by using only those time series for which we have at least one observation within a quarter t. This can be achieved by aggregating one or two available observations to a quarterly frequency. However, in our empirical study, we assume that all monthly observations for a given time series are known. Under these settings, each model is supposed to perform better due to more information available.

To compare the forecasting power of each model, we select the last 22 quarters (Q1 2017 – Q2 2022) and conduct a one-step-ahead out-of-sample forecast. For example, to produce a nowcast for Q2 2022, we use each time series including the observations for the current quarter. Based on this information, the current state of GDP is evaluated. The performance of each model is then calculated by comparing an estimation to the actual value of GDP using (3) and (4). This procedure is repeated for each pre-selected quarter. The periods chosen for the empirical analysis include the infamous COVID-19 crisis. It is fruitful to study the performance of the models under highly unstable conditions.

4.4 Transformed data for GRNN

GDP time series has a clear 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. It implies that for some $y^Q_{t,n}$ its counterpart $\tilde{y}^Q_{t,n}$ may be stationary if it mimics the GDP dynamics. Although, we expect the majority of $\tilde{y}^Q_{t,n}$ to be non-stationary. To highlight a potential difference in dynamics of $\tilde{y}^Q_{t,n}$, we discuss two crucial indicators, namely, the industrial production index (IP) and ifo Business Climate Index.

IP measures the value added of this economic sector and represents the value of output less the values of both intermediate consumption and consumption of fixed capital.[7] IP is an important indicator to assess the current state of economic activity (Eraslan and Götz, 2021). In Figure 12, the actual (red line) and transformed IP (black line) values are depicted.

[7] https://www.destatis.de/EN/Themes/Countries-Regions/International-Statistics/Glossary/IndustrialProductionGeneralIndexc.html

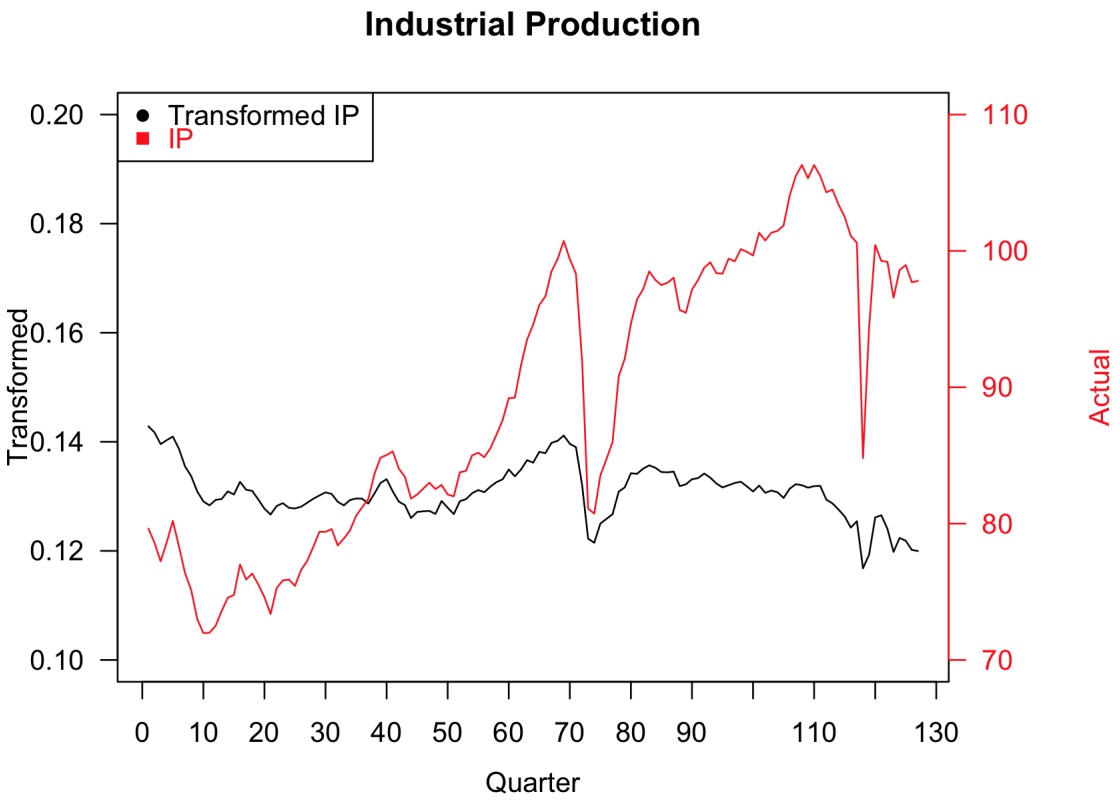


Figure 12: Actual and transformed IP values (Q1 1991 - Q2 2022)

Similar to GDP, IP has significantly increased over the years. We perform Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test to check whether the observable time series is trend-stationary (null hypothesis). The corresponding p-value is bigger than 0.1, meaning that we cannot reject that the time series is not trend-stationary. However, a modified time series upon dividing by GDP seems to be stationary. To test it we perform Augmented Dickey-Fuller (ADF) test with the null hypothesis indicating a non-stationarity. The corresponding p-value equals 0.062. Thus we reject the null hypothesis at the 10% level. In other words, it implies that the transformed IP series does not have a unit root.

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 both GDP and IP (Lehmann, 2022). In Figure 13, the actual (red line) and transformed (black line) values of Business Climate Indexes are depicted.

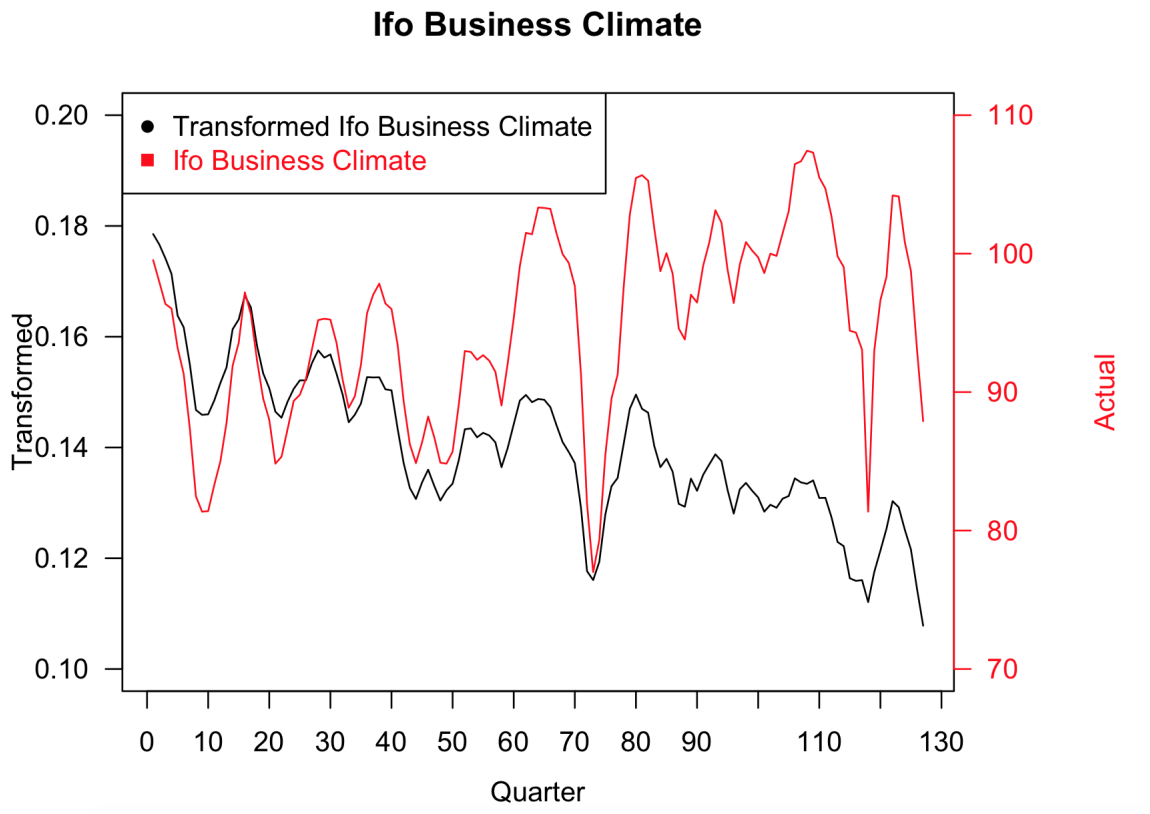


Figure 13: Actual and transformed ifo Index values (Q1 1991 - Q2 2022)

By construction, the actual values of the index, unlike IP, are bounded. That is why oscillations rather than a clear are observed. Correspondingly, this fact forces a modified time series to decay slowly over time, making the latter non-stationary. Nonetheless, the similarities in dynamics between the two lines can be noticed, indicating a positive correlation between GDP and the ifo Business Climate indicator.

4.5 Results

In this subsection, we describe the main empirical results of our analysis. Table 2 presents $MAFE$ and $RMSFE$, defined in (3) and (4), respectively, as well as $rMAFE$ and $rRMSFE$, as presented in (5).

Model	$MAFE$			$RMSFE$		
	$MAFE$	$rMAFE$	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 ADF	1.02	0.59	0.035	1.91	0.54	0.089
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 2: Nowcast performance (Q1 2017 - Q2 2022)

Each GRNN-based models outperform the rest of the models in terms of MAFE and RMSFE. On average, Generalized Regression Neural Network approaches are capable to reduce the forecasting error almost by 50% compared to AR(1) model. Both the model averaging approach and DFM also yield better estimations than the benchmark over the whole period. Additionally, we conducted the one-sided[8] Diebold-Mariano (DM) test with Harvey-Leybourne-Newbold correction for small samples to check whether two forecasts have the same accuracy. We need to use a corrected version of DM test since our sample size is not large enough to implement a classic version. Corresponding p-values are reported in Table 2. DM test results using absolute errors demonstrate that we can reject the null hypothesis when comparing each GRNN-based models and the Model averaging approach to AR(1) at 10%. Similar results are obtained by using a quadratic loss function instead of $MAFE$.

Now, we discuss, how models perform during the COVID-19 crisis. In Table 3 corresponding $rMAFE$ and $rRMSFE$ for Q1 2020 – Q1 2021 are shown.

Model	$MAFE$	$RMSFE$
	$rMAFE$	$rRMSFE$
GRNN additive	0.53	0.50
GRNN no transformation	0.30	0.29
GRNN multiplicative	0.58	0.53
GRNN ADF	0.59	0.53
Model averaging	0.69	0.71
DFM	0.82	0.84
AR(1)	1	1

Table 3: Nowcast performance (Q1 2020 - Q1 2021)

In general, for most models results are similar to Table 2, except for GRNN without data transformation.

[8] Alternative hypothesis is that the first forecast is more accurate than the second forecast, namely AR(1).

This model performs significantly better during Q1 2020 – Q1 2021, most accurately predicting the current state of economic activity. However, if we look at the quarters prior to the COVID-19 outburst, i.e., Q1 2017–Q4 2019, and after this period, namely Q2 2021 – Q2 2022 $rMAFE$ and $rRMSFE$ for GRNN with no transformation yields 1.2 and 1.3, respectively. Recall, that a classic GRNN model cannot capture any kinds of trend by construction. Therefore during the expansion, this specification is unable to produce reliable estimations, unlike, for example, GRNN with additive transformation.[9] On the other hand, GRNN with unmodified data may substantially perform better during the crisis, relying only on previously observed data. Surprisingly, there is no additional gain from using the GRNN model, based on the ADF test. This fact 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 – obtaining different "bivariate" nowcasts and taking a median of the set – time series differentiation based on non-stationarity is redundant.

To sum it up, an approach based on GRNN with additive transformation ceteris paribus shall be more reliable than its counterpart in the case of Germany. However, if one expects a rather drastic drop in economic activity, then GRNN without transformation can seemingly capture this phenomenon better.

5 Conclusion

In this article, we demonstrate the advantages of using a GRNN-based approach for macroeconomic nowcasting. First, we use a classic GRNN model in Monte Carlo investigation. We simulate different univariate DGPs and compare the forecasting performance of GRNN and AR(1). In many cases, the NN approach outperforms the benchmark. Additionally, we manage to show that existing ARMA fitting approaches – even though in many cases yielding closely related predictions – cannot properly identify a true DGP. 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 to use a univariate time-series GRNN model to nowcast the German GDP growth rate. First, the specific data transformation needs to be implemented, i.e., dividing aggregated level values of each indicator by the corresponding GDP value. Then, these ratios are used to perform one-step-ahead forecasting using GRNN. After that, using actual aggregated observations within a given quarter, a set of GDP nowcasts is obtained. 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.

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

References

Knut Are Aastveit, Karsten R Gerdrup, Anne Sofie Jore, and Leif Anders Thorsrud. Nowcasting GDP in real time: A density combination approach. *Journal of Business & Economic Statistics*, 32(1):48–68, 2014.

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

Qasem Abu Al-Haija, Qian Mao, and Kamal Al Nasr. Forecasting the number of monthly active Facebook and Twitter worldwide users using ARMA model. 2019.

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

Marta Bańbura and Gerhard Rünstler. 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, 2011.

Luca Barbaglia, Lorenzo Frattarolo, Luca Onorante, Filippo Maria Pericoli, Marco Ratto, and Luca Tiozzo Pezzoli. Testing big data in a big crisis: Nowcasting under COVID-19. *International Journal of Forecasting*, 2022.

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

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

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

Jon Faust and Jonathan H Wright. Forecasting inflation. In *Handbook of economic forecasting*, volume 2, pages 2–56. Elsevier, 2013.

Philip Hans Franses. IMA(1,1) as a new benchmark for forecast evaluation. *Applied Economics Letters*, 27(17): 1419–1423, 2020. doi: 10.1080/13504851.2019.1686115. URL `https://doi.org/10.1080/13504851.2019.1686115`.

Iffat A Gheyas and Leslie S Smith. A neural network approach to time series forecasting. In *Proceedings of the World Congress on Engineering*, volume 2, pages 1–3, 2009.

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

David Gibson and Darfiana Nur. Threshold autoregressive models in finance: a comparative approach. 2011.

Edward James Hannan and Manfred Deistler. *The statistical theory of linear systems*. SIAM, 2012.

Bruce E Hansen. Lecture notes on nonparametrics. *Lecture notes*, 2009.

Bruce E Hansen. Threshold autoregression in economics. *Statistics and its Interface*, 4(2):123–127, 2011.

Wolfgang Härdle. *Applied nonparametric regression*. Number 19. Cambridge university press, 1990.

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

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

Tanja Krone, Casper J Albers, and Marieke E Timmerman. A comparative simulation study of AR (1) estimators in short time series. *Quality & quantity*, 51(1):1–21, 2017.

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

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

Dong Li and Howell Tong. Nested sub-sample search algorithm for estimation of threshold models. *Statistica Sinica*, pages 1543–1554, 2016.

Helmut Lütkepohl. Vector autoregressive models. In *Handbook of research methods and applications in empirical macroeconomics*, pages 139–164. Edward Elgar Publishing, 2013.

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

Francisco Martínez, Francisco Charte, María Pilar Frías, and Ana María Martínez-Rodríguez. Strategies for time series forecasting with generalized regression neural networks. *Neurocomputing*, 491:509–521, 2022.

Adam Richardson, Thomas Mulder, et al. Nowcasting New Zealand GDP using machine learning algorithms. 2018.

Jeremy Rudd and Karl Whelan. Modeling inflation dynamics: A critical review of recent research. *Journal of Money, Credit and Banking*, 39:155–170, 2007.

Simon J Sheather. Density estimation. *Statistical science*, pages 588–597, 2004.

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

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

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

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

Rebecca M Warner. *Spectral analysis of time-series data*. Guilford Press, 1998.

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

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

Appendices

A Nadaraya-Watson Gaussian Kernel Regression Estimator

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

$$Y_i = g(X_i) + \epsilon_t, \tag{13}$$

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

Additionally, we assume that the error terms ϵ_t are iid 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):

$$g(x) = E(y|x) = \int \frac{yf(x,y)}{f(x)} dy, \tag{14}$$

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

Applying kernel density estimator – nonparametric 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 $g(\cdot)$:

$$\begin{aligned} \hat{g}(x) = \hat{E}(y|x) &= \frac{\sum_{i=1}^N Y_i K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})} = \sum_{i=1}^N \frac{K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})} Y_i = \sum_{i=1}^N W_i(x) Y_i \\ W_i(x) &= \frac{K(\frac{x-X_i}{h})}{\sum_{i=1}^N K(\frac{x-X_i}{h})}, \end{aligned} \tag{15}$$

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 $K(\cdot)$, 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
- $\int_{-\infty}^{\infty} u^2K(u)du \neq 0$ - second moment differs from 0

Even though plenty functions satisfy the above-mentioned properties, it is argued that a choice of a specific kernel should not significantly influence the precision of density estimation. Still, the most commonly used kernels in the empirical studies are Epanechnikov and Gaussian (Härdle, 1990; Hansen, 2009). In our paper, we select 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:

$$K_G(u) = \frac{1}{\sqrt{2}} \exp\left(-\frac{u^2}{2}\right) \tag{16}$$

Many scholars argued, Nadaraya-Watson kernel estimator heavily relies on a selection of a bandwidth h, which control 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 roughly be divided up 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).