International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 53 No. 2 November 2025, pp. 71-79

Time Series Forecasts Of Co2 Emission Variations In Madagascar Based On 1d-Cnn

[1] Harimino Andriamalala RAJAONARISOA, [2]Adolphe Andriamanga RATIARISON

[1]Dynamic Laboratory of the Atmosphere, Climate and Oceans,
Doctoral School of Physics and Applications
University of Antananarivo,
Madagascar
[1]hariminondriamalala@gmail.com, [2] adolphe.ratiarison@univ-antananarivo.mg
Corresponding Author: Harimino Andriamalala RAJAONARISOA. E-mail: hariminondriamalala@gmail.com

Abstract— The aim of this work was to use the 1D-CNN model to model the evolution of CO_2 emission variations relative to the reference year 1990 in Madagascar. More specifically, the goal was to determine the optimal number of epochs for the 1D-CNN model that produces the best modeling performance. The datasets consisted of an annual time series of CO_2 emission variations in Madagascar relative to 1990, covering the years from 1991 to 2022. For the experiment, the dataset was split into two parts: 80% (from 1991 to 2018) was used to train the model, and 20% (from 2018 to 2022) was used to test the model. The simulation was performed every 5 epochs, and the difference between the actual and predicted values was measured using the MAE (Mean Absolute Error) metric. The optimal number of epochs was determined based on the curve showing the evolution of the average MAE between the training and test data as a function of the number of epochs. After simulation, the minimum average MAE was observed at the 1965th epoch. The results of the 1D-CNN model forecasts, extending to a 10-step horizon, predict a more or less stationary trend in CO_2 emissions relative to the reference year 1990 in Madagascar beyond 2023.

Keywords—1D-CNN; CO_2; Epoch; Mean Absolute Error; Time series forecasts

I. Introduction

CO_2 emissions are one of the key indicators of air pollution. Globally, CO_2 emissions have been increasing since the Industrial Era [1]. However, during the year 2020, marked by the COVID-19 pandemic, emissions significantly dropped due to worldwide lockdowns [2]. In Madagascar in particular, the trend in the curve showing the variation in CO_2 emissions for the current year compared to 1990 shows a similar pattern to the global average, but with a much less pronounced slope [1] [3] [4]. After the lockdown, the evolution curve takes on a new pattern. It is therefore interesting to study the evolution of this parameter following the resumption of human activities.

The problem is therefore to find a model capable of simulating the future evolution of CO_2 emissions in Madagascar. Among the models capable of performing this task, the 1D-CNN model [5] [6] [7] [8] [9] was studied. This work aimed to determine the optimal 'epoch' parameter to obtain the best modeling of CO_2 emissions variation relative to the reference year 1990 using the said model.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 53 No. 2 November 2025, pp. 71-79

II. Materials and Methods

A. *EXPERIMENTAL DATA*

Table 1 summarizes the datasets used for the experiment. The datasets can be downloaded from:

https://data.worldbank.org/indicator/EN.GHG.CO2.ZG.AR5?locations=MG

These data relate to the variation in emissions (in %) for the current year compared to the reference year 1990 for carbon dioxide concerning carbon dioxide—one of the six greenhouse gases (GHGs) covered by the Kyoto Protocol. Emissions come from the agriculture, energy, waste, and industry sectors, excluding the LULUCF (Land Use, Land-Use Change and Forestry) sector. This measurement is standardized in carbon dioxide equivalents using the Global Warming Potential (GWP) factors from the IPCC's Fifth Assessment Report (AR5).

TABLE I. Variation in CO_2 Emissions IN Madagascar Compared to the Reference Year 1990 (in %)

Source: https://data.worldbank.org/indicator/EN.GHG.CO2.ZG.AR5?locations=MG

Year	Variation in CO_2 emissions in Madagascar Compared to the Reference Year 1990 (in %)
1991	6.9
1992	8.3
1993	12.2
1994	38.3
1995	55.8
1996	49.7
1997	64.2
1998	83.4
1999	88.1
2000	86.6
2001	88.4
2002	28.4
2003	70.7
2004	84.6
2005	97
2006	88.8
2007	98.6
2008	106
2009	95.6
2010	116.8
2011	152.7
2012	214
2013	235.9
2014	245.1

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 53 No. 2 November 2025, pp. 71-79

Year	Variation in CO_2 emissions in Madagascar Compared to the Reference Year 1990 (in %)
2015	276.5
2016	258.1
2017	291.3
2018	271.9
2019	340.77
2020	211.4
2021	236.9
2022	251.4

B. STEPS OF THE MODELING PROCESS

The modeling of the time series data was carried out according to the following steps:

- ✓ Step 1: Preprocessing of the experimental data;
- ✓ Step 2: Training of the 1D-CNN model;
- ✓ Step 3: Testing of the model;
- ✓ Step 4: Selection of the optimal number of epochs;
- ✓ Step 5: Modeling using 1D-CNN with the optimal epoch.

B.1 Step 1: Preprocessing of the experimental data

In the context of our experimentation, the data preprocessing steps were:

- Splitting the dataset into training data and test data;
- Creating the sequences.

B.1.1 Splitting the dataset into training data and test data

The training data consisted of 80% of the total data, that is, the data from 1991 to 2014 (over 24 years). The test data were made up of 20% of the entire dataset. The test data consisted of the data from 2015 to 2020 (over 6 years).

B.1.2 Creating the sequences

In this study, the data consisted of univariate time series, representing the variation (in %) of CO_2 emissions for the current year compared to the reference year 1990 in Madagascar. Before training, the data were first normalized to facilitate the learning process of the model and ensure a more stable convergence. This scaling can be done, for example, using a MinMaxScaler from scikit-learn in Python.

Next, the normalized series was split into sliding subsequences of length n_steps = 3, which corresponded to the number of inputs for the 1D-CNN model [5] [6] [7] [8] [9].

For example, from the sequence:
scaled_seq_train = [0.2, 0.3, 0.5, 0.7, 0.6]
The slicing produces the following sequences:
• $X_1 = [0.2, 0.3, 0.5] \rightarrow y_1 = 0.7$
• $X_2 = [0.3, 0.5, 0.7] \rightarrow y_2 = 0.6$

This transformation was applied to both the training data (scaled_seq_train) and the test data (scaled_seq_test), resulting in pairs (X_i, y_i) for supervised learning.

B.2 Step 2: Training of the 1D-CNN model

The training of the 1D-CNN model [5] [6] [7] [8] [9] was done using only the data from 1991 to 2014.

B.2.1 Architecture of the 1D-CNN model used.

The proposed model was based on a one-dimensional convolutional neural network (1D-CNN) architecture [5] [6] [7] [8] [9] developed with the Keras library of TensorFlow in Python.

This type of network is particularly suited for processing sequential data such as time series or signals. The architecture consists of the following layers:

- A one-dimensional convolutional layer (Conv1D) with 64 filters and a kernel size of kernel_size = 2, with a ReLU activation function;
- A flattening layer;
- The first dense layer;
- The second and final dense layer.

a) The one-dimensional convolutional layer Conv1D

This layer allows capturing local temporal patterns in the input sequences [5] [6] [7] [8] [9]. For example, an input sequence [x_1, x_2, x_3] to be passed through a Conv1D layer with a single filter and kernel_size = 2 that is, filter = [w_1, w_2], and let b be the bias. At the output of the Conv1D layer (without activation), the outputs are pos1 and pos2 such that:

$$pos1 = w_1 * x_1 + w_2 * x_2 + b \quad (1)$$
$$pos2 = w_1 * x_2 + w_2 * x_3 + b \quad (2)$$

This process was repeated 64 times with different weights, giving 64 outputs for pos1 and 64 outputs for pos2. The application of the ReLU activation function sets any negative values potentially found in the 64 outputs for pos1 and the 64 outputs for pos2 to zero.

Indeed, the ReLU (Rectified Linear Unit) is defined by:

$$ReLU(z) = max(0, z) \quad (3)$$

It transforms each output z_i of the convolution as follows:

$$\text{If } z_i > 0 \text{ then } ReLU(z_i) = z_i \quad (4)$$
$$\text{If } z_i \leq 0 \text{ then } ReLU(z_i) = 0 \quad (5)$$

Thus, at the input of the convolution layer, we have a tensor of dimension (3, 1), and at the output, a tensor of dimension (2, 64).

b) A flattening layer

The flatten layer [5] [6] [7] [8] [9] flattens the convolutional output into a one-dimensional vector so that it can be used in the subsequent dense layers. In our case, the 2×64 matrix was flattened into a vector of size 128.

c) The first dense layer

The first dense layer consists of 64 fully connected neurons with a ReLU activation function. This layer learns complex combinations of features extracted by the convolutional filters, acts as an interpreter of the local patterns previously detected, and enables the network to model non-linear relationships between the temporal sequences and the target output.

The dense layer [5] [6] [7] [8] [9] receives a vector of size 128 from the flattened layer as input. The parameters of this layer are:

- The weight W, which is a matrix of dimensions 128x64 (each of the 64 neurons is connected to all 128 elements);
- The bias b, which consists of 64 values (one value per neuron).

Each of the 64 neurons in this layer performs the following operations:

$$z_i = \sum_{j=1}^{128} w_{ji} x_j + b_i \qquad (6)$$

$$\text{Then } a_i = ReLU(z_i) \qquad (7)$$

Where :

- w_{ji} denotes the learned weight connecting input j to neuron i in the Dense (64) layer;
- x_j is the input value received by the Dense (64) layer;
- b_i is the bias associated with neuron i.

Therefore, the final output of this layer is a vector of size 64, with the ReLU activation applied.

d) The second and last dense layer

The final dense layer is the output layer of the 1D-CNN model [5] [6] [7] [8] [9]. This layer consists of a single fully connected neuron without an activation function.

The layer receives a vector of size 64 from the previous dense layer as input and simply performs the following calculation:

$$\hat{y} = \sum_{i=1}^{64} w_i \, x_i + b \qquad (8)$$

Where:

$\hat{y}$: The value predicted by the model;

x_i : The output value from the previous layer;

w_i : The learned weight;

b : The learned bias.

B.2.2 Compilation and training of the 1D-CNN model

Compilation is the preliminary step before training the 1D-CNN model [5] [6] [7] [8] [9]. The compilation parameters are the optimizer and the loss function.

The model was compiled using the Adam optimizer [10], known for its robustness and fast convergence. This optimizer dynamically adjusts the learning rate of each parameter.

The loss function used was Mean Square Error (MSE) given by the formula:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2 \quad (9)$$

Where:

- $\hat{y}_i$ is the predicted value from the model;
- y_i is the expected value;
- n : is the number of observations.

The model was trained with the default batch size value of 32 and a maximum epoch value of 2000.

The batch size [11] determines how many samples are processed together before updating the model weights.

An epoch [12] corresponds to one full pass through the entire training dataset.

B.3 Step 3: Testing the 1D-CNN model

After training, the model was evaluated on the test data. The performance was measured using the Mean Absolute Error (MAE) metric, which measures the average difference between the model's output and the observed data. The mathematical formula to calculate the MAE is:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (10)$$

Where:

$\hat{y}_i$ is the predicted value from the model;

y_i is the observed value.

B.4 Step 4: Selection of the optimal number of epochs

The simulation was carried out with a maximum epoch value of 2000. The selection of the optimal number of epochs is based on the evolution of the MAE curve between the training and test data of the model as a function of the epoch. The optimal epoch is the one corresponding to the minimum Mean Absolute Error.

B.5 Step 5: Modeling with 1D-CNN using the optimal epoch

Once the optimal epoch value was determined, the model was trained using the training data and the optimal epoch as the training parameter.

The model was then tested again using the test data. The trained and tested model was then capable of forecasting the time series over a 10-step horizon.

III. RESULTS AND INTERPRETATIONS

A. Result of the Search for the Optimal Epoch Value

Figure 1 shows the evolution of the error curve (in %) as a function of the epoch for the training data (in blue), the test data (in red), and the average of the two curves (in green). The modeling error decreases rapidly during the initial epochs for both the training and test data. After the 50th epoch, the error continues to decrease more gradually, and the minimum error is only reached at the 1965th epoch. This corresponds to the optimal epoch value sought.

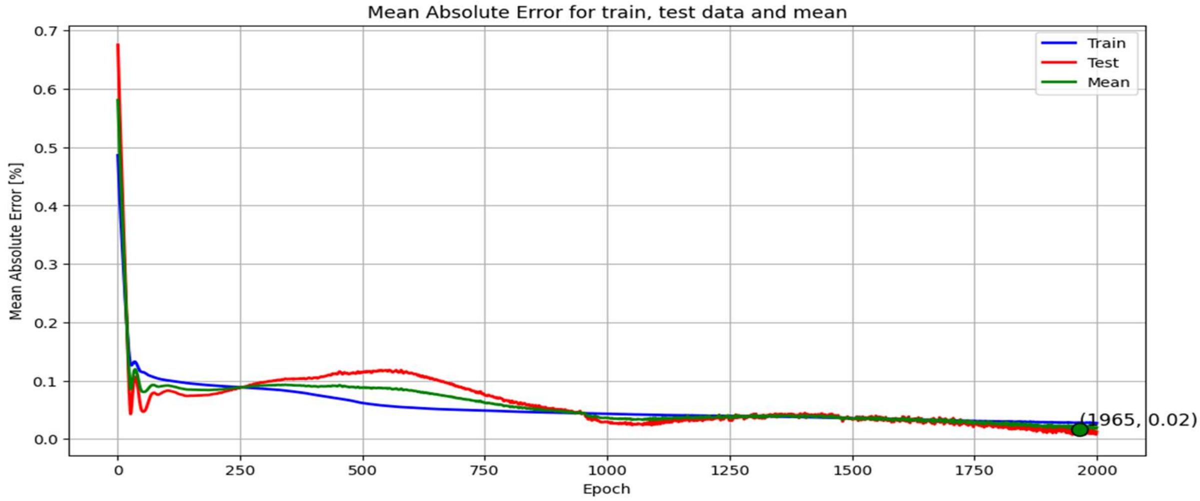

Fig. 1 MAE Curve for the Search of the Optimal Epoch Value

B. Result of the Modeling Using the Optimal Epoch Value

Figure 2 shows the result of modeling the evolution of CO_2 emissions in Madagascar (in %) compared to the reference year 1990. The black curve represents the observed data. The blue curve corresponds to the training data, the red curve to the test data, and the green dashed curve shows the forecast up to horizon 10. The results of the 1D-CNN model's forecast up to horizon 10-year horizon predict a more or less stationary trend in CO_2 emissions compare to the reference year 1990 in Madagascar beyond 2023.

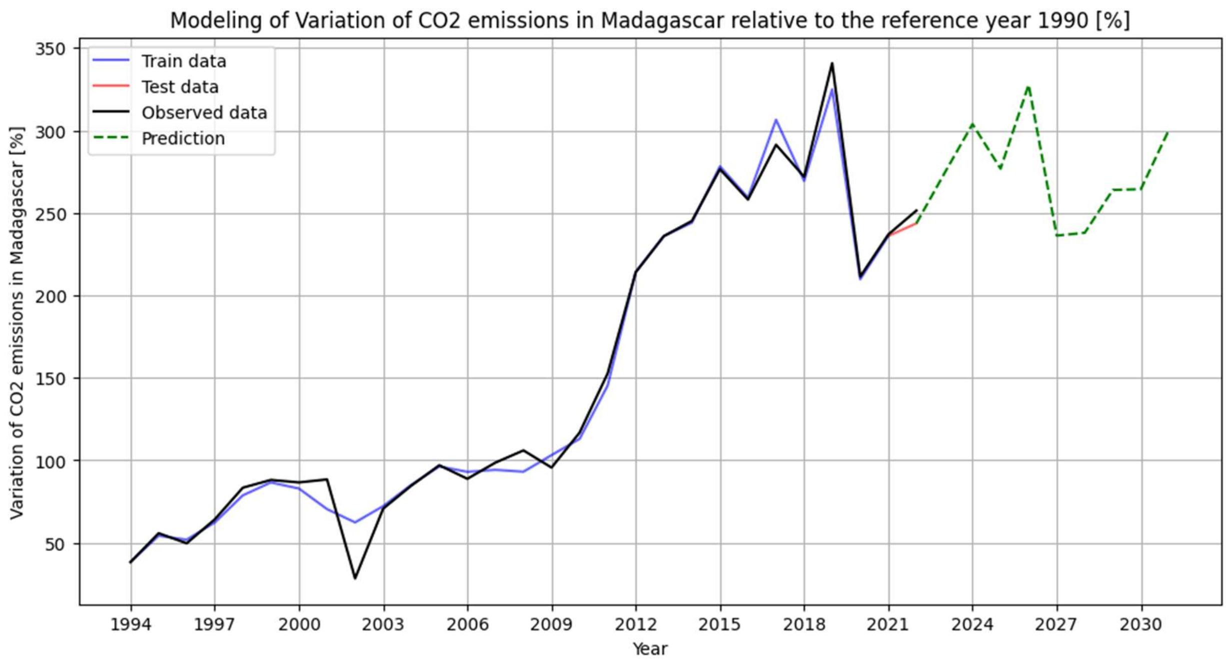

Fig. 2 Results of modeling CO_2 variation in Madagascar using the 1D-CNN model.

IV. DISCUSSION

The results obtained using the 1D-CNN model [5] [6] [7] [8] [9] demonstrated that this approach is relevant for modeling the evolution of CO_2 emission variations in Madagascar compared to the reference year 1990. The search for the optimal number of epochs, based on the analysis of the behavior of the Mean Absolute Error (MAE) as a function of epochs, identified epoch 1965 as offering the most satisfactory compromise between fitting the training data and generalizing to the test data. This point corresponds to a local minimum of the average MAE curve, suggesting a good balance between underfitting and overfitting [13].

Choosing a step size of 5 epochs for the analysis provided sufficient granularity to detect significant changes while maintaining a reasonable computational complexity. This type of strategy is also employed in other time series modeling studies to identify optimal hyperparameters [14].

Furthermore, the relatively stationary trend observed in the forecasts beyond 2023 aligns with uncertainties related to Madagascar's current climate, economic, and environmental policies. In the absence of major shifts in energy policy or economic activity, it is plausible that CO_2 emission variations remain stable, as also suggested by other regional studies in sub-Saharan Africa [15].

The 1D-CNN model, commonly used for pattern recognition in univariate time series data, appears well suited here for environmental time series despite its relative simplicity, compared to recurrent models such as LSTM or GRU. Its effectiveness lies notably in its ability to automatically extract relevant local features through convolutional filters while maintaining a less resource-intensive architecture [16].

However, some limitations should be noted. First, the study did not consider exogenous explanatory factors (economic, demographic, energy data) that could improve prediction accuracy. Additionally, the forecasting horizon remained short (10 years), which limited the scope of medium- and long-term conclusions.

V. CONCLUSION

This study showed that one-dimensional convolutional neural networks (1D-CNN) provide an effective solution for modeling simple time series. By automatically extracting local features within sequences, the model was able to produce accurate predictions while being lighter and faster to train than traditional sequential models such as LSTMs. Future works include extending the approach to multivariate time series, adding regularization layers like Dropout or Batch Normalization, comparing with other architectures such as LSTM, GRU, or Transformers, and incorporating cross-validation for increased robustness.

REFERENCES

[1] Ritchie, H., Rosado, P., & Roser, M. (2023). CO_2 and greenhouse gas emissions. *Our world in data.*

[2] Friedlingstein, P., O'sullivan, M., Jones, M. W., Andrew, R. M., Hauck, J., Olsen, A., ... & Zaehle, S. (2020). Global carbon budget 2020. *Earth System Science Data Discussions, 2020*, 1-3.

[3] Le Quéré, C., Jackson, R. B., Jones, M. W., Smith, A. J., Abernethy, S., Andrew, R. M., ... & Peters, G. P. (2020). Temporary reduction in daily global CO2 emissions during the COVID-19 forced confinement. *Nature climate change, 10*(7), 647-653.

[4] Watch, C. (2023). Historical GHG Emissions (1990–2020). Washington, *DC: World Resources Institute*.

[5] Markova, M. (2022, April). Convolutional neural networks for forex time series forecasting. In AIP Conference Proceedings (Vol. 2459, No. 1). *AIP Publishing.*

[6] Guessoum, S., Belda, S., Ferrandiz, J. M., Modiri, S., Raut, S., Dhar, S., ... & Schuh, H. (2022). The short-term prediction of length of day using 1D convolutional neural networks (1D CNN). *Sensors, 22*(23), 9517.

[7] Lang, C., Steinborn, F., Steffens, O., & Lang, E. W. (2020). Applying a 1D-CNN network to electricity load forecasting. In *Theory and Applications of Time Series Analysis: Selected Contributions from ITISE 2019* (pp. 205-218). Springer International Publishing.

[8] Lang, C., Steinborn, F., Steffens, O., & Lang, E. W. (2019). Electricity Load Forecasting--An Evaluation of Simple 1D-CNN Network Structures. *arXiv preprint* arXiv:1911.11536.

[9] Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing, 151,* 107398.

[10] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv*:1412.6980.

[11] Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv*:1804.07612.

[12] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.

[13] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[14] Zhang, Z., Liu, Y., Gong, Y., & Lu, H. (2019). Forecasting air pollutant concentration levels with deep learning models. *Sustainability*, 11(7), 2041. https://doi.org/10.3390/su11072041

[15] Ritchie, H., Roser, M., & Rosado, P. (2020). CO_2 and Greenhouse Gas Emissions. *Our World in Data*. https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions

[16] Kiranyaz, S., Ince, T., & Gabbouj, M. (2021). 1D Convolutional Neural Networks and Applications: A Survey. *Mechanical Systems and Signal Processing*, 151, 107398. https://doi.org/10.1016/j.ymssp.2020.107398