IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 25, 2021, accepted December 12, 2021, date of publication January 4, 2022, date of current version January 11, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3140377

Time Series Forecasting by Generalized Regression Neural Networks Trained With Multiple Series

FRANCISCO MARTÍNEZ[1], MARÍA P. FRÍAS[2], MARÍA D. PÉREZ-GODOY[1], AND ANTONIO J. RIVERA[1]

[1]Department of Computer Science, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Jaén, 23071 Jaén, Spain
[2]Department of Statistics and Operations Research, University of Jaén, 23071 Jaén, Spain

Corresponding author: Francisco Martínez (fmartin@ujaen.es)

This work was supported by the Spanish Ministry of Science, Innovation and Universities under Project PID2019-107793GB-I00.

ABSTRACT Time series forecasting plays a key role in many fields such as business, energy or environment. Traditionally, statistical or machine learning models for time series forecasting are trained with the historical values of the series to be forecast. Unfortunately, some time series are too short to suitably train a model. Motivated by this fact, this paper explores the use of data available in a pool or collection of time series to train a model that predicts an individual series. Concretely, we train a generalized regression neural network with the examples drawn from the historical values of a pool of series and then use the model to forecast individual series. In this sense several approaches are proposed, including to draw the examples from a pool of series related to the series to be forecast or the training of several models with mutually exclusive series and the combination of their forecasts. Experimental results in terms of forecasting accuracy using generalized regression neural networks are promising. Furthermore, the proposed approaches allow to forecast series that are too short to build a traditional generalized regression neural network model.

INDEX TERMS Generalized regression neural networks, model combination, time series forecasting.

I. INTRODUCTION

Machine learning models require a suitable amount of data to be properly trained and this way produce good results; after all, they learn from examples. Therefore, it is no wonder that time series forecasting with machine learning excels in fields where data abound and sampling is done at high frequency levels, such as demand or energy forecasting [1], [2]. However, low frequency data, such as yearly time series, are also common in real-life situations. Unfortunately, these series are often too short to effectively train a machine learning model.

In the recent M4 forecasting competition [3] the contenders faced the herculean task of predicting 100,000 time series. The winner of the second prize [4] proposed a methodology that combines the forecasts of eight different models using a weighted average; the weights applied in the combination are determined by means of an XGBoost model trained with the whole data set of 100,000 series. That is, to forecast a time series eight models are trained with the historical values of the series; however, the weights used to combine their forecasts are computed using information extracted from the entire data set.

This idea of using information drawn from a pool of series to forecast an individual time series has already been used in other ways; for example, in [5] the seasonal indices of a short time series are estimated using a collection of time series of its own category. Another interesting approach is [6], in which in order to predict the future behavior of a time series, the most similar series—according to several similarity measures—from a set of rich and diverse reference series are found. The average of the future paths of these series are used to forecast the future values of the target series.

This paper proposes a new way of using information from a set of series to forecast an individual series. We are motivated by the fact that there exist time series that are too short to train a model. Our proposal is to forecast these series using a model trained with the historical values of other series, that is, the examples used to train the model are drawn from the

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

historical values of several series. This is a great contrast to current forecasting methodologies in which the model used to forecast a time series is trained exclusively with the historical values of the series being predicted. The model applied in our experimentation is a generalized regression neural network, a kind of neural network that has proved its usefulness in forecasting time series [7], [8].

The rest of this paper is structured as follows. Section II explains how to forecast time series using generalized regression neural networks and the modeling decisions made to apply this kind of neural network in our experimentation. Section III describes our proposal for training a generalized regression neural network using examples extracted from a pool of time series. In Section IV some experimentation is done to assess the forecast accuracy of the proposed methodology and some variations of the initial proposal. Finally, Section V analyzes the results of the experimentation and Section VI draws some conclusions.

II. TIME SERIES FORECASTING WITH GENERALIZED REGRESSION NEURAL NETWORKS

A generalized regression neural network (GRNN) is a variation of a radial basis neural network proposed by Specht [9]. If enough samples are available, a GRNN can approximate any continuous function to an arbitrary accuracy. Given a training set consisting of n examples: n training patterns (vectors $\{x_1, x_2, \ldots x_n\}$) and their corresponding training targets (scalars $\{y_1, y_2, \ldots y_n\}$), the output for an input pattern x is computed in two steps. First, the weights associated with the training patterns are calculated:

$$w_i = \frac{\exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right)}{\sum_{j=1}^{n} \exp\left(-\frac{\|x-x_j\|^2}{2\sigma^2}\right)} \tag{1}$$

the weights sum to one and represent the closeness of x to the training patterns, the closer the higher. Secondly, the training targets are averaged according to their weights to produce the output:

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

so a weighted average of the training targets is obtained, where the weights represent the closeness of the input to the training patterns. The role of the smoothing factor, σ, in (1) is to control the degree of smoothing. When σ is large all the targets have a small and similar weight, so the result is close to the mean of the targets. On the other hand, when σ is small only the targets whose patterns are close to the input have significant weights.

Fig. 1 shows the structure of a generalized regression neural network which is trained with n examples. As can be seen, a generalized regression neural network consists of three layers: the input, hidden and output layer. It should be noted that the only parameter of a generalized regression neural network is the smoothing factor.

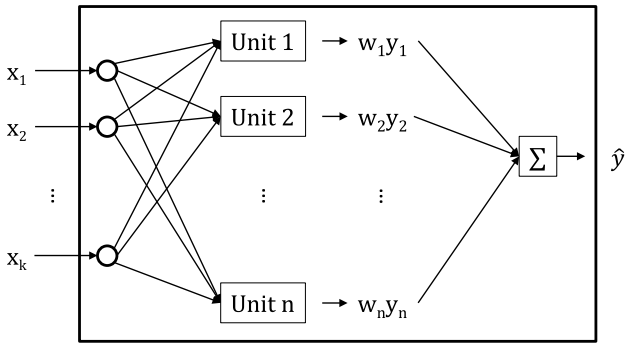


FIGURE 1. Structure of a generalized regression neural network.

TABLE 1. Training examples from series $\{y_1, y_2, \ldots, y_{40}\}$ when a training pattern is the four lagged values of its target.

Training pattern	Training target
y_1, y_2, y_3, y_4	y_5
y_2, y_3, y_4, y_5	y_6
y_3, y_4, y_5, y_6	y_7
...	...
$y_{36}, y_{37}, y_{38}, y_{39}$	y_{40}

To use GRNN in a time series forecasting context the training targets are historical values of the time series and the training patterns are lagged (i.e., previous) values of the targets. For example, given the time series $s = \{y_1, y_2, \ldots, y_{40}\}$ and assuming that a training pattern is formed by the four lagged values of its target, a subset of the 36 training examples that can be extracted from s are shown in Table 1. Fig. 2 shows an artificial quarterly time series with a strong seasonal behavior: in a year the levels of the first two quarters are similar and higher than the levels of the last two quarters, which are also similar. Again, it is assumed that a training pattern consists of the four lagged values of its target. Fig. 2 highlights the input to the GRNN, formed by the last four historical values of the series, its closest training pattern (i.e., the one with the highest weight) and its associated target. Also, this figure shows two different predictions for the next future value of the series (the first quarter of the next year). In one of the predictions the smoothing factor is small and only the targets of the training patterns very close (similar) to the input have significant weights in the weighted average (due to the seasonal behavior these targets will likely be first quarter values). In the other prediction the smoothing factor is large and, therefore, all the training targets have a similar weight, so the prediction is close to the mean of the historical values of the series. It should be noted that, owing to the seasonal pattern of the series, the prediction done with the smaller smoothing factor is clearly more suitable.

Generalized regression neural networks find patterns in a series similar to its last historical values, hoping that the subsequent values of these patterns are also similar to the future behavior of the series. The smoothing factor determines the

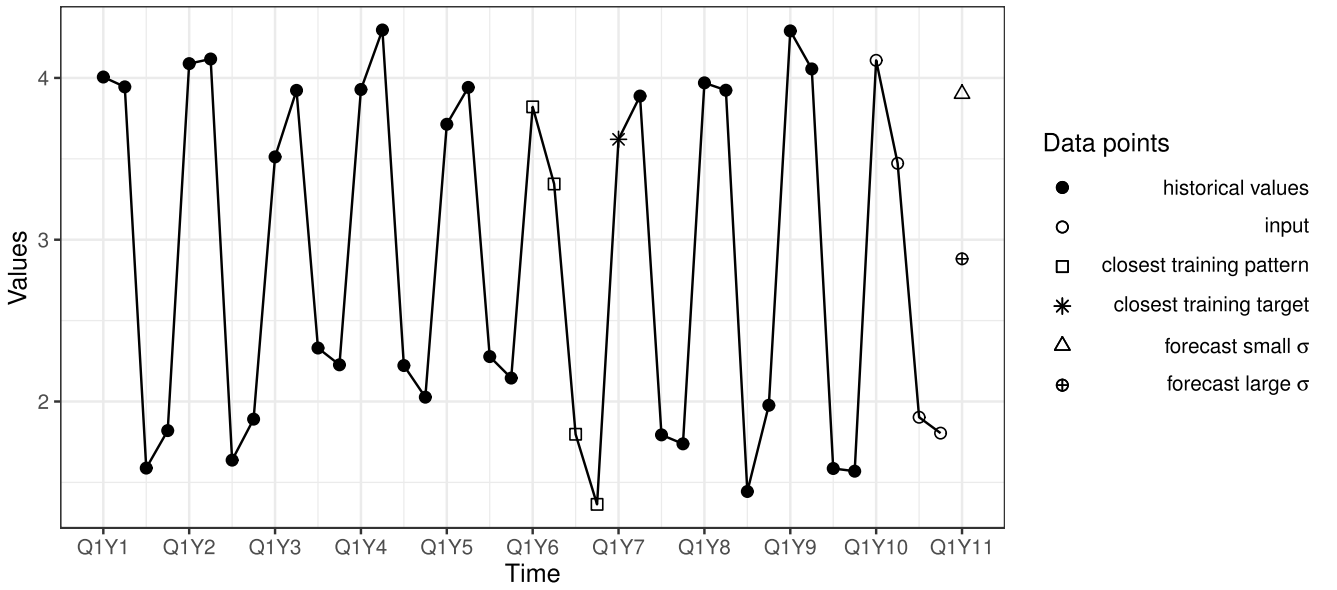


FIGURE 2. Forecasting with GRNN.

degree of similarity required to a pattern to play an important role in the prediction.

A. GENERALIZED REGRESSION NEURAL NETWORKS METHODOLOGY

This section explains the modeling decisions made in our experimentation to forecast times series by means of generalized regression neural networks. From the previous section, it should be clear that a correct choice of the smoothing factor is crucial for accurate forecasts. To this end, the last h historical values of a time series, where h is the forecasting horizon (i.e., the number of future values to be forecast), are used as a validation set to find a proper smoothing factor for the series. A model is fitted with the historical values previous to the validation set and an optimization tool is used to find the smoothing factor that minimizes a forecasting accuracy measure on the validation set. Once the smoothing factor is optimized, all the historical values can be used to train the model that predicts the future behavior of the series.

Other important decision is to choose the lagged values of the targets used as training patterns. Since our experimentation deals with short yearly time series, we have decided to only use the three lagged values of a target as its training pattern.

In our experimentation the next 6 years of yearly time series are predicted and therefore some strategy to forecast multi-step-ahead values has to be used [10]. We have chosen the iterative (also called recursive) strategy because it is efficient and produces good results [11]. With the iterative approach the forecasting model only predicts the one-step-ahead value of a series. Therefore, if the next h future values need to be predicted, the model is used h times in an iterative way. The inputs to the model are historical values of the series or previous predictions when historical values are not available.

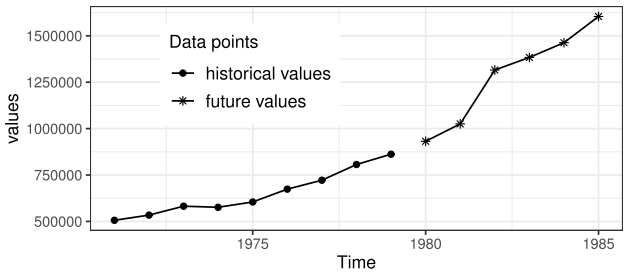


FIGURE 3. Historical values and values to be forecast for one of the shortest yearly series of the M1 competition.

III. OUR PROPOSAL

Our proposal is motivated by the difficulties we found in forecasting the yearly time series of the M1 competition [12] using generalized regression neural networks. This forecasting competition consists of, among others, 181 yearly series, with lengths ranging from 9 to 52 historical values. For all the yearly series their next 6 future values are to be predicted. Fig. 3 and Fig. 4 show the historical values and the future values to be predicted of the shortest and longest yearly series of the M1 competition respectively.

A time series with 9 historical values cannot be predicted with the GRNN methodology described above. Because the forecasting horizon is 6 we use as validation set, to choose a proper smoothing factor, the last 6 values of the series. Therefore, the training set, used to fit the model that assesses the forecast accuracy on the validation set, is formed by the remaining first 3 historical values of the series. However,

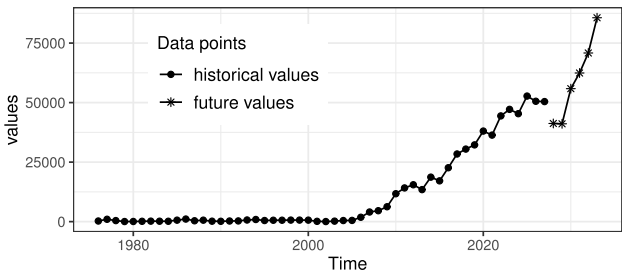


FIGURE 4. Historical values and values to be forecast for the longest yearly series of the M1 competition.

FIGURE 5. Training and validation sets for time series 3 when the pool of series is formed by 3 series with the same length.

with 3 values it is not possible to train a model, because no training example can be extracted: a training example needs 4 consecutive historical values of the series (a training pattern of 3 consecutive values and a target with the next value).

Taking into account the aforementioned problems for training GRNN models with short series, we have developed the following strategy to forecast the yearly series of the M1 competition using GRNN:

1) As mentioned previously, an example for training a model consists of a target (a historical value of the series) and a training pattern formed by the three lagged values of the target.
2) Also, as discussed above, the iterative approach is used to forecast multi-step-ahead values.
3) Given a time series, the validation set applied to select the smoothing factor is formed by its last 6 values. The GRNN model used to forecast the validation set is trained with the historical values of the series previous to the validation set and with the historical values of the other 180 yearly time series of the M1 competition. As an example, Fig. 5 shows the training and validation set for a time series trained with a pool of three time series.
4) Once the smoothing factor is chosen, the training examples used by the GRNN model to forecast the time series are drawn from the values of all the 181 yearly time series of the M1 competition.
5) The inputs to the GRNN model are extracted from the series being forecast.

The main novelty of our strategy is that a GRNN model is trained not only with the historical values of the series to be forecast, but also with the historical values of a collection of series. For example, if a GRNN model is trained with the following two series: $s1 = \{y_1, y_2, y_3, y_4, y_5\}$ and $s2 = \{x_1, x_2, x_3, x_4, x_5, x_6\}$, then its training examples are the ones shown in Table 2. These training examples would be used to forecast both $s1$ and $s2$.

TABLE 2. Training examples from series $\{y_1, y_2, y_3, y_4, y_5\}$ and $\{x_1, x_2, x_3, x_4, x_5, x_6\}$ when a training pattern is the three lagged values of its target.

Training pattern	Training target
y_1, y_2, y_3	y_4
y_2, y_3, y_4	y_5
x_1, x_2, x_3	x_4
x_2, x_3, x_4	x_5
x_3, x_4, x_5	x_6

The weights used by GRNN to average the training targets are related to the closeness (similarity) of the input to the training patterns, where the closeness is based on (1). Because our proposal uses training examples from different time series with different magnitudes it is necessary to scale the training examples and the input so that they share the same scale and the weights are not affected by the different magnitudes of the series. To this end, our proposal transforms the training examples and the inputs as described next.

Given a training example, consisting of a training pattern (a vector p of length 3) and a training target (a scalar t) the training pattern is transformed as follows:

$$\frac{p - \bar{p}}{\sigma_p} \tag{3}$$

where σ_p is the standard deviation of the training pattern p. The training target is transformed relative to its training pattern:

$$\frac{t - \bar{p}}{\sigma_p} \tag{4}$$

Finally, an input vector i is transformed in the same way as a training pattern:

$$\frac{i - \bar{i}}{\sigma_i} \tag{5}$$

where σ_i is the standard deviation of the input vector i. The one-step-ahead forecast f generated for the input i is back transformed as: $f\sigma_i + \bar{i}$.

IV. EXPERIMENTATION

In this section the model proposed in the previous section has been used to forecast the 181 yearly series from the M1 competition. The forecast accuracy of our model will be compared to several benchmarks to assess its relative efficiency. To achieve reproducible results publicly available implementations of the benchmarks (packages from CRAN, the Comprehensive R Archive Network, that is, the official repository of R packages) have been used. The benchmarks are the following ones:

- The ARIMA methodology [13]. We have used the implementation of this methodology supported by the *auto.arima* function from the *forecast* package [14]. In this implementation a non-exhaustive search of

ARIMA models is done, using the corrected Akaike information criterion (AICc) to select a suitable model.

- Exponential smoothing models [15]. Exponential smoothing encompasses a set of models (SES, Holt, Holt-Winters, etc) that decompose a time series into level, trend and seasonal components using exponential smoothing. To forecast with exponential smoothing the *ets* function from the *forecast* package has been used, this function automatically selects the best exponential smoothing model taking into account the AICc.
- A classical GRNN model trained with the time series to be predicted. We have used the *grnn_forecasting* function from the *tsfgrnn* package [16]. The GRNN model is configured in the same way as our proposal, the only difference is that this model is not trained with the whole data set. Also, because this model is trained in the traditional way, with the series that it predicts, the training examples are transformed in a different way. Concretely, the value of a training example is subtracted by the mean of its training pattern and a prediction is back transformed by adding the mean of the input.
- The KNN (k-nearest neighbors) model implemented in the *tsfknn* package [17]. We have used the default parameters of the *knn_forecasting* function.
- The approach developed in [6]. We have used the R implementation of its proposal provided in the paper. In the experimentation we have called this approach *Similarity*.

The forecasting accuracy is measured according to the Mean Absolute Scaled Error, MASE [18]. Scaled errors are an alternative to using percentage errors when comparing forecasting accuracy across series with different magnitudes. For a yearly time series $s = \{y_1, y_2, \ldots, y_n\}$, in which the forecasting horizon is h, the MASE is computed as follows:

$$MASE = \frac{1}{h} \frac{\sum_{t=n+1}^{n+h} |y_t - f_t|}{\frac{1}{n-1}\sum_{t=2}^{n} |y_t - y_{t-1}|} \tag{6}$$

where y_t and f_t are the actual value and the forecast for period t respectively. MASE computes the mean absolute forecast error relative to the mean absolute forecast error of a simple forecasting method. For yearly time series the naive approach, whose forecast is the last value of a series, is normally chosen as "simple method". As can be seen in (6), the mean absolute forecast error is scaled by the mean absolute forecast error of the naive method for the one-step-ahead predictions of the historical values of the series. In order to measure the forecast accuracy of a model across a set of series, the MASE of its forecasts for each series is computed and then averaged. The lower the MASE value, the more accurate the predictions are. In our experimentation the expected MASE value will be higher than one, because we forecast the six future values of a series, and the errors of these forecasts are scaled by one-step-ahead errors of the naive approach.

TABLE 3. Average MASE of the different approaches across the yearly series of the M1 competition.

Method	short	medium	long	all
Exponential smoothing	3.65	4.24	2.75	3.74
ARIMA	3.11	4.24	2.67	3.45
Traditional GRNN	3.22	4.02	**2.46**	3.40
KNN	3.31	4.05	2.47	3.46
Similarity	2.97	**3.77**	2.47	**3.19**
Experiment 1	3.49	4.05	2.91	3.61
Experiment 2	3.09	3.98	2.95	3.38
Experiment 3	3.13	4.03	2.83	3.41
Experiment 4	**2.92**	4.17	3.09	3.38

TABLE 4. Median MASE of the different approaches across the yearly series of the M1 competition.

Method	short	medium	long	all
Exponential smoothing	2.48	**2.52**	1.77	2.32
ARIMA	**2.00**	2.55	1.61	**2.16**
Traditional GRNN	2.24	2.53	1.79	2.21
KNN	2.46	2.54	1.66	2.34
Similarity	2.19	2.55	**1.41**	2.31
Experiment 1	2.61	2.56	2.82	2.58
Experiment 2	2.08	2.60	2.56	2.40
Experiment 3	2.18	2.66	2.45	2.35
Experiment 4	2.11	3.31	1.99	2.45

The first five rows of Table 3 show the average MASE of the benchmarks across the 177 yearly series of the M1 competition. The series have been divided into three categories according to their length: short (10 to 16 values), medium (from 17 to 30) and large (more than 30). There are 92 short series, 62 medium series and 23 long series. The category all represents the 177 yearly series. The original dataset contains 181 yearly series, but we have removed 4 time series of length 9, because they are too short to be forecast by the traditional GRNN model. The method labelled as Experiment 1 is the approach proposed in the previous section, it obtains a worse overall average accuracy than the classical GRNN model across all the series, but it outperforms exponential smoothing.

The remaining rows in Table 3 correspond to additional experiments about training a GRNN model with examples drawn from a pool of series. These experiments are described next. Fig. 6 shows a boxplot for the MASE obtained by the different models on the 177 yearly series of the M1 competition. As can be observed almost all the models obtain very poor results with two series. These bad results will dramatically affect the average MASE across the series. Therefore, it would be interesting to use other measures of central tendency. In this sense, Table 4 shows the median of the MASE for the models across the 177 yearly series of the M1 competition.

A. EXPERIMENT 2

The series of the M1 competition are organized into categories, which are listed in Table 5, with the number of yearly series in each category. In this second experiment

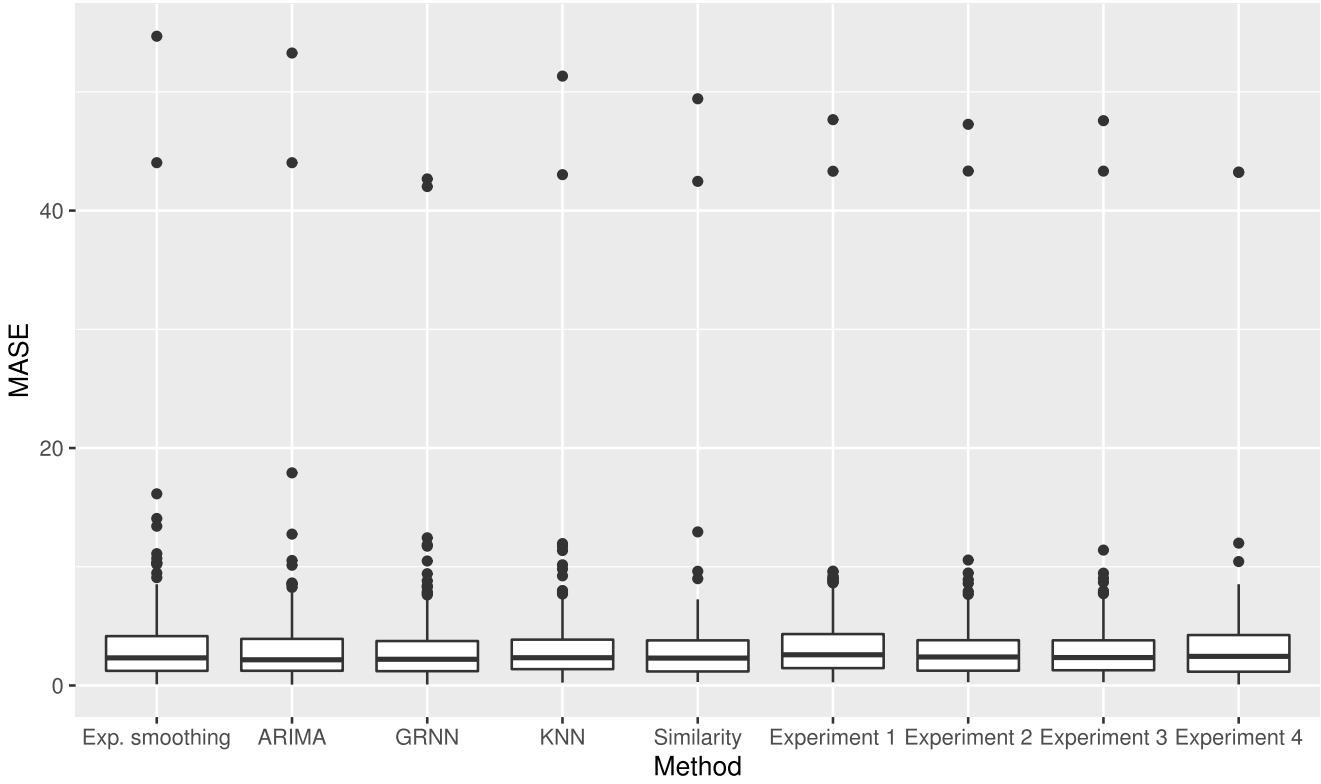


FIGURE 6. Boxplot of the MASE for the different forecasting models.

TABLE 5. Categories of the yearly series of M1 competition.

Category	# of series
Demographics	30
Industry	35
Macro1	30
Macro2	29
Micro1	16
Micro2	29
Micro3	12

we try to discover whether information about the category of a series can be used to improve the forecast accuracy of our proposal. The experiment consists in training the GRNN model forecasting a series s with the pool of series that belong to the category of s. For example, a GRNN model forecasting an Industry series is trained with the 35 series in this category. As can be seen in Tables 3 and 4, this strategy seems to produce better results than the original proposal (Experiment 1), in which the GRNN models are trained with all the yearly series.

B. EXPERIMENT 3

This experiment is a slight variation of the previous one. As in Experiment 2, a GRNN model forecasting a series s is trained with the pool of series belonging to the category of s. However, now s is excluded from the pool. The goal of this experiment is to discover whether a GRNN model can be properly trained by using only *external* series. It must be noted that a minimum of data is needed from the series being predicted. We use the last h historical values of the predicted series as validation set to choose the smoothing factor. Also, the input to GRNN is formed by the last historical values of the series. However, the training examples used by the GRNN model assessing the validation set and the GRNN model forecasting the series belong to *external* series. The forecast accuracy obtained with this experiment (see Tables 3 and 4) is similar to the accuracy of Experiment 2.

It should be noted that with all the proposed approaches the minimum number of historical values needed to forecast a series is reduced. For example, with the modeling decisions described in Section II-A the minimum length needed to forecast a series with the traditional GRNN model is 10. However, with the proposed approaches is 6—the maximum between the length of the forecasting horizon (6) and the length of an input pattern (3).

C. EXPERIMENT 4

The last experiment is based on model combination, that is, in combining the forecasts of several models. Model combination has a long history in time series forecasting [19], [20] and its usefulness has been demonstrated once again in the recent M4 competition [3] where the top-performing methods were combinations.

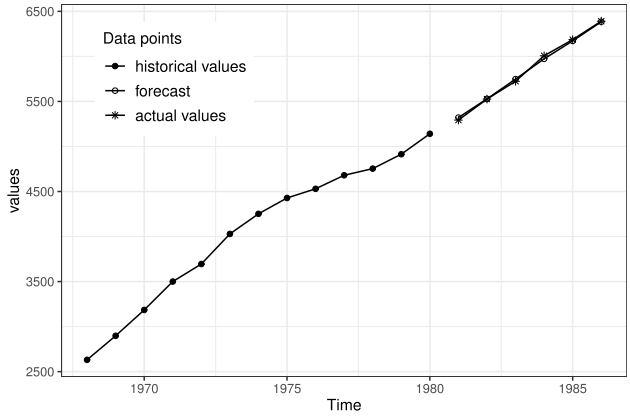


FIGURE 7. Historical values and forecasts for series 142 with a MASE of 0.09 (best forecast).

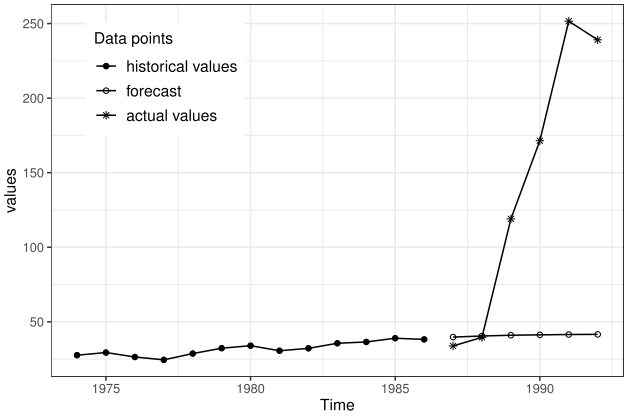


FIGURE 8. Historical values and forecasts for series 76 with a MASE of 43.24 (worst forecast).

In this experiment a time series s belonging to category C is forecast as follows:

1) For every time series t of C a GRNN model is built as in Experiment 1, but the model is trained exclusively with the historical values of t.
2) If category C has N series, the previous step produces N models. Each model is trained with a different series of C and it is used to generate its own forecasts.
3) The N forecasts are combined using the median to produce the final forecast for s.

The forecast accuracy obtained with this experiment (see Tables 3 and 4) across all the series is not bad, in spite of the fact that only one model of the combination is trained with the values of the series being predicted.

Fig. 7 shows the best forecast for the 177 yearly series of the M1 competition in terms of MASE using experiment 4, the forecasts are almost equal to actual future vales. On the other hand, Fig. 8 shows the worst forecast in terms of MASE, it should be noted that the series to be forecast is a difficult one.

V. DISCUSSION

This section analyzes the results of the experimentation conducted in the previous section. Looking at Tables 3 and 4 it can be noted that for the traditional and benchmark models—exponential smoothing, ARIMA, GRNN, KNN and Similarity—the forecasts for long series are clearly more accurate than for short series. This is a reasonable result because, in general, the more data the best a model can be trained. However, the forecast accuracy for the medium-sized series is worse than for the short series. Probably, the set of medium-sized series is particularly difficult to predict. On the other hand, observing Table 4 it seems that our four proposals do not improve with the increasing length of the series. The reasonable explanation for this is that our models are mainly trained by examples drawn from *external* series to the series being predicted and are therefore less sensitive to the length of the predicted series.

We have conducted some statistical tests to detect whether there are significant differences between the models. The tests have been done for the different length categories into which the series have been divided: short, medium and long. According to the recommendations given in [21] a two stage procedure is applied. First, the Iman and Davemport's extension of the Friedman's test is applied to check whether all the approaches are not equivalent in terms of forecast accuracy—the test takes into account the average rank of the different models over the different test sets. If the null hypothesis (i.e., the average rank of all the approaches over the test sets are the same) is rejected, the post-hoc Nemenyi test is used to see whether there are differences between any two models.

For the short series the Iman and Davemport's test produces a p-value of 4.3×10^{-8}, with such a low value the null hypothesis is rejected at any reasonable significance level. One advantage of the post-hoc Nememyi test is that its outcome can be neatly presented with a critical difference diagram. Fig. 9 shows this diagram. According to it, the best average rank among the models is obtained by Experiment 4. In a critical difference diagram groups of models that are not significantly different are connected with a horizontal bar. For example, in Fig. 9 Experiment 4, Experiment 2, Experiment 3, Similarity and ARIMA are connected and are not, therefore, significantly different—at a significance level of 0.05. However, Experiment 4 and Experiment 2 are significantly different from GRNN. This later result is important, because it shows that for short series GRNN can take advantage of being trained by external series, at least, if the training is done as in Experiments 4 and 2.

For the medium-sized series the Iman and Davemport's test produces a p-value of 0.11 and for the long series a p-value of 0.06, therefore at a significance level of 0.05 the null hypothesis is not rejected. However, although there are no significant differences we have decided to show the result of the Nemenyi test to see the average rank of each model. Fig. 10 and Fig. 11 show the critical diagrams for the medium-sized and long series respectively. For long series our proposals obtain the worst results in terms of average rank.

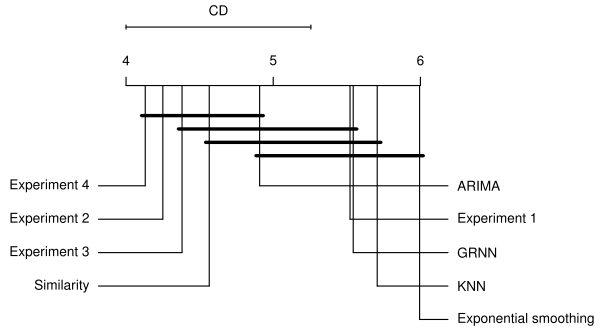


FIGURE 9. Critical diagram of the Nemenyi test for short series comparing all models against each other.

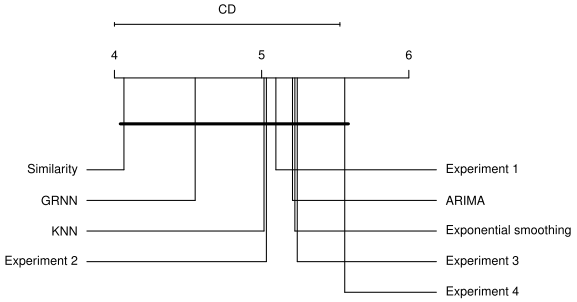


FIGURE 10. Critical diagram of the Nemenyi test for medium-sized series comparing all models against each other.

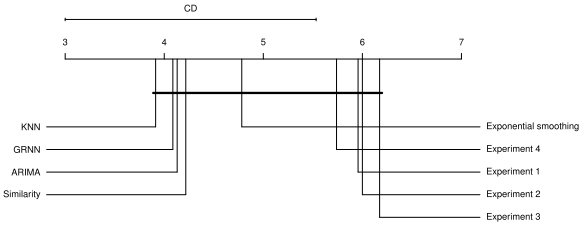


FIGURE 11. Critical diagram of the Nemenyi test for long series comparing all models against each other.

Taking these results into account we can conclude that when we want to forecast the future values of a short series using GRNN it might be effective, in terms of forecast accuracy, to train the model with a pool of series, at least using Experiments 4 or 2.

Our experimentation has used the yearly series of the M1 competition, both as pool of training series and as the target series to be predicted. In order to apply our proposal to any series it would be interesting to have a pool of series that are similar to the one being predicted. If this is not the case, public data sets, such as the 100,000 series of the M4 forecasting competition [3], can be used to find a set of similar series.

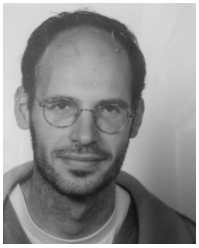
VI. CONCLUSION

This paper has presented new approaches for time series forecasting with generalized regression neural networks models. The main feature of these approaches is that the training examples used to build the models are taken from a pool of series. The experimentation shows that, for short series, two of the approaches improve the forecast accuracy of the traditional GRNN model trained exclusively with the series being forecast. The experimental results also seem to indicate that forecast accuracy might be improved if the series of the pool are similar, in some way, to the series being predicted. A striking result is that, for short series, an effective model can be built using only *external* series as training set. Another interesting outcome is that the proposed approaches can be used as an effective alternative to forecast series that are too short to train a traditional GRNN model.

REFERENCES

[1] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah, "A review on time series forecasting techniques for building energy consumption," *Renew. Sustain. Energy Rev.*, vol. 74, pp. 902–924, Jul. 2017.
[2] G. Dudek, "Neural networks for pattern-based short-term load forecasting: A comparative study," *Neurocomputing*, vol. 205, pp. 64–74, Sep. 2016.
[3] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *Int. J. Forecasting*, vol. 36, no. 1, pp. 54–74, Jan. 2020.
[4] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, "FFORMA: Feature-based forecast model averaging," *Int. J. Forecasting*, vol. 36, no. 1, pp. 86–92, Jan. 2020.
[5] J. E. Boylan, H. Chen, M. Mohammadipour, and A. Syntetos, "Formation of seasonal groups and application of seasonal indices," *J. Oper. Res. Soc.*, vol. 65, no. 2, pp. 227–241, Feb. 2014.
[6] Y. Kang, E. Spiliotis, F. Petropoulos, N. Athiniotis, F. Li, and V. Assimakopoulos, "Déjà vu: A data-centric forecasting approach through time series cross-similarity," *J. Bus. Res.*, vol. 132, pp. 719–731, Aug. 2021.
[7] J. Aznartem, J. Benitezsanchez, D. Lugilde, C. Delinaresfernandez, C. Delaguardia, and F. Sanchez, "Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models," *Expert Syst. Appl.*, vol. 32, no. 4, pp. 1218–1225, May 2007.
[8] M. M. Mostafa, "Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait," *Expert Syst. Appl.*, vol. 37, no. 9, pp. 6302–6309, Sep. 2010.
[9] D. F. Specht, "A general regression neural network," *IEEE Trans. Neural Netw.*, vol. 2, no. 6, pp. 568–576, Nov. 1991.
[10] S. B. Taieb, G. Bontempi, A. F. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 7067–7083, Jun. 2012.
[11] F. Martínez, M. P. Frías, M. D. Pérez, and A. J. Rivera, "A methodology for applying k-nearest neighbor to time series forecasting," *Artif. Intell. Rev.*, vol. 52, no. 3, pp. 2019–2037, Oct. 2019.
[12] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler, "The accuracy of extrapolation (time series) methods: Results of a forecasting competition," *J. Forecasting*, vol. 1, no. 2, pp. 111–153, Apr. 1982.
[13] K. Ord, R. Fildes, and N. Kourentzes, "Autoregressive integrated moving average (ARIMA) models," in *Principles of Business Forecasting*, 2nd ed. New York, NY, USA: Wessex Press, 2017, pp. 155–206.
[14] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *J. Stat. Softw.*, vol. 27, no. 1, pp. 1–22, 2008.
[15] E. S. Gardner, "Exponential smoothing: The state of the art—Part II," *Int. J. Forecasting*, vol. 22, no. 4, pp. 637–666, Oct. 2006.
[16] F. Martínez, F. Charte, A. J. Rivera, and M. P. Frías, "Automatic time series forecasting with GRNN: A comparison with other models," in *Proc. Int. Work-Conf. Artif. Neural Netw.*, Gran Canaria, Spain, 2019, pp. 198–209.
[17] F. Martínez, M. P. Frías, F. Charte, and A. J. Rivera, "Time series forecasting with KNN in R: The tsfknn package," *R J.*, vol. 11, no. 2, pp. 229–242, 2019.
[18] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *Int. J. Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.
[19] J. M. Bates and C. W. Granger, "The combination of forecasts," *J. Oper. Res. Soc.*, vol. 20, no. 4, pp. 451–468, 1969.

[20] M. Hibon and T. Evgeniou, "To combine or not to combine: Selecting among forecasts and their combinations," *Int. J. Forecasting*, vol. 21, no. 1, pp. 15–24, Jan./Mar. 2005.
[21] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.



FRANCISCO MARTÍNEZ was born in Granada, Spain, in 1971. He received the B.S. degree in computer science from the University of Granada, Spain, and the Ph.D. degree from the University of Jaén, Spain. He is currently an Assistant Professor with the Department of Computer Science, University of Jaén. He worked in the fields of parallel programming and computer graphics. His main research interests include time series forecasting and neural networks.

MARÍA P. FRÍAS was born in Cazorla, Spain. She received the B.S. degree in statistical sciences and techniques from the University of Granada and the Ph.D. degree from the University of Jaén, Spain. She is currently an Assistant Professor with the Department of Statistics and Operation Research, University of Jaén. Her research interests include spatial statistics, long-range dependence random field models, and parameter estimation of spatiotemporal random field models.

MARÍA D. PÉREZ-GODOY received the B.Sc. degree in computer science from the University of Granada, in 1993, and the Ph.D. degree in computer science from the University of Jaén, Spain, in 2010. She is currently an Associate Professor with the Department of Computer Science, University of Jaén. She belongs to the Research Group "Intelligent Systems and Data Mining" https://simidat.ujaen.es since its creation, in 2001, where she works in the area of data mining and computational intelligence. Her current main research interests include neural networks, deep learning, classification, regression, and time series forecasting.

ANTONIO J. RIVERA received the B.Sc. and Ph.D. degrees in computer science from the University of Granada, in 1995 and 2003, respectively. He is currently an Associate Professor of computer architecture and computer technology with the Computer Science Department, Universidad of Jaén, Spain. He belongs to the Research Group "Intelligent Systems and Data Mining" https://simidat.ujaen.es since its creation, in 2001, where he works in the area of data mining and computational intelligence. His research interests include deep learning, neural networks, time series forecasting, and multi-label classification and regression.