Artificial Neural Networks for COVID-19 Forecasting in Mexico: An Empirical Study

C. M. Castorena[1], R. Alejo[2], E. Rendón[2], E. E. Granda-Gutíerrez[3], R. M. Valdovinos[4], and G. Miranda-Piña[2](✉)

[1] Tecnologías de la Información y las Comunicaciones, Universidad de Valencia, Av. Universitat s/n, 46100 Burjassot, Valencia, Spain
[2] Division of Postgraduate Studies and Research, National Institute of Technology of Mexico, (TecNM) Campus Toluca, Av. Tecnológico s/n, Agrícola Bellavista, 52149 Metepec, Mexico
MM22280266@toluca.tecnm.mx
[3] UAEM University Center at Atlacomulco, Universidad Autónoma del Estado de México, Carretera Toluca-Atlacomulco Km. 60, 50450 Atlacomulco, Mexico
[4] Faculty of Engineering, Universidad Autónoma del Estado de México, Cerro de Coatepec S/N, Ciudad Universitaria, 50100 Toluca, Mexico

Abstract. Artificial Neural Networks (ANN) have encountered interesting applications in forecasting several phenomena, and they have recently been applied in understanding the evolution of the novel coronavirus COVID-19 epidemic. Alone or together with other mathematical, dynamical, and statistical methods, ANN help to predict or model the transmission behavior at a global or regional level, thus providing valuable information for decision-makers. In this research, four typical ANN have been used to analyze the historical evolution of COVID-19 infections in Mexico: Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Long Short-Term Memory (LTSM) neural networks, and the hybrid approach LTSM-CNN. From the open-source data of the Resource Center at the John Hopkins University of Medicine, a comparison of the overall qualitative fitting behavior and the analysis of quantitative metrics were performed. Our investigation shows that LSTM-CNN achieves the best qualitative performance; however, the CNN model reports the best quantitative metrics achieving better results in terms of the Mean Squared Error and Mean Absolute Error. The latter indicates that the long-term learning of the hybrid LSTM-CNN method is not necessarily a critical aspect to forecast COVID-19 cases as the relevant information obtained from the features of data by the classical MLP or CNN.

Keywords: COVID-19 · Forecasting · Artificial Neural Networks · Deep learning

This work has been partially supported under grants of project 11437.21-P from TecNM and 6364/2021SF from UAEMex.

D.-S. Huang et al. (Eds.): ICIC 2022, LNCS 13393, pp. 168–179, 2022.
https://doi.org/10.1007/978-3-031-13870-6_14

1 Introduction

Artificial Neural Networks (ANN) have become a hot topic in artificial intelligence, particularly Deep Learning ANN (DL-ANN), which have been successfully employed in the classification of images, audio, and text, among others [9]. In addition, the DL-ANN have shown remarkable effectiveness in approximation functions, and prediction or forecast [10]. In the recent pandemic occasioned by coronavirus disease (COVID-19), DL-ANN have confirmed the ability to forecast COVID-19 cases. Ref. [28], presents a comparative study of five deep learning (DL) methods to forecast the number of new cases and recovered cases; the promising potential of a deep learning model in forecasting COVID-19 is demonstrated. Similarly, in [4] a comparative study of DL and machine learning models for COVID-19 transmission forecasting was performed; experimental results showed that the best performance was archived by DL, especially the LSTM-CNN model (which is the combination of Long Short-Term Memory-LSTM and Convolutional Neural Networks-CNN).

Much work has been performed to forecast COVID-19 cases in different regions, and countries [15]. For example, Ref. [1] reported the forecast results of COVID-19 cases (obtained by Recurrent ANN-LSTM and Recurrent ANN-GRU models) throughout 60-day in ten countries (USA, Brazil, India, Russia, South Africa, Mexico, Peru, Chile, United Kingdom, and Iran). Refs. [6, 12]) have followed the same direction, and they have shown experimental results of forecast covid-19 cases for multiple countries or populations. Also, specialized studies over particular countries have been developed; Ref. [23] proposes a recurrent and convolutional neural network model to forecast COVID-19 cases confirmed daily in 7, 14, and 21 days in India. Similarly, Ref. [21] studies statistical and artificial intelligence approaches to model and forecast the prevalence of this epidemic in Egypt.

In the Mexican context, some studies have been presented. In [22] cases of COVID-19 infection in Mexico are modeled and predicted through mathematical and computational models using only the confirmed cases provided by the daily technical report COVID-19 Mexico, from February 27th to May 8th, 2020. Ref. [11] uses ANN to predict the number of COVID-19 cases in Brazil and Mexico until December 12, 2020. In the same year, Ref. [18] presents an analysis of the ensemble of the neural network model with fuzzy response aggregation to predict COVID-19 confirmed cases of COVID-19 in 109 days ahead for Mexico (whole Country), which confirms other studies where ensemble method works better than monolithic classifiers, in this case on predicting the COVID-19 time series in Mexico. Another very interesting paper [8] compares traditional and powerful forecasting methods (vector autoregression and statistical curve-fitting methods) concerning DL techniques (in particular, the LSTM model) to identify the pandemic impact in Mexico in a period of 70 days (January 22 to March 31, 2020); it concludes that the best practice is to use LSTM over classical models.

In this paper, we present an empirical study of four popular ANN: Multilayer Perceptron (MLP), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and LSTM-CNN, to forecast COVID-19 cases in Mexico. These ANN have been reported in the state of art as the best models for this type of goal. We use recent COVID-19 data (from February 22, 2020, to April 4, 2022); therefore, we now have more information about COVID-19 than in previous works, suggesting an improvement in the forecasting of COVID-19 cases. In this sense, this study intends to be conceived as a basis for comparing deep learning techniques in the context of similar problems, characterized by being highly sensitive to data variability in applications other than classical classification tasks. Results indicate that LSTM-CNN achieves the best qualitative performance, but the CNN model reports the best quantitative metrics.

2 Theoretical Framework

Feed-forward constitutes the most conventional ANN architecture. It is commonly based on at least three layers: input, output, and one hidden layer [16]. In the DL context, feed-forward DL-ANNs have two or more hidden layers in their architecture. This allows to reduce the number of nodes per layer and uses fewer parameters, but it leads to a more complex optimization problem [9]. However, this disadvantage is less restrictive than before due to the availability of more efficient frameworks, such as Apache-Spark or TensorFlow (which use novel technology like GPU or Cluster Computing). Another important DL model is recurrent neural networks, which are a type of network for sequential data processing, allowing to scale of very long and variable-length sequences [27]. In this type of network, a neuron is connected to the neurons of the next layer, to those of the previous layer, and to itself using weights (values that change in each time step). A summary of the DL models studied in this work is featured below.

2.1 MLP

MLP is a classical ANN with one input and one output layer, and at least one hidden layer, trained with a different set of features based on the previous layer's output (see Fig. 1). It is possible to select features in a first layer, and the output of this will be used in the training of the next layer [16]. The number of inputs and outputs of the problem to be solved is the factor that will determine the number of neurons in the input and output layers, and every neuron can feed into the next neuron of the next layer by repeating

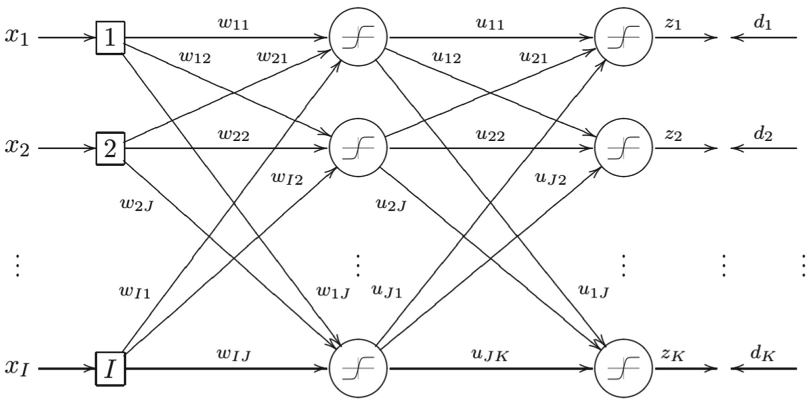

Fig. 1. Classical MLP architecture comprised by 3 layers, *I* input nodes, *J* hidden nodes and *K* output nodes.

the process from the input until the output layer [13]. An MLP structure can achieve significant performance in small models sizes, but when its size scales up, the model is affected by the over-fitting [31]. MLPs could approximate any continuous function and can solve not linearly separable problems.

2.2 CNN

CNN is a deep neural network architecture that combines the multilayer perceptron with a convolutional layer to build a map that has the function of extracting important features (see Fig. 2). Furthermore, it implements a pooling stage to reduce the dimensionality of the features and save the most informative features [3]. The main idea behind these models is that abstract features can be extracted by the convolutional layers and the pooling operation, where the convolutional kernels convolve local filters with sequential data without processing and produce non-variant local features, and the subsequent pooling layers will extract the essential features within fixed-length sliding windows [30], in other words, a CNN is a powerful extractor that applies convolution on multiple blocks to extract meaningful features [25]. CNN models have shown to be effective in problems related to modeling image data, summarization, and classification [14].

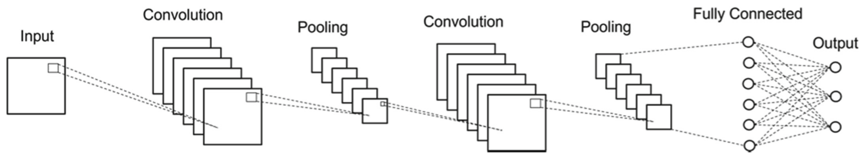

Fig. 2. A CNN architecture comprised by two convolutional layers and two pooling layers.

2.3 LSTM

LSTM network is a particular type of recurrent neural network that can learn in the long term to avoid dependency [19]. To achieve this, LSTM uses different cells to allow actions such as “forget” and “remember” [3]; in other words, LSTM units consist of elements such as an input gate, a forget gate, a memory cell, and an output gate [30] (see Fig. 3). It is important to say that LSTM was designed to prevent the backpropagating error from disappearing or exploding; likewise, forget gates were included to achieve long-term non-dependence, being able to control the use of state cell information [29]. These architectures were designed to work with data in constant times that occur between elements of a given sequence [2]. Due to its ability to capture long-range dependencies, this model has been successfully applied in many areas, such as speech recognition, handwriting recognition, image recognition, and natural language processing [29].

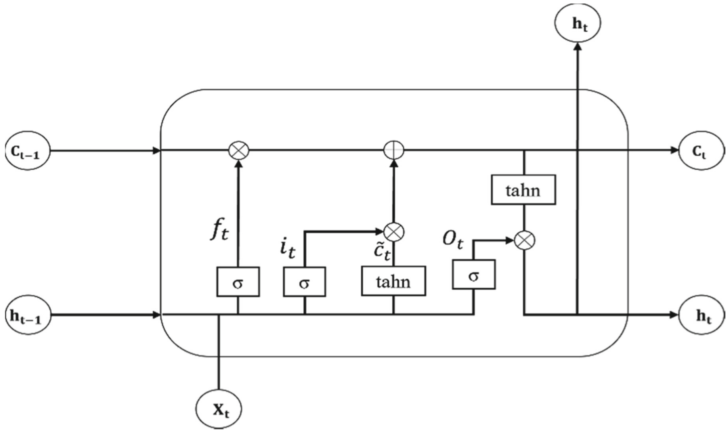

Fig. 3. Architecture of an LSTM unit with a forget gate (f_t), current input (X_t), memory cell (C_i) and output (h_t).

2.4 LSTM-CNN

In recent years, hybrid deep learning architectures have been applied to different tasks showing better results than the baseline models. A clear example of these hybrid frameworks are the LSTM and CNN architectures which have shown excellent performances in tasks such as time series classification, video recognition, and text classification due to their unique characteristics [24]. Combining these networks, the advantages of each one is merged to achieve more significant results [26]. The core idea behind the fusing of these models is that CNN can extract time-invariant elements, and LSTM can learn long-term dependency. Both LSTM and CNN receive the same data input, and then the results are concatenated to get the output [17] (see Fig. 4). Therefore, with this fusion, a better structure, and more complete spatial and temporal characteristics can be obtained, improving the results in the state of the art [7].

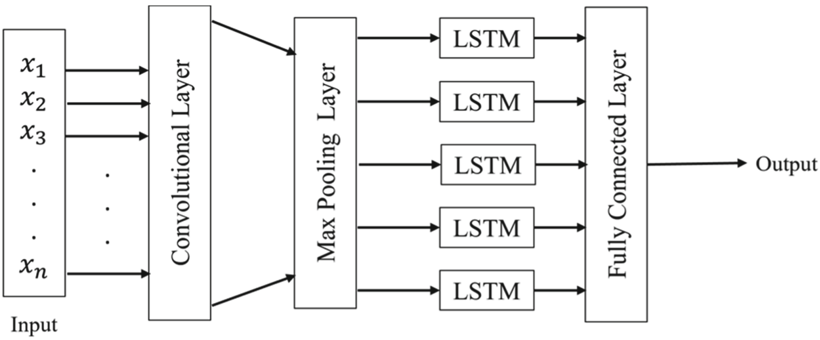

Fig. 4. Architecture of the LSTM-CNN hybrid model comprised by a convolutional layer and a LSTM block.

Traditionally, ANNs have been trained with the back-propagation algorithm (based on the stochastic gradient descent), and the weights are randomly initialized. However, in some late versions of DL-ANN, the hidden layers are pre-trained by an unsupervised algorithm, and the weights are optimized by the back-propagation algorithm [16] or methods based on the descending gradient. To overcome it, the classical Sigmoid activation function has been replaced (commonly) by other functions like Rectified Linear Unit (ReLU) $f(z) = \max(0, z)$, Exponential Linear Unit (ELU $= zif \langle z \geq 0 \rangle \; else \; \langle \alpha * (e^z - 1) \}\rangle$), or softmax ($\varphi(s) = e^{s_i} / \sum_j^C e^{s_j}$), that is associated to the output layer), because typically they learn much faster in networks with many layers, allowing training of a DL-ANN without unsupervised pre-training [9].

The most common algorithms of descending gradient optimization are: a) Adagrad, which adapts the learning reason of the parameters, making more significant updates for less frequent parameters and smaller for the most frequent ones, b) Adadelta is an extension of Adagrad that seeks to reduce aggressiveness, monotonously decreasing the learning rate instead of accumulating all the previous descending gradients, restricting accumulation to a fixed size, and c) Adam, that calculates adaptations of the learning rate for each parameter and stores an exponentially decreasing average of past gradients. Other important algorithms are AdaMax, Nadam, and RMSprop [20].

3 Experimental Set Up

In this section, the experimental details to allow the proper replication of the results of this research and to support the conclusions, are described.

3.1 Dataset

The dataset was extracted from the GitHub repository (https://github.com/CSSEGISandData/COVID-19) from the Resource Center at the John Hopkins University of Medicine, which is updated daily at 9 am EST [5]. It is important to say that only the file of the confirmed cases of COVID-19 was downloaded, and then only the dates and their cases reported from February 22, 2020, to April 4, 2022, for Mexico were extracted. After data collection, the data was organized by week (considering seven dates to form a week), and then the average per week was obtained to work with a smooth curve. In addition, the dataset ($D = \{x_1, x_2, \ldots, x_Q\}$) was split in two disjoints sets, one (TR) to train the model and other (TS) to test its generalized ability, i.e., $D = TR \cup TS$; $TR \cap TS = \varnothing$; and (TR) contains 70% of the samples of D, and TS contains the remaining 30%. Thus, each individual x_q data in D corresponds to the average number of active COVID-19 cases in Mexico in one specific week, i.e., the average number of cases every seven days.

3.2 Free Parameters Specification

The DL model to use is an MLP, and it was designed with an input layer of 5 nodes and 1 hidden layer with 3 nodes, both with the RELU activation and the output layer with linear activation. CNN topologies consist of two CNN layers, the first with 16 filters (or kernels) of dimension 4×4 and the second with 32 filters of 1×1. A pooling layer of size

2 (using the MaxPooling method), and a dense layer of 33 nodes, were used. The LSTM model has a sequence length and an input dimension of 5 and 16 with RELU activation and recurrent sigmoid activation. In addition, the dropout method is applied after the input layer and before another LSTM layer with 64 nodes and a recurrent sigmoid layer; finally, a linear activation for the output layer is used. LSTM-CNN contains an LSTM layer with sequence length, an input dimension of 4 and 64, respectively, a CNN layer with 32 filters of 4 × 4 and a RELU activation, and finally, a dense layer with 1 hidden neuron and a linear activation function.

3.3 Performance of the DL Models

Mean Squared Error(MSE) and Mean Absolute Error (MAE) are metrics widely accepted to assess ANN in the prediction and approximation of functions [9]. In this work, we use both, MSE and MAE (Eq. 1) to test the effectiveness of studied DL models.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(t_i - z_i)^2; \qquad MAE = \frac{1}{N}\sum_{i=1}^{N}||t_i - z_i||; \tag{1}$$

where N is the total of samples, t_i is the desired output, and z_i is the actual or predicted output of the ANN for the sample i.

DL models were developed in Tensorflow 2.0 and Keras 2.3.1 frameworks. Adam was selected as the optimizer method with a batch size of 9 for CNN, 10 for LSTM, and 1 for MLP and LSTM-CNN. The stop criterion was 500 epochs.

4 Results and Discussion

The main results obtained in the experimental stage are presented and discussed in this section. Figures 5, 6, 7 and 8 show the graphs generated by the four ANN studied in this work (CNN, MLP, LSTM and LSTM-CNN, see Sect. 2). Axis x represents the analyzed time periods (in this work, it corresponds to an interval of a week, for more detail see Sect. 3.1), and axis y corresponds to forecast and real COVID-19 cases.

From a qualitative viewpoint, experimental results seem to note that the best performance corresponds to CNN (Fig. 5), where both: real and predicted values, are very similar. It matches with quantitative results, where MSE and MAE values of CNN are the smallest. However, considering that the dataset is small, this behavior may imply that overfitting could be occurring. In MLP (Fig. 6), this behavior is more evident; we observe smaller MSE and MAE in the training data (blue) than the obtained with test data (red); nevertheless, MLP follows the data trend from the qualitative viewpoint.

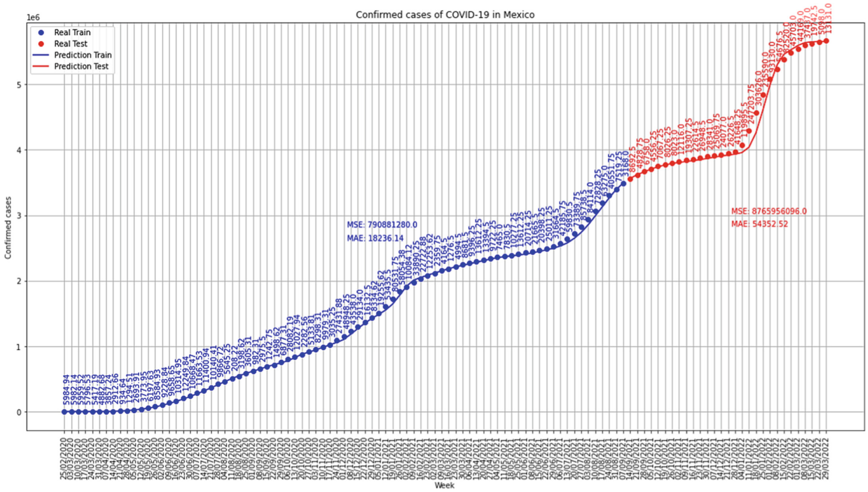

Fig. 5. Results obtained by the CNN model.

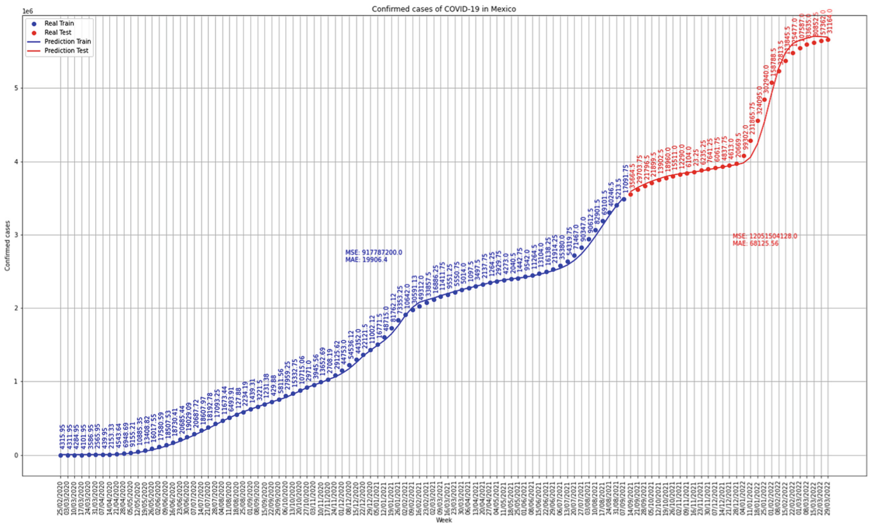

Fig. 6. Results obtained by the MLP model.

LSTM model exhibits a similar trend to MLP for the training dataset, but the behavior of LSTM in the test data does not match the actual data. It is reflected in the MAE and MSE values test, which are more significant than the MLP case. LSTM is characterized by long-term learning dependency. Thus, results presented in Fig. 7 do not notice that this feature of LSTM (by itself) is enough to approximate COVID-19 data with good performance.

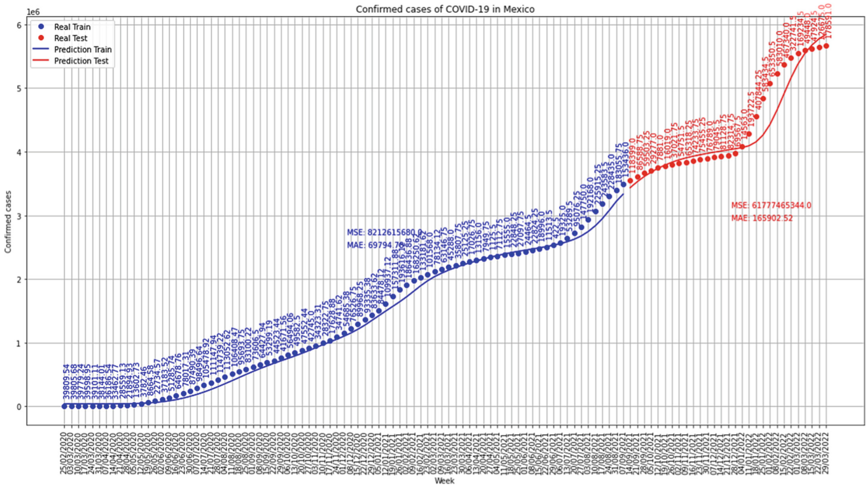

Fig. 7. Results obtained by the LSTM model.

LSTM-CNN shows a better performance from a qualitative viewpoint, as can be seen in Fig. 8 where, although the model does not entirely fit in the training and testing datasets, LSTM-CNN has the best generalization ability, which refers to the capability of the model to give an appropriate answer to unlearned questions. The generalization performance of LSTM-CNN could be explained by the combination of the long-term learning dependency of the LSTM model and the CNN's capacity to exploit the information extracted from the data.

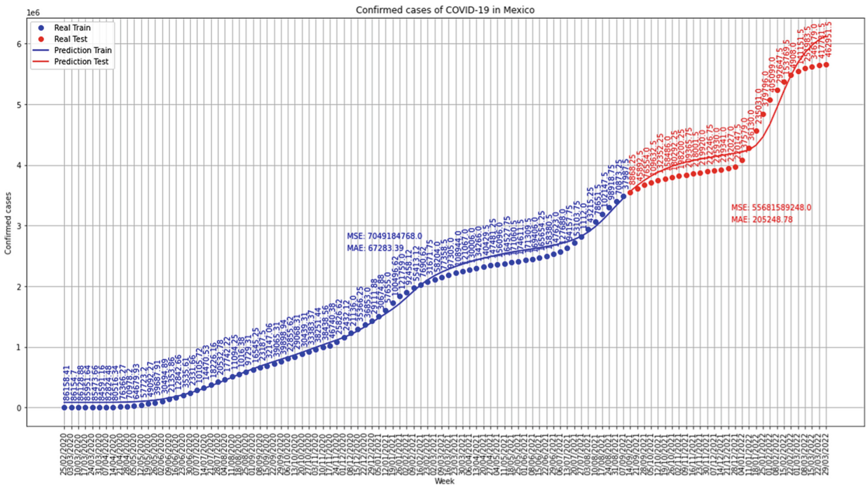

Fig. 8. Results obtained by the LSTM-CNN model.

Finally, in the recent state-of-art about the COVID-19 forecasting, the LSTM and LSTM-CNN appeared to report the better behavior, even overcoming the performance of the other models. However, the results presented in this work (which use more information about confirmed COVID-19 cases in Mexico than previous works) show that from a quantitative viewpoint, MLP and CNN models obtain smaller values of MSE and MAE than LSTM and LSTM-CNN. The best data fit performance is presented by the CNN model (Fig. 5), which would suggest that ability to long-term learning dependency data is not a critical aspect to forecast COVID-19 cases. Thus, from a quantitative viewpoint, results showed by MLP and CNN imply the best performance when the most relevant information is extracted from the data; MLP, however, it is considered that the performance of the CNN is attained due it has a better capability to transform the abstract feature space in the convolutional layers inherent to its architecture. This behavior could be explained by the fact that deep neural network potential is found in its hidden space, where it can abstract high-level patterns. In such hidden space, original data are transformed to another multi-dimensional space, in which the decision boundary could be identified with a higher degree of reliability [9, 16].

5 Conclusion

In this paper, we studied four ANN models: MLP, CNN, LSTM, and LSTM-CNN, to forecast COVID-19 cases. Experimental results analyzed from a qualitative viewpoint suggest that the LSTM-CNN model obtains the best performance. However, from the quantitative perspective, the CNN model overcomes the performance of MLP, LSTM, and LSTM-CNN. These results indicate that long-term learning dependency data is not critical to forecasting COVID-19 cases (see results of LSTM-CNN, but mainly of the LSTM model). Instead, the results exhibited by MLP and CNN imply that to obtain better performance is most important to extract relevant information from data features, which is a highlighted feature of MLP and CNN models.

The results presented in this work are exciting; nevertheless, future work is required to deep into this study and to develop a theoretical explanation for the experimental results due to the potential of the analyzed models to forecast COVID-19 cases in regions like Mexico or Latin America, which have been seriously affected by this pandemic.

References

1. ArunKumar, K., Kalaga, D.V., Kumar, C.M.S., Kawaji, M., Brenza, T.M.: Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUS) and long short-term memory (LSTM) cells. Chaos Solitons Fractals **146**, 110861 (2021). https://doi.org/10.1016/j.chaos.2021.110861
2. Baytas, I.M., Xiao, C., Zhang, X., Wang, F., Jain, A.K., Zhou, J.: Patient subtyping via time-aware LSTM networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 65–74. Association for Computing Machinery, New York (2017). https://doi.org/10.1145/3097983.3097997
3. Bengfort, B., Bilbro, R., Ojeda, T.: Applied Text Analysis with Python. O'Reilly Media, Inc., Sebastopol (2018)

4. Dairi, A., Harrou, F., Zeroual, A., Hittawe, M.M., Sun, Y.: Comparative study of machine learning methods for COVID-19 transmission forecasting. J. Biomed. Inform. **118**, 103791 (2021). https://doi.org/10.1016/j.jbi.2021.103791
5. Dong, E., Du, H., Gardner, L.: An interactive web-based dashboard to track covid-19 in real time. Lancet Inf. Dis. **20**(5), 533–534 (2020). https://doi.org/10.1016/S1473-3099(20)30120-1
6. Fanelli, D., Piazza, F.: Analysis and forecast of COVID-19 spreading in China, Italy and France. Chaos Solitons Fractals **134**, 109761 (2020). https://doi.org/10.1016/j.chaos.2020.109761
7. Gao, J., Gu, P., Ren, Q., Zhang, J., Song, X.: Abnormal gait recognition algorithm based on LSTM-CNN fusion network. IEEE Access **7**, 163180–163190 (2019). https://doi.org/10.1109/ACCESS.2019.2950254
8. Gomez-Cravioto, D.A., Diaz-Ramos, R.E., Cantu-Ortiz, F.J., Ceballos, H.G.: Data analysis and forecasting of the COVID-19 spread: a comparison of recurrent neural networks and time series models. Cogn. Comput., 1–12 (2021). https://doi.org/10.1007/s12559-021-09885-y
9. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
10. Guo, M., Manzoni, A., Amendt, M., Conti, P., Hesthaven, J.S.: Multi-fidelity regression using artificial neural networks: efficient approximation of parameter-dependent output quantities. Comput. Methods Appl. Mech. Eng. **389**, 114378 (2022). https://doi.org/10.1016/j.cma.2021.114378
11. Hamadneh, N.N., Tahir, M., Khan, W.A.: Using artificial neural network with prey predator algorithm for prediction of the COVID-19: the case of Brazil and Mexico. Mathematics **9**(2) (2021). https://doi.org/10.3390/math9020180
12. Hamdy, M., Zain, Z.M., Alturki, N.M.: COVID-19 pandemic forecasting using CNN-LSTM: a hybrid approach. J. Control Sci. Eng. **2021**, 8785636 (2021). https://doi.org/10.1155/2021/8785636
13. Kahani, M., Ahmadi, M.H., Tatar, A., Sadeghzadeh, M.: Development of multilayer perceptron artificial neural network (MLP-ANN) and least square support vector machine (LSSVM) models to predict Nusselt number and pressure drop of TIO2/water nanofluid flows through non-straight pathways. Numer. Heat Transf. Part A Appl. **74**(4), 1190–1206 (2018). https://doi.org/10.1080/10407782.2018.1523597
14. Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S.: Review on convolutional neural networks (cnn) in vegetation remote sensing. ISPRS J. Photogramm. Remote Sens. **173**, 24–49 (2021). https://doi.org/10.1016/j.isprsjprs.2020.12.010
15. Kuvvetli, Y., Deveci, M., Paksoy, T., Garg, H.: A predictive analytics model for COVID-19 pandemic using artificial neural networks. Decis. Anal. J. **1**, 100007 (2021). https://doi.org/10.1016/j.dajour.2021.100007
16. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015). https://doi.org/10.1038/nature14539
17. Li, P., Abdel-Aty, M., Yuan, J.: Real-time crash risk prediction on arterials based on LSTM-CNN. Accid. Anal. Prev. **135**, 105371 (2020). https://doi.org/10.1016/j.aap.2019.105371
18. Melin, P., Monica, J.C., Sanchez, D., Castillo, O.: Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: the case of Mexico. Healthcare **8**(2) (2020). https://doi.org/10.3390/healthcare8020181
19. Muzaffar, S., Afshari, A.: Short-term load forecasts using LSTM networks. Energy Procedia **158**, 2922–2927 (2019). https://doi.org/10.1016/j.egypro.2019.01.952. Innovative Solutions for Energy Transitions
20. Ruder, S.: An overview of gradient descent optimization algorithms. CoRR abs/1609.04747 (2016). http://arxiv.org/abs/1609.04747

21. Saba, A.I., Elsheikh, A.H.: Forecasting the prevalence of COVID-19 outbreak in Egypt using nonlinear autoregressive artificial neural networks. Process Saf. Environ. Prot. **141**, 1–8 (2020). https://doi.org/10.1016/j.psep.2020.05.029
22. Torrealba-Rodriguez, O., Conde-Gutiérrez, R., Hernández-Javier, A.: Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models. Chaos Solitons Fractals **138**, 109946 (2020). https://doi.org/10.1016/j.chaos.2020.109946
23. Verma, H., Mandal, S., Gupta, A.: Temporal deep learning architecture for prediction of COVID-19 cases in India (2021)
24. Vo, Q.H., Nguyen, H.T., Le, B., Nguyen, M.L.: Multi-channel LSTM-CNN model for Vietnamese sentiment analysis. In: 2017 9th International Conference on Knowledge and Systems Engineering (KSE), pp. 24–29 (2017). https://doi.org/10.1109/KSE.2017.8119429
25. Wu, J.L., He, Y., Yu, L.C., Lai, K.R.: Identifying emotion labels from psychiatric social texts using a bi-directional LSTM-CNN model. IEEE Access **8**, 66638–66646 (2020). https://doi.org/10.1109/ACCESS.2020.2985228
26. Yan, R., Liao, J., Yang, J., Sun, W., Nong, M., Li, F.: Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. Expert Syst. Appl. **169**, 114513 (2021). https://doi.org/10.1016/j.eswa.2020.114513
27. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019). https://doi.org/10.1162/neco_a_01199
28. Zeroual, A., Harrou, F., Dairi, A., Sun, Y.: Deep learning methods for forecasting COVID-19 time-series data: a comparative study. Chaos Solitons Fractals **140**, 110121 (2020). https://doi.org/10.1016/j.chaos.2020.110121
29. Zhao, R., Wang, J., Yan, R., Mao, K.: Machine health monitoring with LSTM networks. In: 2016 10th International Conference on Sensing Technology (ICST), pp. 1–6 (2016). https://doi.org/10.1109/ICSensT.2016.7796266
30. Zhao, R., Yan, R., Wang, J., Mao, K.: Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors **17**(2) (2017). https://doi.org/10.3390/s17020273
31. Zhao, Y., Wang, G., Tang, C., Luo, C., Zeng, W., Zha, Z.: A battle of network structures: an empirical study of CNN, transformer, and MLP. CoRR abs/2108.13002 (2021). http://arxiv.org/abs/2108.13002