A study of temporal and recurrent neural networks for CO2 emission forecasting

Oleg Rudenko[a], Oleksandr Bezsonov[a], Nataliia Serdiuk[a] and Kateryna Pasichnyk[a]

[a] *Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine*

Abstract

Time series forecasting is a rapidly growing field of research and provides many opportunities for future work not only in the field of forecasting, but also in many other areas of life. A large number of interesting and noteworthy projects and technologies have already been made in this direction, and various methods of combining forecasting models with ANNs have been proposed in the literature. The advantage of the ANN methods proposed in this paper is that they provide a methodology to approximate real data, i.e., the estimation of the weight vector does not depend on any model. This frees one from model-based selection procedures and sample data assumptions. When nonlinear systems are still in a state of development, it is possible to conclude that the ANN approach offers a competitive and reliable method for system analysis, prediction, and control.According to the Scripps Institution of Oceanography, last year the concentration of carbon dioxide in the atmosphere reached a maximum level of 416.43 ppm (parts per million) for the first time in human history. Every year, this number grows. Therefore, monitoring and forecasting of this indicator is necessary for timely mitigation of the consequences of climate change and ensuring the global climate security of mankind. This paper proposes forecasting the level of carbon dioxide in the atmosphere based on artificial neural networks, namely: recurrent neural network (RNN) and temporal convolutional neural network (TCN).

Keywords [1]

Carbon dioxide concentrations, basic recurrent architectures, temporal convolutional neural networks, app Jupyter Notebook

1. Introduction

Climate change is undoubtedly one of the most serious problems facing humanity, which is why experts consider it an existential threat to our species, that is, one that can lead to irreversible consequences. According to climate data from open sources, the increase in the frequency of extreme weather events is associated with climate change, and decisive action is needed worldwide to address this problem. Otherwise, millions of people will suffer, and their quality of life will decrease significantly in the years to come.

Greenhouse gases such as carbon dioxide (CO_2) and methane (CH_4) trap heat in the atmosphere, thereby keeping our planet warm and friendly to biological species. Despite this, human activities, such as fossil fuel burning and energy generation, emit huge amounts of greenhouse gases, resulting in excessive increases in the Earth's average global temperature. According to the Scripps Oceanographic Institute [1], last year the concentration of carbon dioxide in the atmosphere for the first time in human history reached the maximum level - 416.43 ppm (parts per million). This number,

[1]The Sixth International Workshop on Computer Modeling and Intelligent Systems (CMIS-2023), May 3, 2023, Zaporizhzhia, Ukraine
EMAIL: oleh.rudenko@nure.ua (O. Rudenko); oleksandr.bezsonov@nure.ua (O. Bezsonov); nataliya.serdyuk@nure.ua (N. Serdiuk); kateryna.pasichnyk1@nure.ua (K. Pasichnyk);
ORCID: 0000-0003-0859-2015 (O. Rudenko); 0000-0001-6104-4275 (O. Bezsonov); 0000-0002-0107-4365 (N. Serdiuk); 0000-0001-9228-2380 (K. Pasichnyk)

unfortunately, increases every year. Therefore, control and forecasting of this indicator is necessary for timely mitigation of the consequences of climate change and ensuring the global climate security of mankind.

Considering the urgency of the problem, it was decided to make a comparative prediction of the level of carbon dioxide in the atmosphere for 2023 based on artificial neural networks (ANN) [2], namely: recurrent neural network (RNN) and temporal convolutional neural network (TCN). After that, an analysis was carried out on the subject of which neural network forecasting method from the selected ones showed the best results according to the criteria of forecast accuracy [3, 4].

The forecast dataset was taken from an open source dataset from the Mauna Loa Observatory (MLO), a volcano in Hawaii. It houses a research center that has been monitoring the atmosphere since 1950, and its remote location provides ideal conditions for recording climate data. In 1958, Charles David Keeling organized a CO2 monitoring program and began recording scientific evidence of rapidly increasing CO2 concentrations in the atmosphere. The CO2 dataset was uploaded to Mauna Loa from the Scripps Institution of Oceanography and includes monthly atmospheric CO2 concentrations in parts per million (ppm) from 1958 to 2022.

2. Building forecasting models in the Jupyter Notebook environment

Software implementation, training of neural networks, obtaining graphical and numerical results will be performed in the Jupyter Notebook application.

To begin with, we import a set of libraries required for the project, namely: Pandas [5], Mathplotlib, as well as various functions from the darts and statsmodels library [6].

After importing the libraries, you need to load the selected CO2 data sample into a Pandas data frame (DataFrame), and be sure to apply data preprocessing methods to it, such as removing empty lines, setting a monthly date and time index, and removing null values.

After preprocessing the data set, we use the plot() function to create a simple line graph to look for a trend or seasonal pattern.

Figure 1 shows a graph of a dataset of dates and levels of atmospheric carbon dioxide from March 1958 to the end of 2022.

As can be seen from the graph in Figure 1, there is a clear upward trend, which highlights the fact that the concentration of CO2 in the atmosphere has been increasing rapidly in recent decades. In addition, there is also seasonality due to the planet's natural carbon cycle, as plants absorb and release CO2 in different seasons of the year. In particular, when plants begin to grow in the spring, they clean the atmosphere of CO2 through photosynthesis. Conversely, when trees lose their leaves in the fall, CO2 concentrations increase through respiration.

In this particular case, we have already identified the components (seasonality, trend) by qualitatively analyzing the linear graph (Figure 1), but the seasonal decomposition (Figure 2) helps a lot in more complex cases.

In the next step, the plot_acf() function of the statsmodels library was used to construct the autocorrelation function of the time series, linear dependencies between its lag values were approximated. Autocorrelation is the correlation of the levels of deviations from the trend with each other, that is, the correlation within the same time series, but with different shifts in time. The autocorrelation of time series levels, if it is significant, indicates the presence of a trend, that is, it serves as one of the methods of trend detection. Programmatically, this is described as follows:

```
fig, ax = plt.subplots(figsize = (8,5))
plot_acf(df, ax = ax)                                          (1)
plt.show()
```

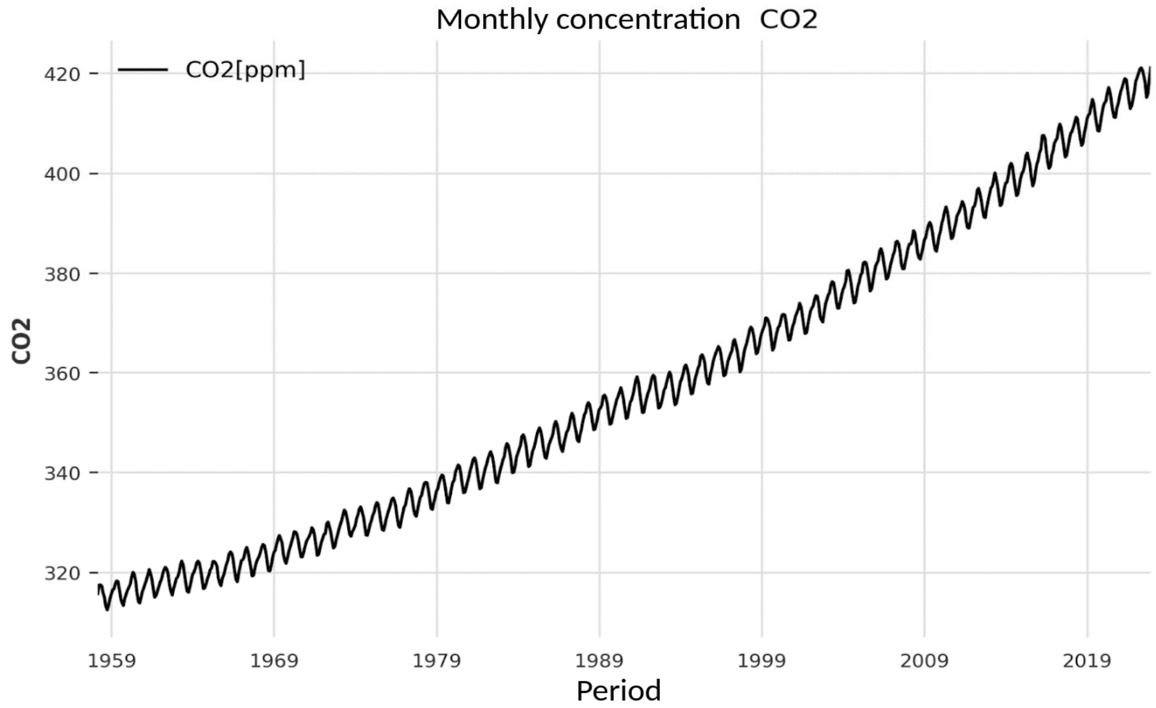

Figure 1: CO2 concentration in the atmosphere since 1958

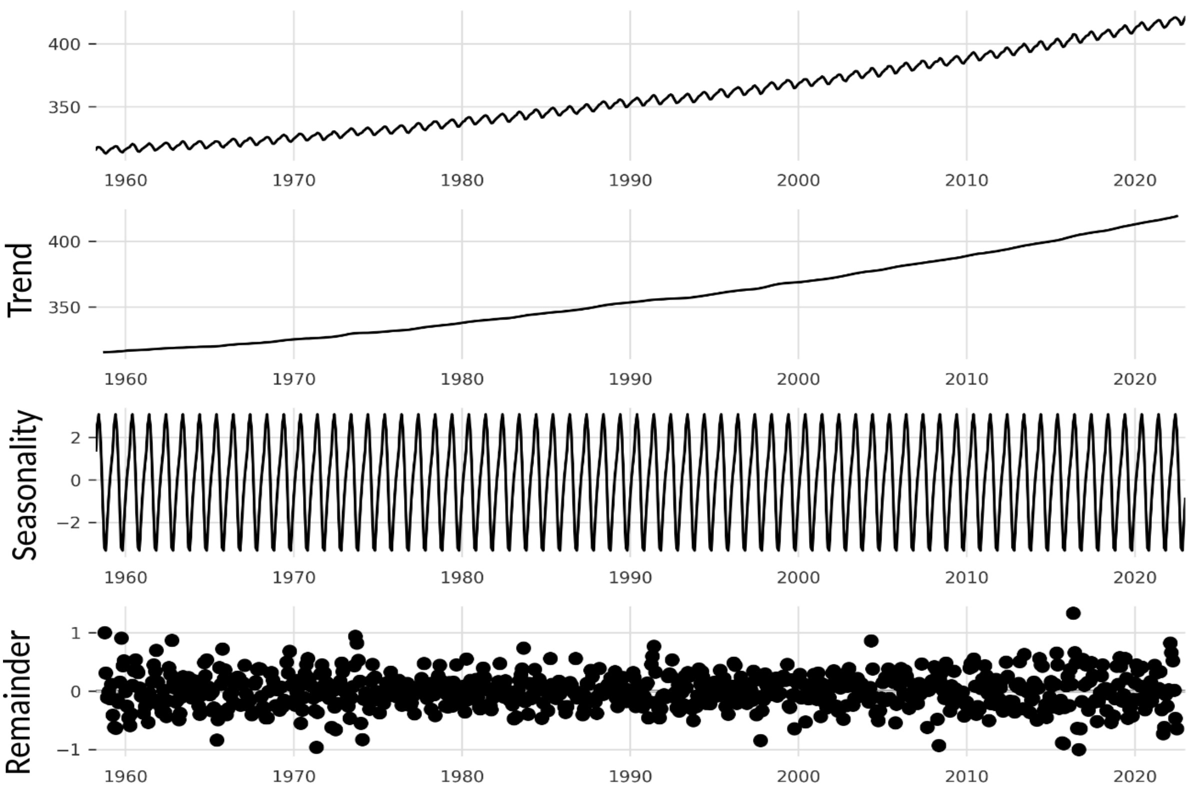

Figure 2: Trend, seasonality and residuals based on the data

The autocorrelation technique consists of sequential calculation of the autocorrelation coefficients of deviations with different shifts in time. Figure 3 shows the result of the autocorrelation function.

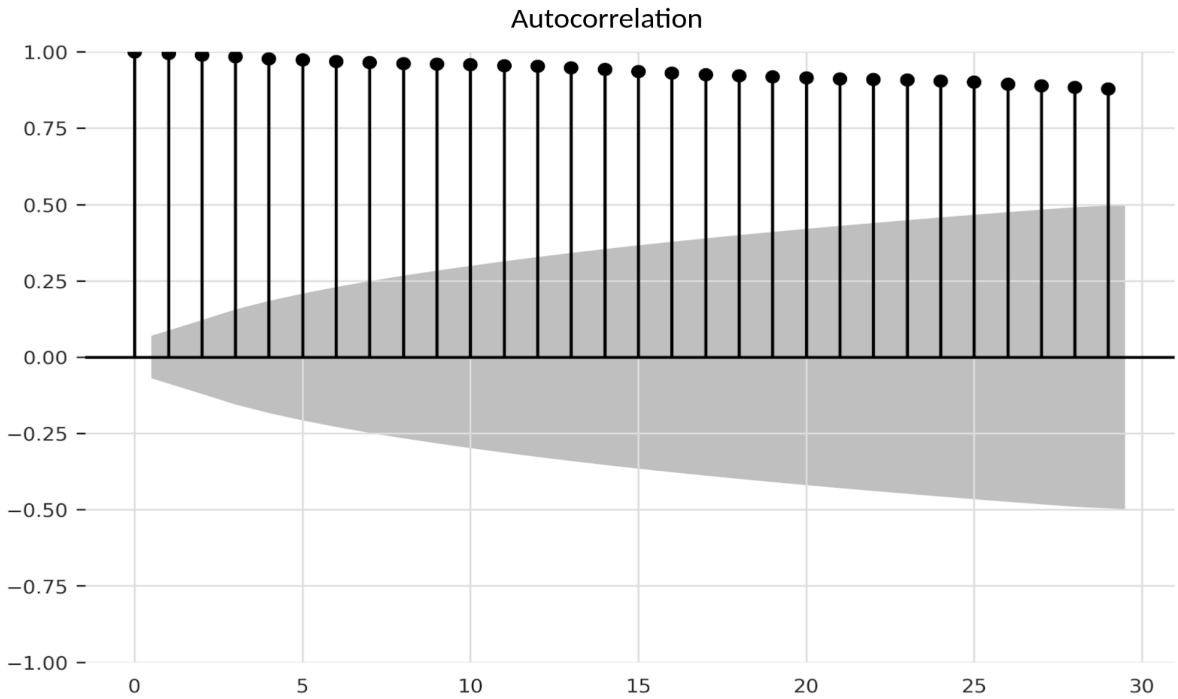

Figure 3: Autocorrelation function

From the trend of the time series, we can see that the autocorrelation is high for small lags and starts to gradually decrease after lag 5. The autocorrelation function should also pick out the seasonal component of the time series, but in our case the seasonal component is not noticeable.

3. Time series forecasting

At this stage, training of selected neural network forecasting models [7] was carried out on the uploaded data set on the level of CO2 in the atmosphere. After the obtained results, a comparison of their effectiveness was performed in order to choose the most accurate model [8, 9] and create a forecast for 2023 based on it.

A number of criteria are used to assess the accuracy of the model. Let's briefly consider each of them:

- The mean absolute error (Mean Absolute Error, MAE) is calculated as the mean absolute difference between the values of the model setting (one step ahead in the example forecast) and the observed historical data
- Mean Squared Error (RMSE) is the square root of the MSE metric. It refers to the same scale as the observed data values. Any small deviation can significantly affect the error rate
- The average percentage of errors (Mean Absolute Percent Error, MAPE) is the average absolute difference in percentages between the values of the model setting and the values of the observed data and the symmetrical average percentage of errors (English: Symmetric Mean Absolute Percent Error, SMAPE)
- Coefficient of determination. In fact, it is a measure of the quality of the model - it is the normalized root mean square error. If it is close to one, then the model predicts the data well, if it is close to zero, then the quality of forecasts can be compared with linear forecasting [8]

Let's start the setup process by loading a pandas data frame into a TimeSeries object as required by the Darts library:

```
series = TimeSeries.from_dataframe(df)
start = pd.Timestamp('123115')                                (2)
df_metrics = pd.DataFrame()
```

Additional functions plot_backtest() and print_metrics() will allow you to plot predictions and display model accuracy metrics, namely MAE, RMSE, MAPE, SMAPE, and R^2.

3.1. Creation of a temporal convolutional network forecasting model

We will consider prediction models based on neural networks, namely, we will create an algorithm for learning a recurrent neural network and a temporal convolutional neural network. For this task, the Darts library is used, which contains standard tools for working with selected neural networks. Convolutional neural networks are a promising field in machine and deep learning [10], and the TCN model was created for even better data analysis, learning and prediction [11]. Recurrent neural networks have become the standard choice in research [12,14] as well as in practical applications [13]. Despite this, the temporal convolutional neural model is an alternative architecture that gives promising results, so its performance will now be tested in practice [15].

First of all, let's set the parameters of the TCN neural network for training:

```
  model = TCNModel(
....input_chunk_length=24,
....output_chunk_length=12,
....n_epochs=100,
....dropout=0.1,
....dilation_base=3,                                    (3)
....weight_norm=True,
....kernel_size=5,
....num_filters=3,
....random_state=0,
)
```

The history_forecasts() function and other service functions are used to test the temporal convolutional network model and display the results. In addition, time series are normalized using the Scaler() class.

The program code describes the given parameters and functions for training and outputting graphical results.

After a series of settings and manipulations with network parameters, training and adjustment of results, we get a graph with a forecast based on the TCN model (Figure 4).

According to the results of the accuracy criteria of this model (Table 1), this model can best predict the level of carbon dioxide in the atmosphere for the coming years. It was possible to achieve the ideal MAPE and SMAPE of 0.1%.

Table 1
Accuracy criteria of the TCN neural network

MAE	RMSE	MAPE	SMAPE	R^2
0.41	0.63	0.1	0.1	0.96

It cannot be argued that the use of neural networks for forecasting tasks is always the best choice, because everything depends on the task and the goal of the research. Each of the methods is successful in its own way. For comparison, we will also train a recurrent neural network.

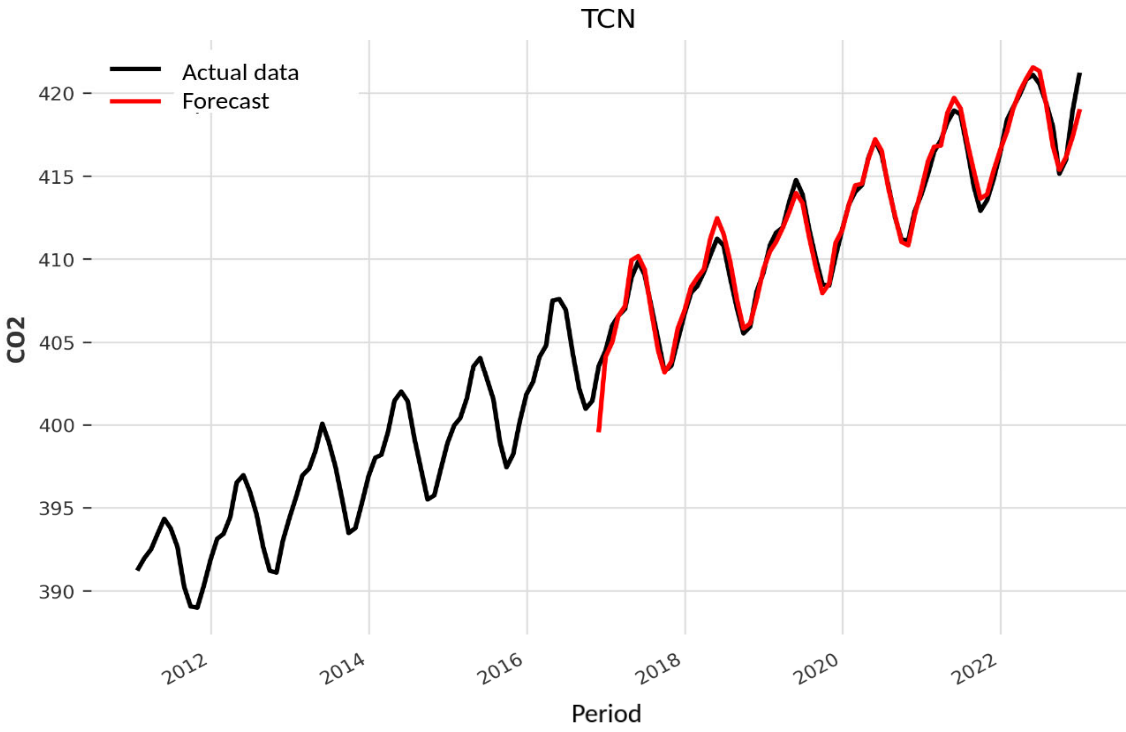

Figure 4: Prediction graph based on TCN network

3.2. Creation of a prediction model based on a recurrent neural network

Modern networks in which connections between nodes form a certain cycle are called backpropagation or recurrent networks. Because of this internal state, the network can exhibit dynamic behavior over time. In recurrent networks, communication between neurons can come not only from the lower layer to the upper one, but also from the neuron to "itself", more precisely, to the previous value of this neuron or other neurons of the same layer. This allows to display the dependence of the variable on its eigenvalues at different points in time [14].

Recurrent neural networks are often compared to temporal convolutional neural networks due to the fact that both are able to learn effectively and are actively used in various tasks, not only in prediction, but also in other areas of deep learning.

The learning process is similar to TCN. Without special skills and programming skills, it is possible to get a forecast thanks to machine learning libraries, in our case it is Darts.

Let's set the parameters for training the RNN network:

```
model = RNNModel(
    model="RNN",
    hidden_dim=40,
    dropout=0,
    batch_size=24,
    n_epochs=50,
    optimizer_kwargs={"lr": 1e-3},
    log_tensorboard=True,
    random_state=40,
    training_length=60,
    input_chunk_length=12,
    force_reset=True,
    save_checkpoints=True,
)
```
(4)

The process of selecting the best input parameters of the recurrent neural network took place at the expense of theoretical skills, as well as at the expense of repeated testing of network training on certain parameters. Sometimes the process of learning the network took up to an hour and the graph showed unsuccessful attempts to learn the network, but after optimizing the parameters, the result was ready in just 5-10 minutes. Below is the code with the functions for the RNN network.

```
model_name = 'RNN'
plt.figure(figsize = (8, 5))
scaler = Scaler()
scaled_series = scaler.fit_transform(series)
forecast = model.historical_forecasts(scaled_series, start=
start,
forecast_horizon=12, verbos=True)
plot_backtest(series, scaler.inverse_transform(forecast), m
odel_name)
df_dl = print_metrics(series, scaler.inverse_transform(fore
cast), model_name)
df_metrics = df_metrics.append(df_dl)
plt.show()
df_dl
```
(5)

The process of learning this network was not easy and required a lot of CPU power. Figure 5 shows the final result of training the RNN network on input data about the CO2 level in the atmosphere.

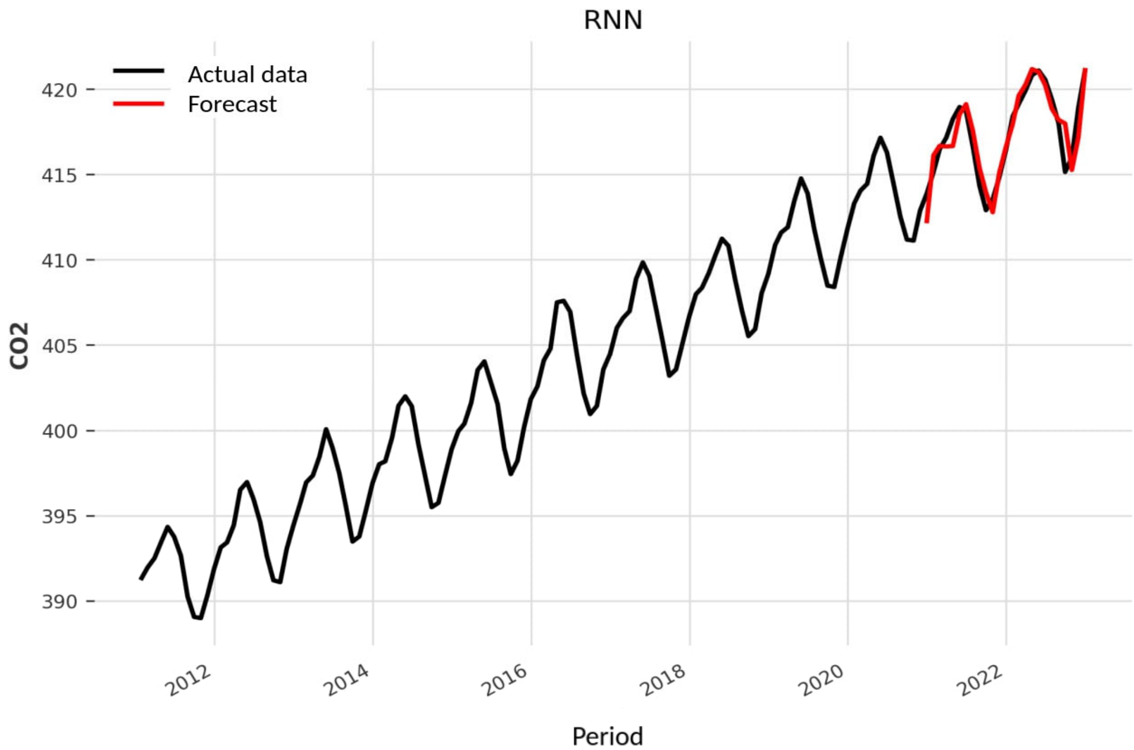

Figure 5: Prediction graph based on RNN network

As for the accuracy criteria of the model (Table 2), they differ significantly from TCN.

It can be seen from Table 2 that recurrent neural networks should not be used for forecasting in this forecasting problem, since the symmetric average relative error indicator is 0.17%, and the coefficient of determination is almost a tenth worse compared to TCN and the exponential smoothing method.

Table 2
Accuracy criteria of a recurrent neural network

MAE	RMSE	MAPE	SMAPE	R^2
0.72	0.96	0.17	0.17	0.84

4. Comparative analysis of the obtained research results

In order to visually present the effectiveness of the researched methods [16] for solving the problem, the results of the accuracy criteria of forecasts for each of them were collected and a comparative analysis was carried out.

Summary Table 3 contains the metrics of forecast accuracy criteria that were used in the work, namely: MAE, RMSE, MAPE, SMAPE and, according to each of the forecasting models: two models based on ANN.

Table 3
Metrics of forecast accuracy criteria for each of the studied forecasting methods

Forecasting method	Forecast accuracy criteria				
	MAE	RMSE	MAPE	SMAPE	R^2
RNN	0.72	0.96	0.17	0.17	0.84
TCN	0.41	0.63	0.1	0.1	0.96

The high efficiency of the TCN model is due to its ability to accept a sequence of any length and output it as a sequence of the same length as the input, as well as preventing information leakage from the future to the past due to the use of causal convolutions.

Recurrent neural networks are famous for their effectiveness in unsegmented handwriting and speech recognition tasks. They are often compared to temporal convolutional neural networks, but usually the latter have more advantages over the former.

5. Creation of a forecast of the level of carbon dioxide for 2023

After comparing all the results of forecasting methods used in this work, our forecast of CO2 concentration in the atmosphere for 2023 will be based on the temporal convolutional neural network, because it gave the best results.

Input parameters for creating a forecast for 2023 are copied from the TCN model. In order to fit the model and then display the results, the fit() function is used:

```
  model_name = 'Forecast'
  plt.figure(figsize = (8, 5))
  scaler = Scaler()
  scaled_series = scaler.fit_transform(series)
  model.fit(scaled_series)                                          (6)
  forecast = model.predict(12)
  plot_backtest(series, scaler.inverse_transform(forecast),
model_name)
  print(forecast.pd_dataframe())
```

Figure 6 shows the resulting graph with a forecast for the next year based on a temporal convolutional neural network.

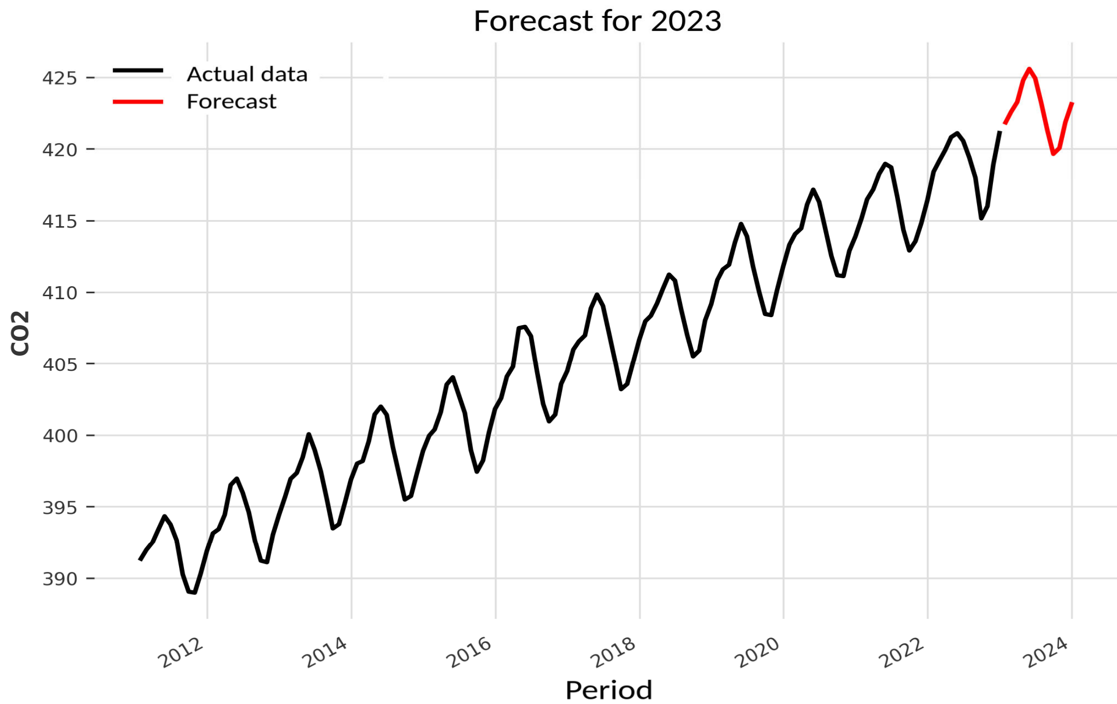

Figure 6: The result of the forecast of the level of CO2 in the atmosphere for 2023

It can be visually observed that the components of the time series were successfully determined by the model. Table 4 contains the numerical indicators of the level of CO2 in parts per million corresponding to each month of 2023.

Table 4
Forecast for 2023 created by the TCN model

Date in year-month-day format	CO2 level, (ppm)
2023-01-31	421.836774
2023-02-28	422.576312
2023-03-31	423.262611
2023-04-30	424.778723
2023-05-31	425.581957
2023-06-30	424.922701
2023-07-31	423.185616
2023-08-31	421.267827
2023-09-30	419.635597
2023-10-31	420.052928
2023-11-30	421.851032
2023-12-31	423.111864

Unfortunately, the level of carbon dioxide is constantly increasing, which is a reason to worry and take decisive action to save the environment from the consequences. With the help of a customized forecasting model based on a temporal convolutional neural network, forecasts can be made several years ahead. Undoubtedly, humanity must develop a strategy to reduce CO2 levels in the atmosphere in order to prevent further climate change, a negative impact on all living organisms and the planet as a whole.

6. Conclusions

It should be noted that a perfect forecast is impossible due to the presence of a large number of factors that are difficult to estimate with a high percentage of accuracy. Therefore, instead of searching for an ideal forecast, it is much more important to have good knowledge in the area of existing models and their correct application depending on the specifics of the data and subject area, the ability to adapt to non-ideal forecasts.

Due to the fact that the forecast depends on past data, its reliability and accuracy will decrease in proportion to how far into the future the task of calculating it is. It is worth noting that the accuracy of the forecast and the costs of its implementation are interrelated. The best forecasts are not necessarily the most accurate. Factors such as its purpose and data availability play an important role in determining the desired level of accuracy.

The dynamic behavior of most time series in our real life, with its autoregressive and legacy moving average terms, makes it difficult to forecast nonlinear time series that contain legacy average terms using computational intelligence methodologies such as neural networks. This model method allows you to determine specific models at low computational costs.

The results show that temporal convolutional neural networks convincingly outperform basic recurrent architectures in a wide range of sequence modeling tasks.

At the moment, new approaches to forecasting time series are emerging, the purpose of which is to overcome the problems of assessing the state of the market, dimensionality of models, identifying signs of higher-level systems, climate, etc. These approaches are based on the application of such branches of modern mathematics as: evolution, the theory of stochastic modeling, the theory of catastrophes and the theory of self-organizing systems, including genetic algorithms and fuzzy logic.

This work can be useful in the future for solving the problems of excessive amount of carbon dioxide in the ambient atmosphere and the prediction model based on TCN can be optimized due to better setting of hyperparameters. This method is more complex and takes more time, but can show significant improvements in prediction results.

7. References

[1] NOAA: Global Monitoring Laboratory, 2022. URL: https://gml.noaa.gov/.

[2] Rudenko O. G., Bodyanskyi E. V. Artificial neural networks: a study guide. Kharkiv: SMIT Company LLC, 2006. 404 p.

[3] Ratnadip Adhikari, Agrawal R. K. An Introductory Study on Time Series Modeling and Forecasting. LAP Lambert Academic Publishing, 2013. 76 p.

[4] Lara-Benítez P., Carranza-García M., Luna-Romera J. M., Riquelme J. C. Temporal Convolutional Networks Applied to Energy-Related Time Series Forecasting. Applied Sciences. 2020. Vol. 10, no. 7, p. 2322.

[5] Pandas: User guide, 2019. URL: https://pandas.pydata.org/docs/user_guide/

[6] Darts: Time Series Made Easy in Python, 2022. URL: https://unit8co.github.io/darts/

[7] Song G., Li H., Witt S.F. Recent developments in econometric modeling and forecasting. Journal of Travel Research. 2005. Vol. 44, 82–99.

[8] Archer, B. H. Demand forecasting and estimation. In J. R. B. Ritchie, & C. R. Goeldner (Eds.). New York: Wiley. 1987. 75–85.

[9] Bodyanskiy Ye., Kokshenev I., Kolodyazhniy V. An adaptive learning algorithm for a neo-fuzzy neuron // Proc. 3rd Int. Conf. of European Union Soc. for Fuzzy Logic and Technology (EUSFLAT 2003). – Zittau, Germany, 2003. – P. 375–379.

[10] L.J. Cao, Francis E.H. Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transaction on Neural Networks. 2003. Vol. 14, 1506-1518.

[11] C. Müller i S. Guido, Introduction to machine learning with Python: a guide for data scientists. 1st ed. Sebastopol: CA: O'Reilly Media, Inc, 2016, 398 p.

[12] O. Rudenko, O. Bezsonov, i O. Romanyk, Neural network time series prediction based on multilayer perceptron. Development Management. 2019. Vol. 17, vol. 1. P. 23–34.

[13] How to Build a Tensorflow Training Pipeline. Medium: Towards Data Science, 2022. URL: https://medium.com/towards-data-science/how-to-quickly-build-a-tensorflow-training-pipeline-15e9ae4d78a0
[14] Bai S., Kolter J. Z., Koltun V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv, Arp. 19, 2018.
[15] Smith J. E. A Temporal Neural Network Architecture for Online Learning. arXiv, Feb. 22, 2021.
[16] T. Rashid, Make Your own neural network. s.l.: CreateSpace Independent Publishing Platform, 2016. 274 p.