UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

Predicting time series with LSTM and CNN deep learning models. Application on a COVID-19 database

Felipe Esteban Puente Cárdenas

Ingeniería en Ciencias de la Computación

Trabajo de fin de carrera presentado como requisito
para la obtención del título de
Ingeniero

Quito, 20 de diciembre de 2022

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

HOJA DE CALIFICACIÓN
DE TRABAJO DE FIN DE CARRERA

Predicting time series with LSTM and CNN deep learning models. Application on a COVID-19 database

Felipe Esteban Puente Cárdenas

Nombre del profesor, Título académico **Noel Pérez, Ph. D**

Quito, 20 de diciembre de 2022

© DERECHOS DE AUTOR

Nombres y apellidos:	Felipe Esteban Puente Cárdenas
Código:	00207784
Cédula de identidad:	1718898776
Lugar y fecha:	Quito, 20 de diciembre de 2022

ACLARACIÓN PARA PUBLICACIÓN

UNPUBLISHED DOCUMENT

RESUMEN

La pandemia del virus COVID-19 ha tenido un enorme impacto en la sociedad. Millones de personas se vieron afectadas por el virus SARSCOV-2, lo que tuvo terribles consecuencias en los ámbitos económico y sociales en todo el mundo. Para hacer frente a este problema

hay que aplicar las medidas necesarias en el momento oportuno, y para ello se necesita una previsión eficaz. En este contexto, este trabajo pretende utilizar técnicas de aprendizaje profundo, es decir, memoria a corto plazo (LSTM) y redes neuronales convolucionales (CNN) con el fin de predecir el número de casos confirmados de COVID-19. En Se compara el rendimiento de las arquitecturas CNN y LSTM para predecir el número de infectados a un día y a siete días en el futuro. En el experimento de este estudio, los indicadores de error medio porcentual absoluto y error cuadrático medio fueron utilizados para examinar la eficacia global de ambas arquitecturas. Los resultados indican que el modelo CNN propuesto en este estudio supera el de la LSTM, mostrando una mayor eficacia de predicción, obteniendo una puntuación MAPE promedio de 0,91 y de 4.85 para la predicción de un día y siete días respectivamente. Finalmente se demuestra que el uso de las arquitecturas LSTM y CNN es adecuado para tareas de predicción.

Palabras clave: LSTM, CNN, COVID-19, predicción, series de tiempo, aprendizaje profundo.

ABSTRACT

The COVID-19 virus pandemic has had a huge impact on society. Millions of people were affected by the SARSCOV-2 virus, and this had terrible consequences in economic and social fields worldwide. To deal with this issue, necessary measures need to be implemented at the right moment, and for this purpose, efficient forecasting is needed. In this context, this paper aims to use deep learning techniques, i.e., long short-term memory (LSTM) and convolutional neural networks (CNN) to predict the number of COVID-19 confirmed cases. It compares the performance of both CNN and LSTM architectures in forecasting the number of infected one day and seven days into the future. In this study's experiment, the indicators of mean absolute percentage error and root mean square error were used to examine the overall efficacies of several methods. Findings indicate that the CNN model proposed in this study outperforms that of the LSTM, showing a better prediction efficacy, obtaining a mean MAPE score of 0.91 and 4.85 for one day and seven-day prediction respectively, by applying 10-fold time series split. Results show that using both long-short term memory and convolutional neural networks is appropriate for forecasting tasks.

Key words: LSTM, CNN, COVID-19, prediction, time-series, deep learning

TABLE OF CONTENTS

INDEX OF TABLES

INDEX OF FIGURES

INTRODUCTION

In recent years, the world's population has been affected by the SARS-COV-2 virus global pandemic [1]. The rapid spread of this virus meant that efforts to counteract it did not yield the expected results, thus having more than one hundred million seven hundred thousand confirmed cases in little more than two years [2]. At the beginning of the pandemic, the detection of the presence of the virus in people was a necessity in order to monitor the status and evolution of the virus. For this purpose, there were three different tests: polymerase chain reaction (PCR), antigen detection, and antibody detection. PCR detects the presence of COVID19 RNA (ribonucleic acid) and yields a result. The antigen detection can detect proteins from the nucleocapsid N or of the S1 and S2 spikes and thus define the infectivity of a person. Finally, in the case of antibody detection, the presence of immunoglobulin G (IgG) and immunoglobulin M (IgM) in a blood sample, serum, or plasma, determines whether a person has or recently had the virus [3]. Having virus detection tests available helps to determine the number of infected and, thus, the current evolution of the pandemic. However, predicting the pandemic evolution in the future could help to counteract it by implementing the necessary actions. Some studies based on statistical and mathematical models have played an important role in predicting the spread of diseases and epidemiological situations [4]. One of the most relevant methods is time series analysis. It analyzes a sequence of data collected over a specific time interval in the past, offering a relatively accurate prediction in the short run [5]. Currently, machine learning techniques have been gaining popularity in this area [6]. They use historical data to learn the stochastic dependence between the past and the present. That means learning the behavior of the time series in the past to predict events in the future [7]. For example, artificial neural networks (ANN), which have been proven to perform better in classical statistical methods, can be applied to perform time series predictions [6], [8]. Among several types of ANNs, long short-term memory (LSTM) and one-dimensional convolutional neural networks (CNN) appear as the most practical architectures, the LSTM being the most popular with increased use in time series forecasting, reaching successful results. In, [9], was combined an LSTM model with PSO (particle swarm optimization) for predicting oil production during the exploitation phase of the oilfield, achieving an RMSE (root mean squared error) score of 2.02. In [10], they used LSTM with dropout to predict water table depth in agricultural areas, reaching an RMSE score of 0.142. In [11], was used a stateless variation of the LSTM to predict the stock value in the Indian stock market, achieving an RMSE score of 5.897. Moreover, traffic prediction was forecasted using a variation of LSTM with connectivity layers, reaching an RMSE value of 0.047 [12]. Additionally, a cascaded LSTM has been used to fill incomplete logs by [13], having an RMSE value of 0.75. In [14], LSTM models for COVID19 prediction were used, obtaining sMAPE (symmetric mean absolute percentage error) score of 0.116 in forecasting the spread of the virus in Hubei, China. On the other hand, CNNs are commonly used for image processing [15]–[17], and speech recognition [18] in classification tasks. The use of CNN in time series prediction is limited, although some works related these models to time series forecasting. For example, in [19], was implemented a CNN model for time series classification, obtaining a classification accuracy score of 99.7 on the Wafer data set. Additionally, some CNN variations were implemented for energy time series forecasting, reaching an RMSE value of 2392.88 on a Spanish electricity data set [20]. Furthermore, in [21], a CNN's performance was analyzed with conditioning on the SP500 data set. It reached a MASE (mean absolute squared error) score of 0.699. In [22], was predicted COVID19 number of cases using a CNN with dropout layer, obtaining an RMSE score of 109.439. Nonetheless, they concluded that the limited data was a limitation that could be improved in a later investigation. Similarly, in [23], they concluded that using a reduced number of COVID-19 cases for forecasting tasks limited the performance of developed algorithms. Thus, improvement in terms of databases, predicting models, and final decisions are still challenging.

In this sense, we propose developing forecasting models based on deep learning architectures to predict inflection points (increase or decrease) of COVID-19 infected. The main contribution behind the goal is related to improving the baseline methods based on CNN and LSTM architectures to achieve satisfactory results.

In addition, we hope that this investigation and development outcomes provide a clear path to the virus' propagation in the future. It is worth mentioning that the ISO/IEC TS 4213:2022 standard was used in order to ensure the relevance, legitimacy and extensibility of machine learning classification performance assertions [24]. Engineering Design Process [25] was considered as a guide for the design of this experiment, from the information gathering to the functional structure, implementation and development of this project.

MATERIALS AND METHODS

A. *Database*

This work considered using the publicly available database from the Novel Corona Virus (COVID-19) Cases Data [26]. This database was created and is administrated by Johns Hopkins University Center for Systems Science and Engineering (JHU CCSE). They compiled information from diverse sources, including the World Health Organization and various health institutes around the world. JHU CCSE started the data collection on January 22 of 2020, and the fields available in the database include Province/State, Country/Region, Last Update, Confirmed, Suspected, Recovered, and Deaths. The number of collected cases is variable in the time since the database is constantly updated

B. Deep learning models

Deep learning is making significant strides in resolving issues that have long defied the best efforts of the artificial intelligence field. Deep learning expands on traditional machine learning by incorporating more ”depth” and complexity into the models and modifying the data using various functions that enable hierarchical data representation through several degrees of abstraction [27]. These models can solve complex problems exceptionally well and fast, as they allow massive parallelization [28]. As a consequence, they have a large learning capacity, which allows them to solve classification and prediction problems particularly well [29]. In addition to breaking records in speech and image recognition [15], [30], it has outperformed competing machine learning methods in tasks including analyzing brain circuits [31], interpreting particle accelerator data [32], and predicting the consequences of non-coding DNA mutations on gene expression and disease [33]. Deep learning has demonstrated incredibly promising outcomes for a variety of tasks in natural language understanding [34], including subject classification, sentiment analysis, question answering, language translation, and for time series forecasting as well [35]. Two deep learning architectures are going to be used in this experiment: a bidirectional long short-term memory (BiLSTM) and a convolutional neural network..

The LSTM network is a deep learning architecture derived from the recurrent neural network (RNN) that can learn order dependence in sequence prediction thanks to its ability to store information over an extended period of time, influencing the input and output connection [36]. It was designed to overcome vanishing gradient problems caused by learning long-term dependencies [37]. The base architecture of a modern LSTM is composed of a cell and an input gate, output gate, and forget gate, enabling the network to reboot its state. The three gates control the flow of information related to the cell, and the cell remembers values across arbitrary time intervals [38]. There are several variations of this architecture created by adding or removing layers and manipulating the network's components and parameters. One of the more interesting variations is the bidirectional LSTM or BiLSTM, which has outperformed the basic LSTM in areas like multiple speech disfluencies detection [39]. The BiLSTM provides an additional training capability, as the output layer of the network obtains information both

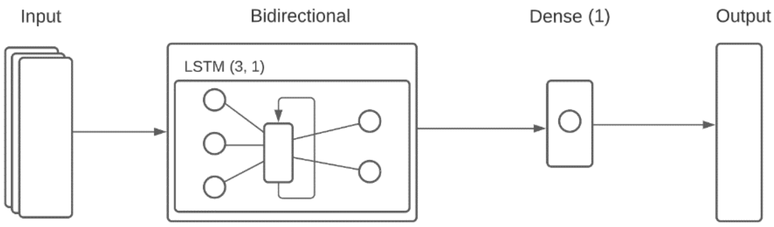


Figure 1: Architecture of the proposed LSTM model

backward and forward simultaneously [40]. This provides better prediction accuracy than the regular (unidirectional) LSTM architecture [41]. Convolutional neural networks are a type of feedforward neural network that is capable of extracting features from data using convolution structures [42]. The convolution and pooling operations introduced in this architecture generate deep features that improve pattern recognition and are the heart of the CNN [43]. An important part of the convolution operation is related to the concept of affine transformations, in which a vector is received as an input and is multiplied with a matrix to create an output. This can be applied to any input type, as its representation can always be transformed to a vector [44]. So convolution starts with this input and with a kernel or filter that ”slides” across the input, and, at each location, the product between the kernel and input element is computed to obtain the current location’s output [45]. Convolution happens in the convolutional layer of the network, where local conjunctions of *features* from previous layers can be detected [35], extracting features and patterns useful for classification problems. Although its use in the area of image processing is the most popular, it has been shown that this neural network model can also be used for prediction, and the neural network model can also be used for time series prediction, and classification [19].

C. Proposed method

We propose using two different deep-learning architectures to perform the COVID-19 time series prediction task. The first model implements a bidirectional LSTM or BiLSTM architecture followed by fully connected and output layers, respectively. We are working with

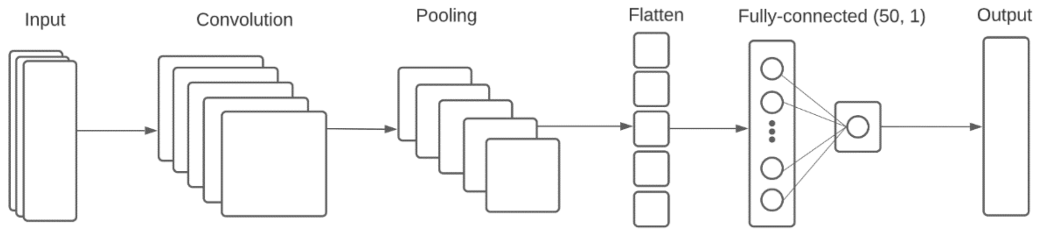


Figure 2: Architecture of the proposed CNN model

a 3-time step input, and because we are working with a univariate series (working with only one data source), the number of cases is the only feature. The bidirectional-LSTM layer was created by wrapping an LSTM layer with 50 LSTM units inside a Bidirectional wrapper layer. This allows the LSTM layer has the advantage of forward and backward simultaneous readings of the BiLSTM. Finally, a fully connected layer was added to the model, and a single numeric prediction value was obtained as an output. A representation of the aforementioned architecture is shown in Figure 1. Two instances of the model are created: LSTM1, for the one-day forecasting and LSTM7 for the seven-day forecasting part of this work. The second model is a one-dimensional CNN or 1DCNN. It is going to be referred to as CNN throughout this work and implements a convolution layer, followed by a pooling layer, a flattened layer, a fully connected layer made up of two dense layers, and the output layer. In this case, the 3- time step and single feature shape of input are repeated from the LSTM model. The convolutional layer contains 64 filter maps and a kernel size of 2 × 2. This layer is followed by a max pooling layer that distills the output of the previous layer to its most important components. Then, the feature maps are condensed to a single one-dimensional vector using a flattened layer placed before the fully-connected layer. The model implements a couple of dense layers that interpret the input feature. As this is a one-dimensional CNN, the data is handled like a one-dimensional image, as a sequence over which convolutional read operations can be carried out. We will have CNN1 and CNN7 for the two different predictions, as with the LSTM1 and LSTM7 instances. A representation of the described CNN model is shown in Figure 2.

D. Experimental setup

1) *Data processing and data set creation:* The original database contains the data distributed by some features such as Province/State, Country/ Region/ Latitude, and Longitude that are irrelevant to the purpose of this work. Therefore, they are removed to avoid noise and to focus the model learning process on predicting the number of infected cases worldwide instead of by countries/states/regions, etc. In this sense, we framed all cases in a single data set, reaching a total number of 1014 data points. Each point represents one day from January 22, 2020, to October 31, 2022. Processing the data in this way helps us to increment the number of cases for the training task. Since we expect to make predictions one and seven days after, it was necessary to create two experimental time series data sets called oneDayData and sevenDayData using the single-step forecasting technique [45], which obtains the observation at the next time step, hence only one-time step is to be predicted from the input, and has demonstrated to be more useful than the multi-step forecasting technique [46]. The oneDayData data set constructs 1014 data points. On the other hand, the sevenDayData data set takes from the original data points sequence, all the points with a seven-day apart, resulting in 144 data points.
2) *Training, validation and test sets:* As data pre-processing, we first divided a sample of 100 data points for the oneDay prediction models and 20 data points for the sevenDay prediction models. This left the oneDay and sevenDay models with 914 and 124 data points, respectively. In order to test the model's performance, we utilized a variation of the traditional cross-validation called time series split with **k = 10** [47], which is frequently used in time series forecasting problems. Similar to the random split methods, the goal of the time series split is to verify the model's predictability regardless of how the different data sets are divided. Nevertheless, as we won't be able

to train on ”future” data, the time series split makes sure the validation data sets are either more recent or older than the train data sets, which is more realistic.

3) *Model configuration:* For all models, CNN and LSTM, there were many similar hyperparameters, so a general model will be discussed to explain them. To begin with, the base model was set up with an Adam optimizer, used to update the weights and bias [9]. The learning rate used with the optimizer was set to 0.0001 samples. For both models created to predict the number of cases a single day into the future, LSTM1 and CNN1, the models were trained for 2000 epochs. For the remaining models, LSTM7 and CNN7, the number of epochs was increased to 2500 epochs due to the fact that the amount of available data is lower (as a result of the pre-processing process), and we seek to compensate this fact increasing the number of epochs.

4) *Assessment metrics:* The models were assessed using test data for each training instance in order to forecast the number of cases in the future and to calculate the mean absolute percentage error (MAPE), RMSE, and mean absolute error (MAE) [9]. Firstly, mean absolute percentage error is the ratio between the absolute difference between the projected and true values, divided by the true values. The root mean squared error is an extension of the mean squared error (MSE), or average of the error squares. Finally, the mean absolute error (MAE) is the average of the absolute difference between actual and predicted values. The model’s performance was evaluated using the mean absolute percentage error, and the one with the slowest value was selected as the best one. We consider using the MAPE metric instead of MAE and RMSE due to its practical interpretation of obtained results in the same scale. Because the number of infected are of a large magnitude (millions), the other metrics are no longer appropriate for evaluating the models’ performances [42], although they will be used as support metrics.

RESULTS AND DISCUSSIONS

The experimental results are divided into the models' evaluation in the training set using the time series splits schema and the test set, which includes data points external to the training process.

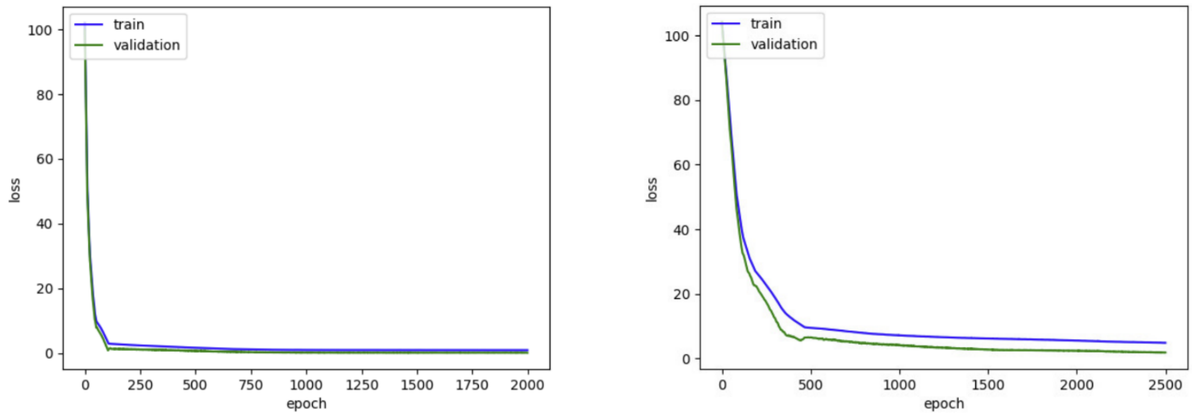
A. *Performance evaluation in the training set*

The trained models were assessed using the corresponding testing set for each fold. The performance evaluation of the proposed methods in terms of the evaluation metrics can be read in Table 1. Where the results for the oneDay and the sevenDay forecasting with the best metrics are highlighted. LSTM1 and CNN1 were the models created to perform oneDay forecasting with the OneDay data set. On one hand LSTM1 had its best performance in the 2000th epoch, reaching a MAPE score of 1.74 and a standard deviation of 1.17. On the other hand, CNN1 reached its best result on the 1995th epoch, obtaining MAPE score of 0.91 and a standard deviation of 0.66. Both of these models show great performance, with an error lower than than 1% and a small standard deviation relative to the metric's score. The models developed to conduct sevenDay forecasting using the SevenDay data set were LSTM7 and CNN7. On the one hand, the 2500th epoch saw LSTM1 perform at its best, with a MAPE score of 8.87 and a standard deviation of 3.61. The 2500th epoch also saw CNN1 achieve its highest performance, with a MAPE score of 4.85 and a standard deviation of 3.92. With accuracy more than 90%, and a low standard deviation relative to the metric's score, both of these models exhibit a good performance. In the case of the oneDay forecasting, the CNN1 model seems to outperform the LSTM1 model. From Table 1, it is possible to read that the CNN1 MAPE score is lower than

Table 1: Training performance results of deep learning models. The best performing model was selected for oneDay and sevenDay forecasting, and are highlighted on the table

LSTM1	2000	1.74 ± 1.68	197996.90 ± 205625.21	117436.98 ± 123737.21
CNN1	**1995**	$\mathbf{0.91 \pm 0.66}$	$\mathbf{49525.91 \pm 36086.36}$	$\mathbf{93709.89 \pm 66536.83}$
LSTM7	2500	8.87 ± 3.61	$2512614.45 \pm 1727948.96$	$1637341.59 \pm 1170417.57$
CNN7	**2500**	$\mathbf{4.85 \pm 3.92}$	$\mathbf{356603.41 \pm 329296.35}$	$\mathbf{207623.40 \pm 143426.95}$

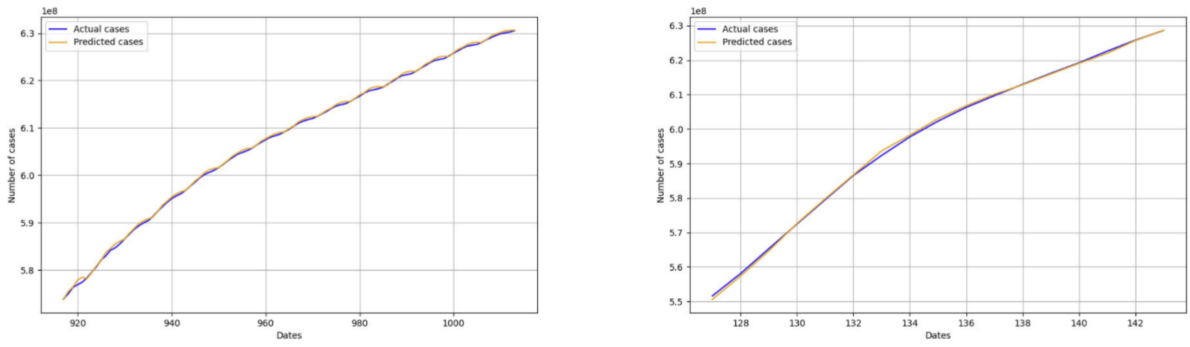
u- units; mean and standard deviation of metrics MAPE, RMSE, MAE over ten-folds



(a) (b)

Figure 3: (a) CNN1 test/train model loss (b) CNN7 test/train model loss

that of the LSTM1 architecture. Nevertheless, the model's performance difference with the LSTM1's was not very significant, having a variation of less than 1%. However, CNN1 and LSTM1 scored a RMSE of 49525.91 and 197996.90 respectively, having a significant difference proving that in fact CNN1 had a more successful training. Figure 3. a) shows the loss relative to the epoch during training stage, showing an effective training with both the train and validation curves close together and with no sign of overfitting. The CNN1 model appears to perform better than the LSTM7 model in the sevenDay forecasting scenario as well. It is clear from Table 1 that the LSTM7 MAPE score is inferior to of the CNN7 by a considerable difference. Figure 3. b) shows the loss relative to the epoch for the CNN7 model during training stage. The figure shows an effective training, with both curves reaching a loss lower than 10%,



(a) (b)

Figure 4: (a) CNN1 (b) CNN7 - Predicted vs actual number of cases

and no apparent overfitting. Additionally, Table 1 shows that for sevenDay predictions, the MAPE and its respective standard deviation was relatively higher than the oneDay predictions. The way the data splits were performed, using time series split, may have affected these average values, as the first folds start with a relatively small amount of training values, resulting in a high MAPE.

B. *Performance evaluation in the test set*

Using the two models chosen in the previous epigraph, CNN1 and CNN7, respectively, were subjected to the combinations testOneDay and TestSevenDay. This makes it possible to confirm the generalization ability of each model's prediction.

The best performing model trained on the last fold of each time series split was used for the testing stage of this experiment. Data obtained from the prediction and data regarding the actual number of infected worldwide was used to generate a plot and analyze it. This can be found in Figure 4. Generally, the closer the lines are to each other the better the prediction performed by the model. As the analysis performed for the test stage, the CNN1 (Figure 4 (a)) and CNN7 (Figure 4 (b)) made really accurate predictions. The curves of the number of current (blue curve on the graph) and predicted cases (yellow curve on the graph) match very well, having a similar behavior in both CNN1 and CNN7. Despite reaching good test results, CNN1 performance is comparably better than that of the CNN7. One of the factors that may have affected the sevenDay predictions is the data preparation. The data pre-processing, as it resulted in a relatively small amount of data points for these cases, and this could've affected negatively the training of the models. Despite training them with 500 more epochs, the results show that the training was not very successful. This means that the data set size was not big enough as to train these models. Additionally, Table 1 shows that for these models the MAPE and its respective standard deviation was relatively high. The way the data splits were performed, using time series split, may have affected these average values, as the first folds start with a very small amount of training values, resulting in a high MAPE. The last folds train with a bigger amount of data, having better results in all metrics.

CONCLUSIONS AND FUTURE WORK

From its start, the pandemic had a huge impact on the socioeconomic aspect of the world. The solution presented in this paper can prepare the authorities and any person to take the appropriate measures to help fight and eventually eradicate the SARS-COV-2 virus. The results obtained during the experimentation show that the proposed methods can help predict the number of COVID-19 cases effectively, having a relatively low score on the metrics. Both of the architectures, LSTM and CNN proved to be very useful while making predictions. The amount of data used to make single-day predictions made the training of the models reach very good results. Despite the seven-day predictions did not have quite the same results, they still offer useful information with their forecasting, making them relevant for this purpose. As future work, we plan to experiment with the proposed method using more data points since the obtained results for seven-day predictions were affected by the time series split technique, which reduced the amount of available data. Additionally, multi-step forecasting could be tested, and compare the results, as it may solve the issue of data reduction to achieve single-step forecasting with seven-day predictions

ACKNOLEGDEMENT

Authors thank to the Applied Signal Processing and Machine Learning Research Group of USFQ for providing the computing infrastructure (NVidia DGX workstation) to implement and execute the developed source code. Publication of this article was funded by the Academic Articles Publication Fund of Universidad San Francisco de Quito USFQ

REFERENCES

[1] World Health Organization, "Informe sobre la salud en el mundo," 2022. [Online]. Available: https://www.who.int/health-topics/coronavirus#tab= tab 1

[2] Organizacion Panamericana de la Salud, "Brote de enfermedad por el ´ coronavirus (covid-19)," 2022. [Online]. Available: https://www.paho. org/es/temas/coronavirus/brote-enfermedad-por-coronavirus-covid-19

[3] Asociacion Espa ´ nola de Pediatr ˜ ´ıa de Atencion Primaria, "Pruebas ´ diagnOsticas de laboratorio de covid-19," 2020. [Online]. Available: ´ https://www.aepap.org/sites/default/files/documento/archivos-adjuntos/ pruebas diagnosticas de laboratorio de covid vfinal.pdf

[4] E. Prades Escobar and D. MarAn S ˜ A¡nchez, "Modelos estad ˜ Asticos ˜ para las predicciones de la COVID-19 en Cuba," Revista Cubana de Higiene y EpidemiologAa˜ , vol. 57, 00 2020. [Online]. Available: http://scielo.sld.cu/scielo.php?script=sci arttext& pid=S1561-30032020000100005&nrm=iso

[5] M. S. A. Abotaleb, "Predicting COVID-19 Cases using Some Statistical Models: An Application to the Cases Reported in China Italy and USA," Academic Journal of Applied Mathematical Sciences, vol. 6, no. 4, pp. 32–40, 04-2020 2020. [Online]. Available: https://ideas.repec.org/a/arp/ajoams/2020p32-40.html

[6] G. Bontempi, S. Ben Taieb, and Y.-A. Le Borgne, Machine Learning Strategies for Time Series Forecasting. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 62–77. [Online]. Available: https://doi.org/10.1007/978-3-642-36318-4 3

[7] J. F. Torres, D. Hadjout, A. Sebaa, F. Mart´ınez-Alvarez, and A. Troncoso, ´ "Deep learning for time series forecasting: a survey," Big Data, vol. 9, no. 1, pp. 3–21, 2021.

[8] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," Future Computing and Informatics Journal, vol. 3, no. 2, pp. 334–340, 2018.

[9] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang, and Z. Cheng, "Time-series well performance prediction based on long short-term memory (lstm) neural network model," Journal of Petroleum Science and Engineering, vol. 186, p. 106682, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0920410519311039

[10] D. ZHANG, Y. CHEN, and J. MENG, "Synthetic well logs generation via recurrent neural networks," Petroleum Exploration and Development, vol. 45, no. 4, pp. 629–639, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876380418300685

[11] A. Yadav, C. K. Jha, and A. Sharan, "Optimizing lstm for time series prediction in indian stock market," Procedia Computer Science, vol. 167, pp. 2091–2100, 2020, international Conference on Computational Intelligence and Data Science. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920307237

[12] Y. Hua, Z. Zhao, R. Li, X. Chen, Z. Liu, and H. Zhang, "Deep learning with long short-term memory for time series prediction," IEEE Communications Magazine, vol. 57, no. 6, pp. 114–119, 2019.

[13] J. Zhang, Y. Zhu, X. Zhang, M. Ye, and J. Yang, "Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas," Journal of Hydrology, vol. 561, pp. 918– 929, Jun. 2018.

[14] M. A. Achterberg, B. Prasse, L. Ma, S. Trajanovski, M. Kitsak, and P. Van Mieghem, "Comparing the accuracy of several network-based covid-19 prediction algorithms," International Journal of Forecasting, vol. 38, no. 2, pp. 489–504, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207020301552

[15] F. Han, J. Yao, H. Zhu, and C. Wang, "Underwater image processing and object detection based on deep cnn method," Journal of Sensors, vol. 2020, 2020.

[16] Y. Ren and X. Cheng, "Review of convolutional neural network optimization and training in image processing," in Tenth International Symposium on Precision Engineering Measurements and Instrumentation, vol. 11053. SPIE, 2019, pp. 788–797.

[17] B. Chen, H. Li, and W. Luo, "Image processing operations identification via convolutional neural network," arXiv preprint arXiv:1709.02908, 2017.

[18] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.

[19] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," Journal of Systems Engineering and Electronics, vol. 28, no. 1, pp. 162–169, 2017.

[20] I. Koprinska, D. Wu, and Z. Wang, "Convolutional neural networks for energy time series forecasting," in 2018 International Joint Conference on Neural Networks (IJCNN), 2018, pp. 1–8.

[21] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," 2017. [Online]. Available: https://arxiv.org/abs/1703.04691

[22] C.-J. Huang, Y.-H. Chen, Y. Ma, and P.-H. Kuo, "Multiple-input deep convolutional neural network model for covid-19 forecasting in china," medRxiv, 2020. [Online]. Available: https://www.medrxiv.org/content/ early/2020/03/27/2020.03.23.20041608

[23] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," International Journal of Forecasting, vol. 36, no. 1, pp. 54–74, 2020.

[24] "Information technology — Artificial intelligence — Assessment of machine learning classification performance," International Organization for Standardization, Geneva, CH, Standard, Mar. 2022.

[25] Y. Haik and T. M. Shanin, Engineering design process: Second edition. Cengage Learning, 2010.

[26] O. services, "Home." [Online]. Available: https://data.humdata.org/ event/covid-19

[27] J. Schmidhuber, "Deep learning in neural networks: An overview," Neural networks, vol. 61, pp. 85–117, 2015.

[28] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2010.

[29] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A ´ survey," Computers and electronics in agriculture, vol. 147, pp. 70–90, 2018.

[30] L. Deng, G. Hinton, and B. Kingsbury, "New types of deep neural network learning for speech recognition and related applications: An overview," in 2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013, pp. 8599–8603.

[31] B. A. Richards, T. P. Lillicrap, P. Beaudoin, Y. Bengio, R. Bogacz, A. Christensen, C. Clopath, R. P. Costa, A. de Berker, S. Ganguli et al., "A deep learning framework for neuroscience," Nature neuroscience, vol. 22, no. 11, pp. 1761–1770, 2019.

[32] A. A. Mishra, A. Edelen, A. Hanuka, and C. Mayes, “Uncertainty quantification for deep learning in particle accelerator applications,” Physical Review Accelerators and Beams, vol. 24, no. 11, p. 114601, 2021.

[33] M. K. Leung, H. Y. Xiong, L. J. Lee, and B. J. Frey, “Deep learning of the tissue-regulated splicing code,” Bioinformatics, vol. 30, no. 12, pp. i121–i129, 2014.

[34] M. Korpusik, Z. Liu, and J. R. Glass, “A comparison of deep learning methods for language understanding.” in Interspeech, 2019, pp. 849– 853.

[35] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” nature, vol. 521, no. 7553, pp. 436–444, 2015.

[36] IBM Cloud Education, “Recurrent neural network,” 2020. [Online]. Available: https://www.ibm.com/cloud/learn/recurrent-neural-networks

[37] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation, vol. 9, pp. 1735–80, 12 1997.

[38] G. Van Houdt, C. Mosquera, and G. Napoles, “A review on the long ´ short-term memory model,” Artificial Intelligence Review, vol. 53, no. 8, pp. 5929–5955, 2020.

[39] T. Kourkounakis, A. Hajavi, and A. Etemad, “Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory,” in ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 6089– 6093.

[40] R. L. Abduljabbar, H. Dia, and P.-W. Tsai, “Unidirectional and bidirectional lstm models for short-term traffic prediction,” Journal of Advanced Transportation, vol. 2021, 2021.

[41] J. Wang, F. Hu, and L. Li, “Deep bi-directional long short-term memory model for short-term traffic flow prediction,” in International conference on neural information processing. Springer, 2017, pp. 306–316.

[42] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, “A survey of convolutional neural networks: analysis, applications, and prospects,” IEEE transactions on neural networks and learning systems, 2021.

[43] S. Liu, H. Ji, and M. C. Wang, “Nonpooling convolutional neural network forecasting for seasonal time series with trends,” IEEE Transactions on Neural Networks and Learning Systems, vol. 31, no. 8, pp. 2879–2888, 2020.

[44] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” arXiv preprint arXiv:1603.07285, 2016.

[45] J. Brownlee, “4 strategies for multi-step time series forecasting,” Mar 2017. [Online]. Available: https://machinelearningmastery.com/ multi-step-time-series-forecasting/

[46] S. Suradhaniwar, S. Kar, S. S. Durbha, and A. Jagarlapudi, “Time series forecasting of univariate agrometeorological data: a comparative performance evaluation via one-step and multi-step ahead forecasting strategies,” Sensors, vol. 21, no. 7, p. 2430, 2021

[47] Stanghong, “How to do time series split using sklearn,” 2021. [Online]. Available: https://medium.com/@Stan DS/ timeseries-split-with-sklearn-tips-8162c83612b9