Original scientific paper

Long Short-Term Memory Prediction for COVID19 Time Series

Milan S. Milivojević, *Student Member, IEEE*, and Ana Gavrovska, *Member, IEEE*

***Abstract* —Entire world has been dealing with the number of new Coronavirus 2 or COVID-19 cases. The spread of this severe acute respiratory syndrome has produced many concerns worldwide. Having data related to coronavirus available for tests, novel models for forecasting the number of new cases can be developed. In this paper, a long short-term memory (LSTM) based methodology is applied for such prediction. Here, experimental analysis is performed with the parameters, such as the number of layers and units of the network. The root mean squared error is calculated for data corresponding to the Republic of Serbia, as well as per different continents. The results show that LSTM model can be useful for further analysis and time series prediction.**

***Keywords* — Prediction, neural network, LSTM, Acute Respiratory Syndrome, COVID-19, Root Mean Squared Error.**

I. Introduction

COVID19 is the name of an infectious disease that appeared at the end of 2019, and that produced a lot of concern around the globe since it has expanded to more than two hundred countries and territories around the world. The SARS-CoV-2 is an acronym that stands for Severe Acute Respiratory Syndrome Coronavirus 2, where the disease caused by SARS-CoV-2 is called COVID-19. The novel coronavirus (COVID-19) outbreak has been declared by the World Health Organization (WHO) on March 11, 2020 [1]-[2].

The virus has caused numerous negative effects. The flu-like symptoms, such as dry cough, sore throat, fever and breathing issues, can be mild or extremely severe. It could be helpful to develop a forecasting model in order to predict the number of COVID-19 cases. Since there is available coronavirus data from the WHO - World Health Organization, or the ECDC - European Center for Disease Prevention and Control, developing such models for forecasting is possible.

A number of mathematical models for prediction have been proposed so far in the literature [3]-[4], where machine and deep learning methodologies, like neural network (NN) based ones, are expected to be high quality solutions for forecasting COVID-19 time series [5].

One of the common types of neural networks is a recurrent neural network (RNN), valuable for time series modeling. It can be useful for language/speech processing, stock price prediction, etc. [6]. In this paper, Long Short-Term Memory (LSTM) approach is applied, which is a specific RNN architecture designed for time series and analysis of long-range dependencies. This network can make effective results, converges quickly, and may outperform a deep feed forward neural network [7].

Here, we particularly focus on an experimental analysis consisted of five phases in order to test different parameters of the applied LSTM base architecture, like the number of layers and hidden units. The error is estimated according to different parameter values.

The paper is organized as follows. In Section II the long short-term memory (LSTM) cell is described for understanding the basics of the architecture. In Section III, materials and methods used for the simulation are described. An initial experimental analysis is performed, and further details about the simulation are given. Section IV is dedicated to the presentation of experimental results and discussing the main steps in the selection of the parameters. Finally, the obtained conclusions are summarized in Section V.

II. Long Short-Term Memory (LSTM) Cell

With conventional backpropagation for recurrent learning, error flowing backward in time tends to affect the estimation process. The temporal evolution of the backpropagated error may lead to oscillating weights applied within the architecture. The time series based approaches can be time-consuming.

Long short-term memory (LSTM) is designed in order to overcome error flowing issue, even in the case of noisy input sequences, without loss of short-time lag capabilities. The architecture is based on a gradient-based algorithm, where the gradient computation is truncated at certain specific points [8]-[9]. In Fig. 1 a LSTM cell is illustrated.

In a LSTM neural network, LSTM cell blocks are placed instead of the standard layers used in neural network models. The cell blocks consist of three building blocks: input gate, forget gate and output gate.

Paper received April 16, 2021; revised July 20; accepted August 02, 2021. Date of publication December 30, 2021. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Irini Reljin.

This paper is a revised and expanded version of the paper presented at the 28th Telecommunications Forum TELFOR 2020 [18]

This work was supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia.

Milan S. Milivojević is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: msmilance@etf.bg.ac.rs, milansmilivojevic@gmail.com).

Ana M. Gavrovska is with the School of Electrical Engineering, University of Belgrade, Bulevar kralja Aleksandra 73, 11120 Belgrade, Serbia (e-mail: anaga777@etf.bg.ac.rs, anaga777@gmail.com).

The input vector x(t) is first combined with the output from the previous cell h(t-1). The activation function tangent hyperbolic (tanh) is applied to the new vector thus obtained. Via the input gate, the input vector is shaped using the sigmoid σ activation and multiplies element by element with the previously obtained vector. The purpose of the input gate is to reduce the unnecessary elements. The forget gate has the task related to storing information about the internal state. A time lag equal to one is applied, where both variables s(t) and s(t-1) are stored.

The sigmoid activation function is applied. Multiplication with s(t-1) is performed in order to finally get the output s(t), as illustrated in Fig. 1. The addition presented in Fig. 1 instead of multiplication is used to overcome the vanishing gradient issue, where the addition result is formed by the tanh function multiplying the element by the element with the output of the output gate. The output gate applies the σ activation function and controls which values will appear at the output of the network [10].

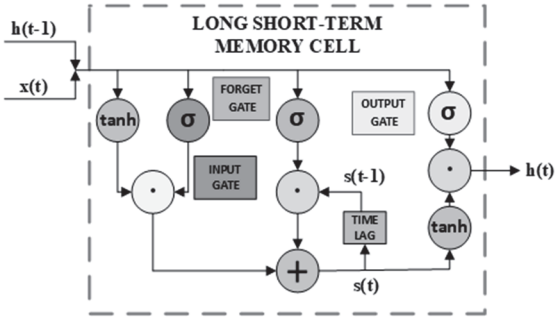

Fig. 1. LSTM cell.

The dimension of the hidden layer (L) characterizes LSTM. Namely, each sigmoid, hyperbolic tangent or hidden layer represents a set of nodes whose number is equal to the dimension of the hidden layer. Therefore, each node in an LSTM cell actually represents a cluster of standard neural network nodes [11].

GRU (Gated Recurrent Unit) aims to solve the vanishing gradient issue that is characteristic of a standard RNN. GRU can also be considered as similar to a LSTM because both are designed similarly and, in some cases, produce equally satisfying results. To solve the RNN vanishing gradient issue, GRU uses a so-called update gate and reset gate. The GRU network can be combined with LSTM cells in the form of an input block [12].

Besides LSTM, ARIMA (Autoregressive Integrated Moving Average model) is applied. The ARIMA uses time-series, interprets data using past values and performs statistical analysis to make data predictions using a linear regression. A nonseasonal ARIMA model is classified as an ARIMA(p,d,q) model, where p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationarity, and q is the number of lagged forecast errors in the prediction equation [13]-[14].

III. Simulation

A. Simulation material

In order to make an adequate response to COVID-19, ECDC's Epidemic Intelligence team has started collecting the number of cases and deaths from the beginning of the pandemic [15]. The numbers have been collected on a daily basis from reports of health authorities around the world. Such material is helpful for research analysis and prediction of COVID-19 trends.

This outbreak has been global. It has been monitored worldwide. The point was to collect data that correspond to all affected continents, countries or territories. The data is organized in the form of Excel/CSV tables, and the tables contain a lot of information related to the course of pandemic. The organization of material is according to date information. The tables consist of the number of newly confirmed cases, number of deaths according to specific time, as well general information, like the number of inhabitants, etc.

Data, which correspond to the period from March 21 to September 1 of the last year (2020) for the Republic of Serbia, is presented in Fig.2. The number of newly confirmed cases on a daily basis for the Republic of Serbia is presented for this period.

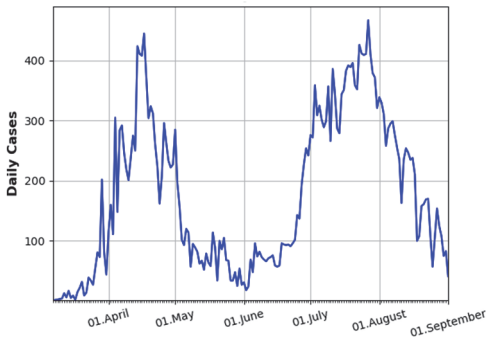

Fig. 2. Confirmed COVID-19 cases on the daily basis in the Republic of Serbia for a specific period (from March to September).

Similarly, for any specific time interval from the beginning of the pandemic for a selected country data can be analyzed. Specific time series are of interest for monitoring changes and trends during different periods.

B. Simulation methods

For simulation, the Python framework is used in this paper. Python libraries like numpy, pandas, keras and tensorflow, have been used for the analysis of time series. In Fig. 3 the basic statistical analysis is illustrated for the time series, which represents the number of new cases on a daily basis in the Republic of Serbia (Fig.2). A data histogram is generated, and the values corresponding to the moments of the first and the second order are estimated. The histogram was fitted according to the normal distribution, and the values of the Loc and Scale parameters were estimated. These parameters correspond to the mean value and the standard deviation and their values are shown within the legend. For the parameters estimated like this, the fitted normal distribution is shown on the same graph where the histogram is shown.

In order to observe the extent to which the data distribution corresponds to the normal distribution, the PP (Probability Plot) is also shown, as in Fig.3. The PP axes show the theoretical quantiles corresponding to the normal distribution and the data values ordered [16]. It can be observed in Fig. 3 that there are certain deviations from the

trend that corresponds to the straight line with an inclination of 45 degrees. The statistics of data that correspond to COVID-19 new cases are more complex, so the use of complex models involving the neural network, such as LSTM, is quite justified.

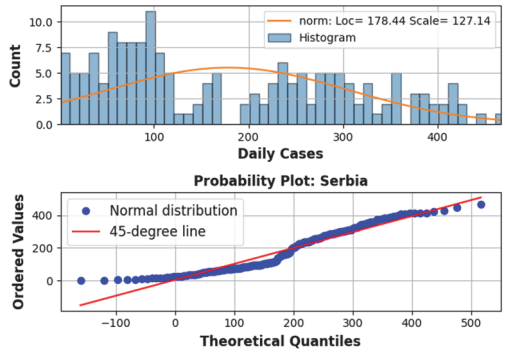

Fig. 3. Basic statistical analysis of the new daily-confirmed cases of COVID-19 in the Republic of Serbia.

In the experiment with the neural network performed in this paper, time series are firstly preprocessed. Data values x are normalized:

$$X = \frac{x - \min(x)}{\max(x) - \min(x)}, \quad (1)$$

and the obtained values X correspond to the range [0, 1]: This kind of preprocessing is not rare [17].

Data is then divided into a training and a test set. The applied ratio is eighty versus twenty percent of the total data, where 80% represent the training part and 20% represent the test set. Supervised learning is performed. Data is organized in order to have input and output patterns. The observation at the previous step represents an input to predict the observation at the current time step. So, it can be stated that the target value named X(t) at time instance t is a function of X(t-1). Scaling of the data is necessary in order to perform further processing or error calculation. Also, data shifting needs to be performed having in mind time axis and time period used for the calculation [18].

A large number of parameters describe a neural network architecture, where most of them are adjusted empirically. In Fig. 4, a neural network architecture applied in this paper is illustrated with its main steps and blocks showing data flow and dealing with specific dimensions [19].

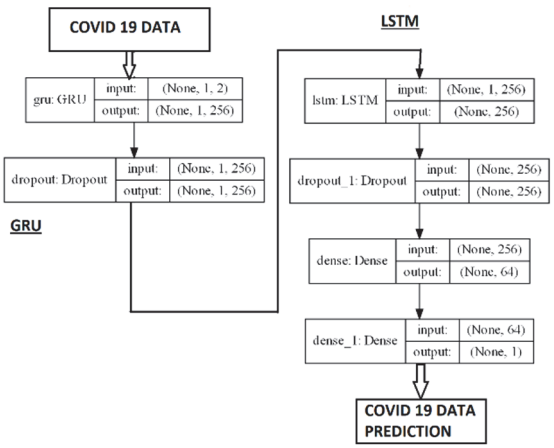

Fig. 4. Neural network architecture with indicated data dimensions at each step.

The input is preprocessed COVID-19 data. The first block represents the GRU network, which is described by the number of units (G). This number can also be explained as a hidden layer dimension, where 256 is the default value of the number of GRU units. In other words, COVID-19 data are mapped into G units in this first block. The second step is dropout. The main block is dedicated to LSTM. Its number of hidden units, noted as L, can describe the LSTM network. In this case the number of hidden units or L takes the value of 256. After another dropout step, there are two dense layers at the output. The dense layer is a neural network layer that is connected deeply. This means that each element or neuron in a dense layer receives input from all other elements/neurons representing its previous layer. The dense layer is found to be one of the most commonly used layers in a neural network model. The dense layer performs a matrix-vector multiplication. Values in the matrix are actually values that can be used for training and updating with the help of backpropagation approach [20]. Dense layer is described by its dimension D. Here, the first dense layer is characterized by dimension 64, and the second one has the dimension equal to D = 1. The output gives one value as the predicted one. The dropout operation rejects according to a set rejection rate. Here, the rejection rate is 25%. The dropout layer randomly sets input units to 0 with the frequency of rejection rate at each step during training time. This is helpful for overfitting prevention. Inputs that are not set to 0 are scaled up by a factor of 1/(1-rate) so that the sum of all inputs is unchanged. In the case of dense layer, activation function is of the ReLU (Rectified Linear Unit) type, where for GRU and LSTM step, the tangent hyperbolic and sigmoid function are used [21].

The number of epochs is set in advance in the training process in order to describe the number of attempts for setting the error to a minimum level having in mind acceptable limits. For the simulation purposes, it was necessary to perform an initial experiment and the obtained comparison between train and test error is shown in Fig. 5. During this initial experiment, it is concluded that more than 50 epochs is not necessary, as illustrated in Fig. 5.

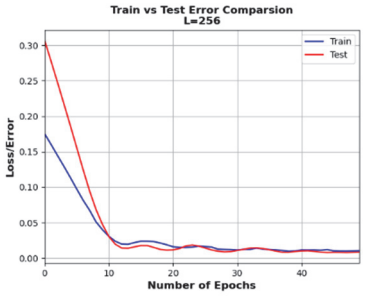

Fig. 5. Overview of neural network training step depending on the number of epochs and test results.

From the error value standpoint, the obtained errors may be considered acceptable for the initial examination. The batch size is a hyperparameter, which describes the number of samples for experiment before model updating. Here, 32 is a default value for the batch size [22].

Since the purpose of the neural network architecture is to predict the number of COVID-19 cases, besides the error comparison the initial experiment included values which

correspond to these numbers. This is illustrated in Fig. 6 for the prediction in the case of the Republic of Serbia. The prediction uses relevant previous data meaning knowledge about the coronavirus pandemic. Similar approaches can be seen in the literature for other countries like India [23].

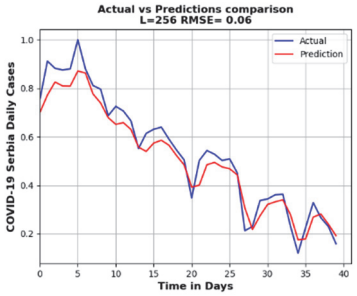

Fig. 6. Initial comparison between actual and predicted values of the number of COVID-19 cases using the neural network architecture.

The success of the prediction is evaluated using error estimation. In this paper, one of the most common error estimations called Root Mean Squared Error (RMSE) is calculated [24], as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N} E_i^2} \,. \qquad (2)$$

The RMSE is based on the difference between the test values and the prediction ones, where the difference is noted as E_i. During the initial experiment, the error is equal to 0.06. Previously, a comparison between actual and predicted values is shown, where the normalization has no significant effect on the error calculation. The true predicted values can be also calculated with the application of the denormalization step. Even though it is common to work with the normalized amounts, the inverse approach can be a part of the neural network. The network-based methodology is not a novel one, but the experiment in this paper is oriented towards the analysis of the parameters applied during the process, as well as using specific period and territories/countries. Particularly, parameters are examined for the Republic of Serbia.

Since the parameter selection in the neural network is of importance, they are investigated according to error estimation using RMSE. In the first step of the simulation parameter G, representing the size of the hidden layer in the GRU network is analyzed. The second phase is dedicated to the analysis of different numbers of hidden units L in the LSTM block. In the third phase, RMSE values for different numbers of LSTM layers are calculated with a fixed number of L units. RMSE values for different numbers of the batch size are calculated in the fourth phase. Finally, in the fifth phase RMSE statistics during error estimation is analyzed per continent.

IV. Experimental Results

In the first part of the experiment, obtained error values are analyzed for different size of the hidden layer in the GRU network (G). In the experiment, values G are taken from the set of values: 64, 128, 256 and 512. The other parameters remained the same as in the previous (default) case. The parameters related to the neural network can be considered as variables, so an estimate of the prediction error is performed in several iterations for each parameter setting. In this way, it is possible to have a better insight into the behavior of the system. Here, 30 iterations were performed for each parameter setting.

In Fig. 7 the results for the first phase in the form of a box plot where the median value of RMSE, minimum and maximum value, as well as the first and third quartile intervals (Q_1 and Q_3 quartile) are indicated. It is evident that with the increase in the number of hidden units, the median value of the error decreases, and that the interquartile range decreases. The limit to which this happens is reached for G = 256. For G = 512, there is a more intensive expansion of the interquartile interval with a decrease of the median error. GRU parameters are often observed together with LSTM, so G value is usually equal to the dimension of the hidden layer L in the LSTM network. In that case, the dimension of the hidden layer in the GRU block (G) is 256.

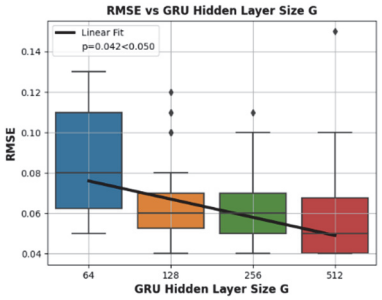

Fig. 7. RMSE values for different numbers of hidden units G in the GRU block.

In the second phase, RMSE error is observed when the size of the hidden layer in the LSTM network takes the values L from the set of values: 64, 128, 256 and 512. The other parameters remained the same as in the previous (default) case. The results for the second phase are shown in Fig. 8 in the form of a box plot. It is evident that with the increase in the number of hidden units, the median value of the error decreases, and that the interquartile range decreases to the value of L = 256. After that, deterioration occurs. The widest range and higher medium values can be seen for L=512. So, it can be considered that the value 256 for L is a good choice.

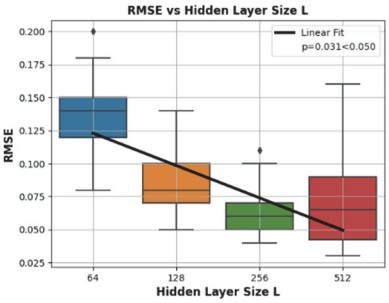

Fig. 8. RMSE values for different numbers of hidden units L in the LSTM block.

In the third phase, the obtained results are presented in Fig. 9. The results are obtained for a fixed number of L, set to 256. In this case the number of LSTM layers changes from one to three. It can be seen that as the number of layers increases, the error median value also decreases. Moreover, it is shown in Fig. 9 that similar median values are obtained

for two and three LSTM layers. Here, the inter-quarter intervals decrease. One should have in mind that even though the RMSE value is decreased, on the other hand the complexity of the system is increased. Therefore, in order to make a compromised selection, the number of layers equal to two is selected for the purposes of prediction of COVID-19 cases.

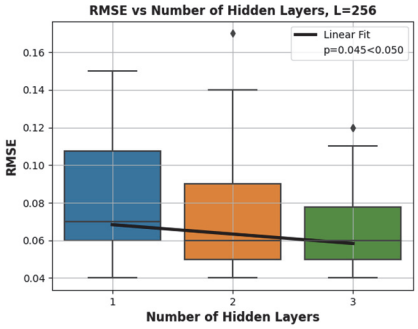

Fig. 9. RMSE values for different numbers of LSTM layers, and for a fixed number of units L = 256.

In addition to the analysis of the error behavior as a function of the number and size of hidden layers within the GRU and LSTM cells, the influence of the size of the batch size parameter is also analyzed. The tested values of the batch size parameter belong to the set of values: 32, 64, 128, 256 and 512. In Fig. 10 the simulation results are presented. It is evident that with an increase in the number of batch size there is an increase in the median value and the interquartile range of RMSE. Therefore, for this hyper parameter, a value equal to 32 is chosen.

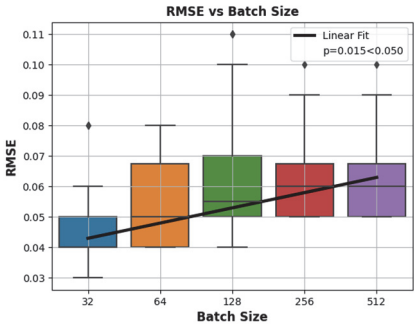

Fig. 10. RMSE values for different batch sizes.

An additional statistical analysis is performed, where for the cases in Figs. 7-10 median value changes are monitored through fitting according to a specific parameter value (G, L, layer number, batch size). A well known two-tailed t-test is performed and p value representing the direction coefficient is estimated. The value $\alpha = 0.05$ was chosen for the level of significance. Since the parameter values in the cases presented in Figs. 7-10 are less than the significance level, the null hypothesis that the direction coefficient is equal to zero is rejected. It can be stated that there is a statistically significant dependence between the median and the parameter values.

In Table 1 the results of comparison between LSTM and ARIMA approach [3] are shown for three different countries and for recent months in the Republic of Serbia.

TABLE 1: COMPARISON BETWEEN LSTM AND ARIMA APPROACHES FOR THREE COUNTRIES

Method	*RMSE- Serbia*	*RMSE-Croatia*	*RMSE-Hungary*
LSTM	0.06	0.10	0.04
ARIMA	0.12	0.17	0.07

For selected values of model parameters (dimensions of hidden layers in GRU and LSTM network, number of hidden layers, as well as batch size), the prediction is performed for the entire currently available database. In other words, available data at the moment of writing this paper corresponding to the number of daily cases of COVID-19 globally for each country and territory of the world is applied for the simulation. As a result of the prediction, RMSE errors for each state or territory are estimated. In Fig. 11 the error histogram corresponding to European countries is presented. Fitting to the normal distribution was performed, and the values of the Loc and Scale parameters corresponding to the mean value and the standard deviation of the RSME, respectively (here, Loc = 0.07 and Scale = 0.05 are obtained). For the values thus obtained, the fitted normal distribution is also presented in the same Fig. 11.

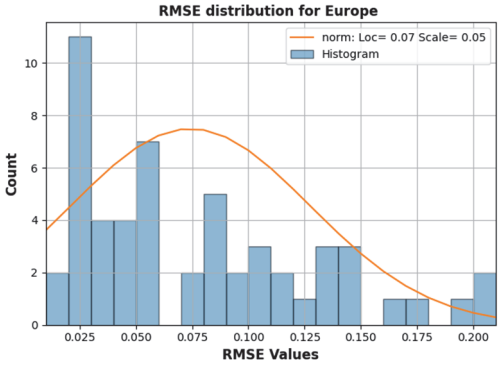

Fig. 11. RMSE error distribution for European countries and selected parameters of LSTM neural network model.

In the next step, this analysis is extended to other continents (Europe, Asia, Africa and the whole of America). The countries are grouped by continents. Thus, the RMSE data is divided into four series. For each of the series, a fitting with a normal distribution is performed, where the mean value ($\overline{\text{RMSE}}$) and standard deviation σ of the RMSE are estimated. The obtained results are shown in Fig. 12. It can be noticed that the mean value is the lowest in the case of European countries. Most similar results per continent are obtained for Asia. As for the standard deviation, it follows the trend of the mean value. The difference between the values per continent is most emphasized here for America (North and South), and for the selected parameters.

The analysis is performed using general error estimation, but one should have in mind other evaluations, as well. RMSE is applied for the evaluation as a standard metric even though a small RMSE value may lead to a relatively high error daily. Despite the selected evaluation, the model behavior can be considered satisfying. Also, different periods during the year may affect the model (e.g. seasons like summer or winter).

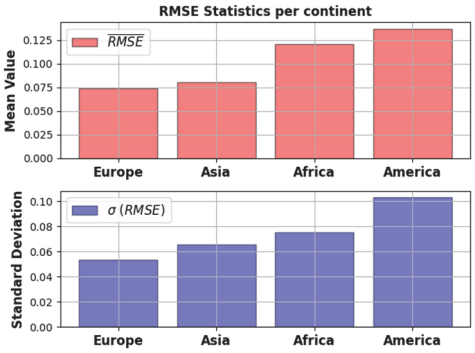

Fig. 12. RMSE statistics of error estimation presented per continent when the selected LSTM neural network parameters are applied.

V. Conclusion

With the advent of the coronavirus pandemic it has become important to generate models which would be helpful in the estimation of the epidemic curve. This paper considers a LSTM neural network and analyzes corresponding parameters for predicting the number of newly infected on a daily basis. Analysis is performed for the data corresponding to the Republic of Serbia, as well as per continent for the selected parameters.

It is of great importance for further management to develop models that adequately predict the trend of COVID-19 cases. In the further work, it is necessary to test the possibilities of prediction models for different territories having in mind particular seasons and parameters.

References

[1] D. Cucinotta and M. Vanelli, "WHO declares COVID-19 a pandemic," *Acta Biomedica*, vol. 91, no. 1. Mattioli 1885, pp. 157–160, Mar. 19, 2020, doi: 10.23750/abm.v91i1.9397.

[2] C. Sohrabi *et al.*, "World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19)," *International Journal of Surgery*, vol. 76. Elsevier Ltd, pp. 71–76, Apr. 01, 2020, doi: 10.1016/j.ijsu.2020.02.034.

[3] İ. Kırbaş, A. Sözen, A. D. Tuncer, and F. Ş. Kazancıoğlu, "Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches," *Chaos, Solitons & Fractals*, vol. 138, p. 110015, Sep. 2020, doi: 10.1016/j.chaos.2020.110015.

[4] M. Djeddou, "Application of Long Short-Term Memory (LSTM) Neural Network for COVID-19 cases Forecasting in Algeria," *COVID-19 Model.*, 2020, doi: 10.13140/RG.2.2.35538.50881.

[5] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553. Nature Publishing Group, pp. 436–444, May 27, 2015, doi: 10.1038/nature14539.

[6] K. Bandara, P. Shi, C. Bergmeir, H. Hewamalage, Q. Tran, and B. Seaman, "Sales demand forecast in E-commerce using a long short-term memory neural network methodology," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Dec. 2019, vol. 11955 LNCS, pp. 462–474, doi: 10.1007/978-3-030-36718-3_39.

[7] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, Jun. 2014, pp. 1724–1734, doi: 10.3115/v1/d14-1179.

[8] S. Hochreiter, S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, "Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies," 2001, Accessed: Sep. 20, 2020. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.7321.

[9] Y. Bengio, P. Simard, and P. Frasconi, "Learning Long-Term Dependencies with Gradient Descent is Difficult," *IEEE Trans. Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994, doi: 10.1109/72.279181.

[10] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[11] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," *Phys. D Nonlinear Phenom.*, vol. 404, Aug. 2018, doi: 10.1016/j.physd.2019.132306.

[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 Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019, doi: 10.1109/MCOM.2019.1800155.

[13] S. L. Ho and M. Xie, "The use of ARIMA models for reliability forecasting and analysis," *Comput. Ind. Eng.*, vol. 35, no. 1–2, pp. 213–216, Oct. 1998, doi: 10.1016/S0360-8352(98)00066-7.

[14] A. Earnest, M. I. Chen, D. Ng, and L. Y. Sin, "Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore," *BMC Heal. Serv. Res. 2005 51*, vol. 5, no. 1, pp. 1–8, May 2005, doi: 10.1186/1472-6963-5-36.

[15] "Sources - Worldwide data on COVID-19." https://www.ecdc.europa.eu/en/publications-data/sources-worldwide-data-covid-19 (accessed Nov. 04, 2020).

[16] P. Mishra, C. M. Pandey, U. Singh, A. Gupta, C. Sahu, and A. Keshri, "Descriptive statistics and normality tests for statistical data," *Ann. Card. Anaesth.*, vol. 22, no. 1, pp. 67–72, Jan. 2019, doi: 10.4103/aca.ACA_157_18.

[17] J. Jin, M. Li, and L. Jin, "Data Normalization to Accelerate Training for Linear Neural Net to Predict Tropical Cyclone Tracks," *Math. Probl. Eng.*, vol. 2015, 2015, doi: 10.1155/2015/931629.

[18] M. S. Milivojević and A. Gavrovska, "Long Short-Term Memory Forecasting for COVID19 Data," *2020 28th Telecommunications Forum (TELFOR), 2020*, pp. 1-4, doi: 10.1109/TELFOR51502.2020.9306601.

[19] S. Gulli, Antonio, Pal, "Deep Learning with Keras: Implementing deep learning models and neural networks with the power of Python." https://www.amazon.com/Deep-Learning-Keras-Implementing-learning/dp/1787128423 (accessed Sep. 26, 2020).

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Nov. 2017, vol. 2017-January, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.

[21] H. H. Sak, A. Senior, and B. Google, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," 2014.

[22] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312–315, Dec. 2020, doi: 10.1016/j.icte.2020.04.010.

[23] P. Arora, H. Kumar, and B. K. Panigrahi, "Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India," *Chaos, Solitons and Fractals*, vol. 139, p. 110017, Oct. 2020, doi: 10.1016/j.chaos.2020.110017.

[24] T. Chai, T. Chai, and R. R. Draxler, "Ozone health and ecosystem impacts View project Root mean square error (RMSE) or mean absolute error (MAE)?-Arguments against avoiding RMSE in the literature," *Geosci. Model Dev*, vol. 7, pp. 1247–1250, 2014, doi: 10.5194/gmd-7-1247-2014.