Visualization of the Epidemics Forecasting Results

Nataliya Shakhovska, Ihor Darmoriz, Yarosvav Vyklyuk, Yurii Kryvenchuk and Pavlo Pukach

Lviv Polytechnic National University, Lviv, Ukraine, 79013

Abstract

Modeling and forecasting of time series is one of the most importance for various practical applications. Many things are more or less time-dependent. Its analysis can forecast the future behavior to take some action for better results in the future. Research purpose is to develop a software product that has the ability to forecast the spread of the epidemic in relation to its specific features. The comparison of linear model, Convolution neural network and Recurrent neural network for epidemic forecasting is given. The spread of epidemics occurs over a period of time where anybody can see trends of some features during some time. Because the result is influenced by a large number of factors, and the training took place only on a short history, the results are of high quality because the MAPE error does not exceed 30% with a prediction for all characteristics.

Keywords 1

machine learning, forecasting, time series, epidemic

1. Introduction

Epidemics produced by infections and viruses usually come to a first-place amount of the large-scale disasters and catastrophes that have attended the entire history of humankind, on a par with starvation, wars, man-made and natural disasters. According to the World Health Organization (WHO), severe respiratory infections account for 60-70% of the total morbidity of the population, with a tendency to develop complexities and chronicity of the process. Due to the extreme variability of the pathogen, acute respiratory infections remain an uncontrolled infection. Another example is coronavirus disease affected by the new virus SARS-CoV-2 (COVID-19). Nearly 241 million people worldwide have contracted COVID 19 (https://index.minfin.com.ua/ua/reference/coronavirus/geography/). Of these, more than 17 million are ill at this moment, and more than 21 million have been cured. More than 4 million people died from the disease. In total, the disease was detected in 203 countries.

The nature of diseases caused by infections and viruses (even with known treatment prevention schemes) depends of numerous factors, namely:

- variability of strains,
- method of distribution,
- parameters of the distribution area: climatic conditions, infrastructure and connections between towns and inside towns, quality of medical care, the most common life style, chronic diseases essential in this area, political situation, etc.

That is why developing simulation models of the spread and character of morbidity and new cases of various infections and viruses is a problematic scientific task. The main characteristics of this task are the following:

- multicriteria: type of spread (epidemic spread, controlled spread in a mild form of the disease), initial parameters, the distribution territory,

Informatics & Data-Driven Medicine, 11, 2021, Lviv, Ukraine
EMAIL: nataliya.b.shakhovska@lpnu.ua (NS); ihor.darmoriz.kn.2017@lpnu.ua (ID); yaroslav.vyklyuk@gmail.com (YaV), yurii.p.kryvenchuk@lpnu.ua (YuK), pavlo.p.pukach@lpnu.ua (PP)
ORCID: 0000-0002-6875-8534 (NS 0000-0003-2549-1873 (ID); 0000-0003-4766-4659 (YaV), 0000-0002-2504-5833 (YuK), 0000-0002-0488-6828 (PP)

- time dependence,
- simulation interval,
- variety of input data.

Therefore, it is necessary to develop a system that is more sensitive to changes in the spread of the disease to predict its reach in the future and monitor other changes that may occur during an epidemic (new cases, recovery, mortality, etc).

The aim of the work is to develop a model and system based on it that would make it possible to monitor and predict the spread of the epidemic on the basis of various characteristics.

The main contributions of this paper are the following:

- The new schema of recurrent neural network for COVID-19 infection forecasting is developed.
- To increase the predictive accuracy, the clustering is used on the preprocessing stage. It allows to reduce the influence of data heterogeneity due to the presence of several locations.

This paper is organized into several sections. In State of the art section, the methods of times series analysis are given. In the section #3 "Methods and means", new schema of recurrent neural network is proposed. The fourth section presents result of proposed methods and gives data interpretation. The last section concludes the paper.

2. State of the Art

In paper [1] describes that the spread of the H5N1 influenza virus in birds has heightened concern about a new human influenza epidemic. Using epidemiological data collected in the early stages of the outbreak, the authors show how to predict the maximum pervasiveness of a pandemic wave and its amplitude and duration by adapting the epidemic model of mass action to observational data by standard regression analysis.

In [2], authors are used tools of mathematics (particularly wavelet theory) and computer science (machine learning). They have developed a new method of modeling the evolution of epidemics, which is not limited to the human population. The most important new feature of the proposed approach is the following:

- an epidemic can occur in several waves;
- these waves can be global and local;
- in addition, they can occur in different periods and places.

Based on the latest data from the Johns Hopkins database, authors apply the model to several countries (the Czech Republic, France, Italy, Germany, and the US states of New York and Florida). After that, they compare the actual rate of diseases and their predictions to established and other recently developed methods and techniques of prognostication.

In [3], it is described that COVID-19 trend forecasting is a significant problem. This work integrates the latest COVID-19 epidemiological data into a logistics model by June 16, 2020 to meet the epidemic trend constraint and then introduce the constraint value into the FbProphet model, a machine-based time series prediction model to obtain an epidemic curve and predict an epidemic trend. Three significant points are summarized from our modeling results for the world countries, Brazil, Russia, India, Peru, and Indonesia.

Paper [4] proposed a new epidemic model (SuEIR) for predicting the spread of COVID-19, based on the number of confirmed deaths in the United States. In particular, the SuEIR model is a variation of the SEIR model, taking into account untested/unregistered cases of COVID-19, and is trained in machine learning algorithms based on historical data messages. In addition to providing baseline predictions for confirmed cases and fatalities, the proposed SuEIR model can also predict the peak date of active cases and estimate the baseline reproduction number.Time series consist of the following components:

- Seasonal changes;
- Trend;
- Cyclical variations;
- Random variations.

After researching time series predictions, we can conclude that the usage of neural networks is a new application, as over the past century, a large number of linear algorithms have been developed to analyze and predict time series, including ARMA [5, 6], ARIMA [7], VAR [8], HWES [9] and others. Although these types have been quite widespread, they may not always be effective enough. Their disadvantages include the following points:

- They require complete data. Some missing values can actually affect the model. But there are also ways to deal with missing data.
- They rely on linear relationships. In many traditional models, their assumptions are based on a linear basis.
- They usually only deal with one-dimensional data. For example, they can analyze a time series for a single characteristic (such as virus mortality), although when dealing with epidemics, we analyze several types of data.
- They usually do not work well in the long run.

Convolutional neural network (CNN) is a class of deep artificial neural networks that has been successfully used in the analysis of visual images [10]. They are mainly used for work or image analysis, but in some cases they can be used effectively for time series. An important characteristic for the use of this network is the correlation of several types of data in the analysis of the series.

A recurrent neural network (RNN) is a class of artificial neural networks where connections between nodes form a directional graph along a time sequence [11, 12]. This allows him to demonstrate temporal dynamic behavior. A well-known RNN is long-term memory or LSTM, and it has the ability to solve time series problems. LSTM networks eliminate the need for a predefined time window due to the ability to study long-term correlations in different sequences and are able to accurately model complex multidimensional sequences. The advantages and disadvantages of each of the models are given in Table 1.

Table 1
Models comparison

Model	Advantage	Disadvantage
Linear	Simple, easy to understand	Requires complete data; Uses only linear connections; Accuracy is not so high for long-term data
CNN	Less sensitivity to noise compared to linear models; Can work with both one-dimensional and multidimensional data; The ability to work with multi-step predictions	Cannot work with time dependencies
RNN	Advantages are the same as for CNN; working with time dependencies	In some cases it cannot process large time dependencies; Tendency to overfitting

Therefore, considering the advantages and disadvantages of different methods, we can conclude that the best choice is RNN, namely LSTM for future data processing.

3. Materials and Methods

The main characteristics for the analysis of epidemics are geographical identification, as well as characteristics that are new values of the following criteria:

- observations,
- confirmation,
- death,
- recovery.

The feature set is also linked to a specific location, such as the city or region that will be predicted.

A data set was used as input data [13]. This is a time series with different metrics that can be used for analysis or prediction. The dataset contains information about the COVID-19 virus in relation to the cities of Ukraine. In this example, a data set for the Lviv region was used.

zvit_date	registration_area	new_susp	new_confirm	new_death	new_recover
2020-01-27	Львівська	1	0	0	0
2020-01-29	Львівська	1	0	0	0
2020-03-13	Львівська	1	0	0	0
2020-03-16	Львівська	9	0	0	0
2020-03-17	Львівська	5	0	0	0
...	...	...	...	...	...
2021-04-25	Львівська	295	22	11	154
2021-04-26	Львівська	378	301	16	346
2021-04-27	Львівська	469	645	13	365
2021-04-28	Львівська	669	458	10	301
2021-04-29	Львівська	459	73	3	229

Figure 1. Dataset example.

The initial data of the developing system should provide the user with an understanding of the situation regarding the epidemiological main indicators: new diseases, recovery, death, etc. The data is calculated as a prediction based on the original data and create a forecast for this set of characteristics.

As output, the user will receive an apology visualization in the form of a graph for a specific data set, as well as a map with a prediction for a specific region for easier visual perception

The proposed in the paper model is built taking into account three main criteria:

- The number of past days for prediction;
- The number of days to anticipate;
- The number of criteria to consider.

For our case, one day was taken into account to predict the next day using four characteristics by analyzing the previous seven days.

In the beginning, neural network decides what information to remember and what to throw out of the cell state. This action is performed in the "Forget gate" part. X presents input data, H - the result of the current stage, t is the step number. In this part, the sigmoid function considers the input data from h_{t-1} and x_t, and then outputs a number between 0 and 1 for each number from cell C_{t-1}, where 1 will mean completely save the state, and 0 - completely forget it.

$$f_t=sig(W(f)[h_{t-1},x_t]+b(f)).$$

The next step is the “Input gate” section to update the cell status. First, the current state X_t and the previously hidden state h_{t-1} are passed to a sigmoid function to transform values between 0 (important) and 1 (unimportant). Next, the same information about the hidden and current state will be transmitted via the tanh function. For network regularization, operator tanh calculates vector $\check{C}_t$ in range from -1 to 1 for the multiplying.

$$i_t=sig(W(i)[h_{t-1},x_t]+b(i)).$$
$$\check{C}_t=gt=tanh(W(g)[h_{t-1},x_t]+b(g)).$$

When the network has prepared information about the data it receives from the two previous layers, the next step is to decide to save information from the new state in the cellular state in the "Cell state". The previous state of cell C_{t-1} is multiplied by the forgetting vector f_t.

$$C_t=f_tC_{t-1}+\check{C}_th_{t-1}.$$

The last step is to determine the values to pass to the next layer. Initially, the values of the current state and the previous hidden state are passed to the last sigma function. This result is further

multiplied by the new cell state generated from the cell state after transmission through the tanh function. Based on the final value, the network decides what information the hidden state should carry. This latent state is used for forecasting. As a result, the new cell state and the new latent state are carried over to the next time step.

$$o_t=\text{sig}(W(o)x_t+U(o)h_{t-1}+b(o)),$$
$$h_t=o_t+\tanh(C_t).$$

The structure of the model is given in Fig. 2.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (7, 4, 4)                 144
_________________________________________________________________
lstm_1 (LSTM)                (7, 4, 4)                 144
_________________________________________________________________
lstm_2 (LSTM)                (7, 4, 4)                 144
_________________________________________________________________
lstm_3 (LSTM)                (7, 4, 4)                 144
_________________________________________________________________
lstm_4 (LSTM)                (7, 4, 4)                 144
_________________________________________________________________
lstm_5 (LSTM)                (7, 4)                    144
=================================================================
Total params: 864
Trainable params: 864
Non-trainable params: 0
_________________________________________________________________
```

Figure 2. RNN schema.

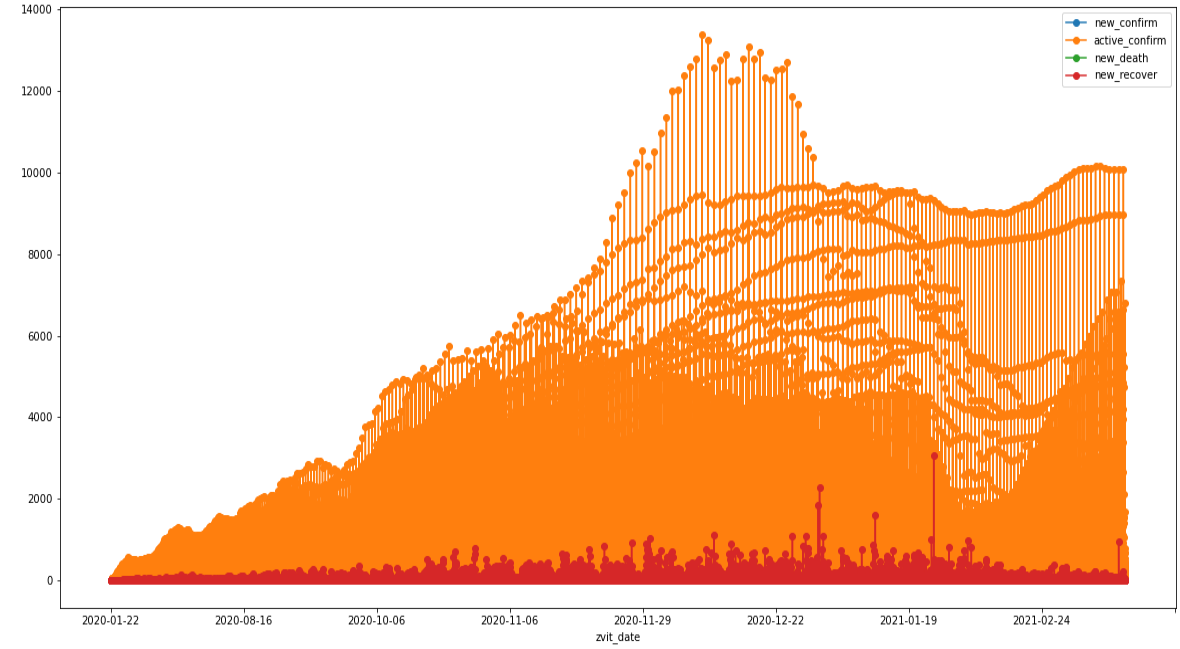
4. Results

Before starting work, all data should be standardized [14], as data is measured at different scales and a large difference can slow down or even hinder the effective learning process:

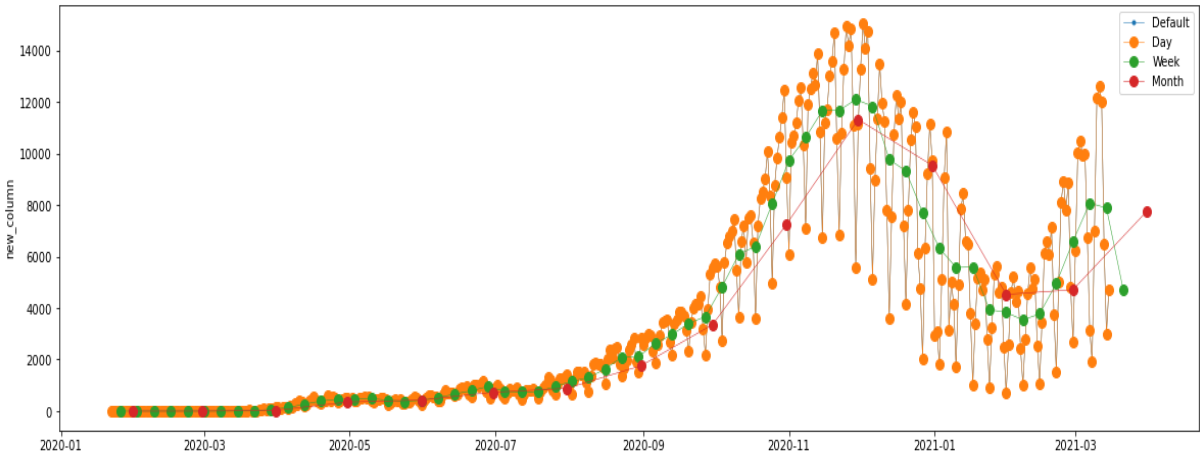
$$x_{scaler} = \frac{x - x_{min}}{x_{max} - x_{min}}.$$

At the preprocessing stage the data gaps are found. To remove them, grouping is used. The distribution before and after grouping is given in Fig.3. Data gaps have narrowed, so grouping data for specific periods is appropriate. It allows to reduce the influence of data heterogeneity due to the presence of several locations

Next, a model will be created to predict the regions, so the next step is to group the data with the selection of a specific area.



a)



b)

Figure 3. Number of new cases for different periods of time before (a) and after grouping (b)

The next step is to train the model. At this stage, the training of the previously created model was carried out with the condition of release, if the accuracy is greater than 92% or all epochs will not be passed. Each model with the lowest loss result is also stored for validation data. The model training process can be seen in Figure 4, and the model training history in Figure 5.

```
- accuracy: 0.7806 - val_loss: 0.1430 - val_accuracy: 0.6765
Epoch 95/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0503
- accuracy: 0.7993 - val_loss: 0.1471 - val_accuracy: 0.6765
Epoch 96/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0527
- accuracy: 0.7709 - val_loss: 0.1418 - val_accuracy: 0.6765
Epoch 97/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0511
- accuracy: 0.7530 - val_loss: 0.1475 - val_accuracy: 0.6765
Epoch 98/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0490
- accuracy: 0.7476 - val_loss: 0.1560 - val_accuracy: 0.5882
Epoch 99/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0507
- accuracy: 0.7062 - val_loss: 0.1563 - val_accuracy: 0.5294
Epoch 100/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0544
- accuracy: 0.7204 - val_loss: 0.1466 - val_accuracy: 0.6765
```

Figure 4. Model training

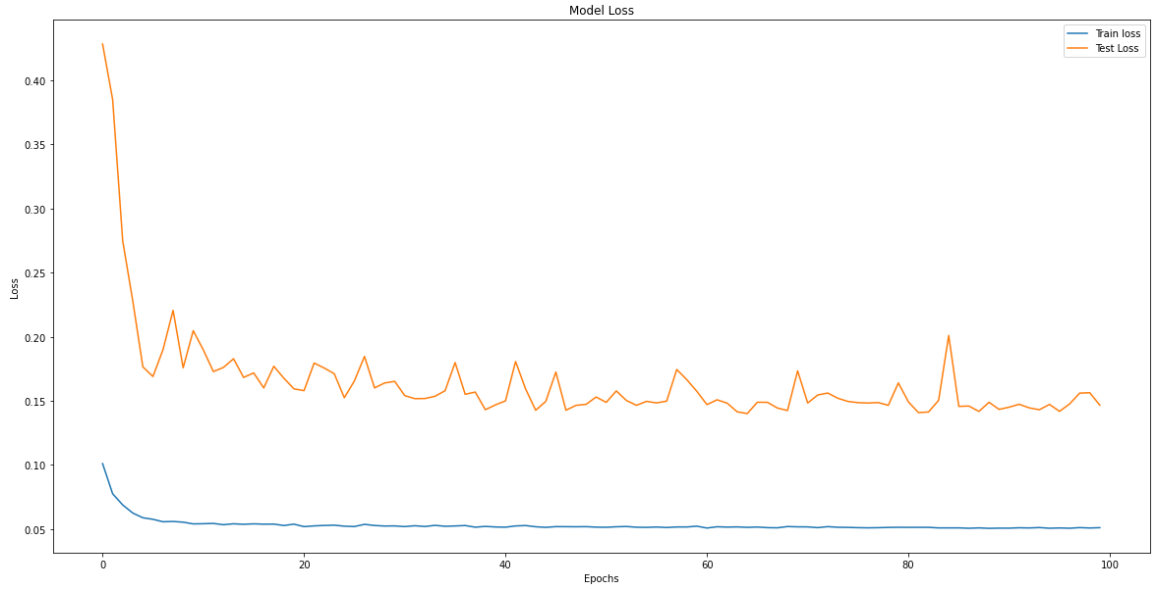
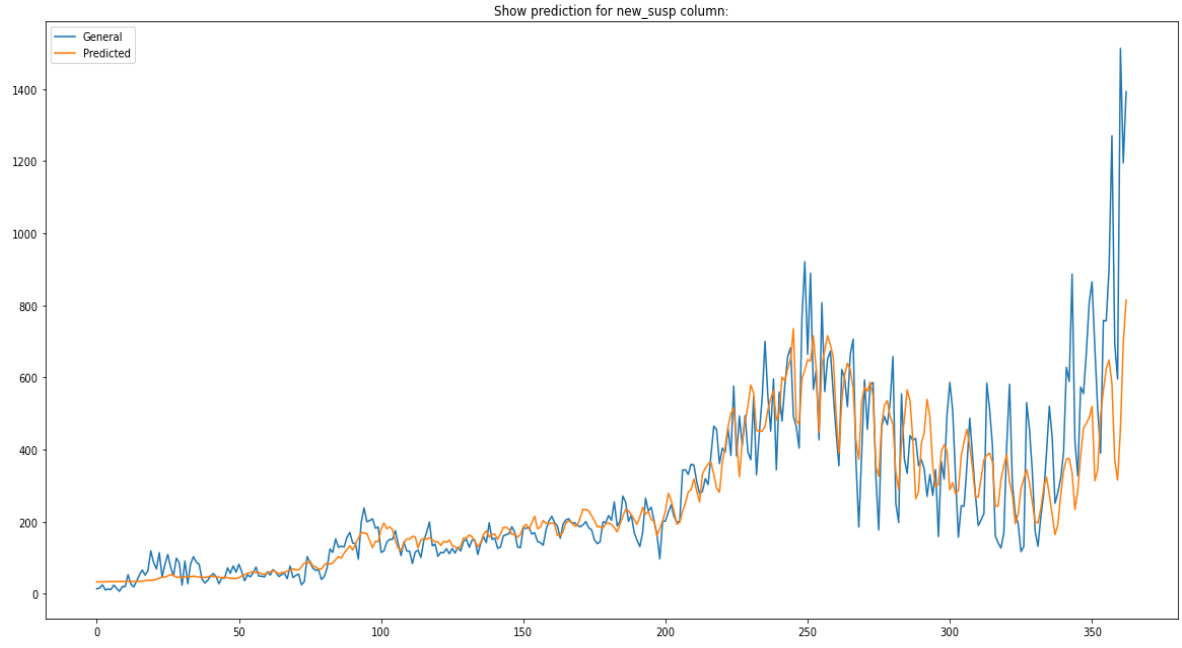
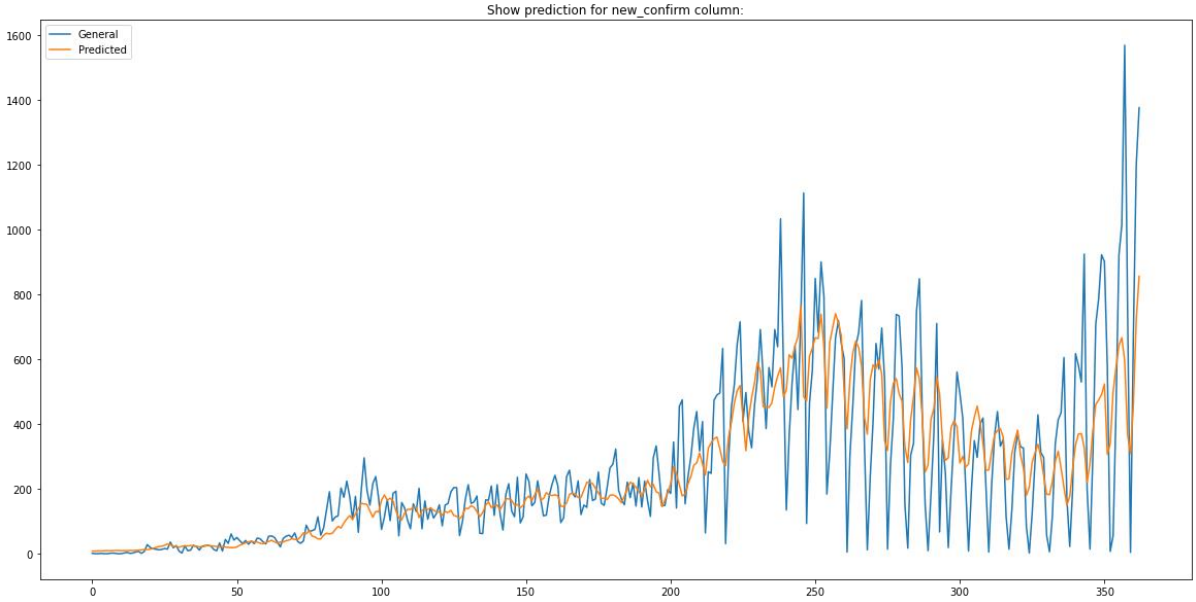


Figure 5. Loss function

The model accuracy on the training data for each of the parameters is given in Fig. 5 and resuls of forecasting is demonstrated in Fig. 6. New cases and new obseravations are modelled separetly.



a)



b)

Figure 6. Forecasting on training data: a) – new observations; b) – new cases.

The graph shows the training data, represented by a blue line, as well as the prediction, represented by an orange line. From these graphs we can say that the accuracy of prediction relative to the test data is very high.

Training process is given in Fig. 7.

```
- accuracy: 0.7806 - val_loss: 0.1430 - val_accuracy: 0.6765
Epoch 95/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0503
- accuracy: 0.7993 - val_loss: 0.1471 - val_accuracy: 0.6765
Epoch 96/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0527
- accuracy: 0.7709 - val_loss: 0.1418 - val_accuracy: 0.6765
Epoch 97/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0511
- accuracy: 0.7530 - val_loss: 0.1475 - val_accuracy: 0.6765
Epoch 98/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0490
- accuracy: 0.7476 - val_loss: 0.1560 - val_accuracy: 0.5882
Epoch 99/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0507
- accuracy: 0.7062 - val_loss: 0.1563 - val_accuracy: 0.5294
Epoch 100/100
363/363 [==============================] - 2s 6ms/step - loss: 0.0544
- accuracy: 0.7204 - val_loss: 0.1466 - val_accuracy: 0.6765
```

Figure 7. Training process.

Mean absolute percentage error (MAPE) was used to verify the accuracy of the losses.

The results are shown in Table 2.

Table 2

Error for testing dataset

Measure name	Error (%)
Number of observations	26.3
Number of confirmed cases	26.5
Number of death	29.2
Number of recoveries	29.4

Because the result is influenced by a large number of factors, and the training took place only on a short history, the results are of high quality because the MAPE error does not exceed 30% with a prediction for all characteristics.

To work with new data, a ready-made data model and a ready-made MinMaxScaler for further correct alignment of variables relative to previous data is required. After completing the data normalization phase, the model is trained and then the function plt_result () is called to graphically display the accuracy of learning for each parameter, which are in separate columns.

To display graphic data on the map, you need to perform some pre-processing of data. To do this, use the function update_df_to_plt (), where an important parameter is "registration_area". This parameter is responsible for the area that will be displayed later.

Then the show_map () function is executed to display the data on the map. The result is shown in Fig. 8. The virus power is marked in different colors.

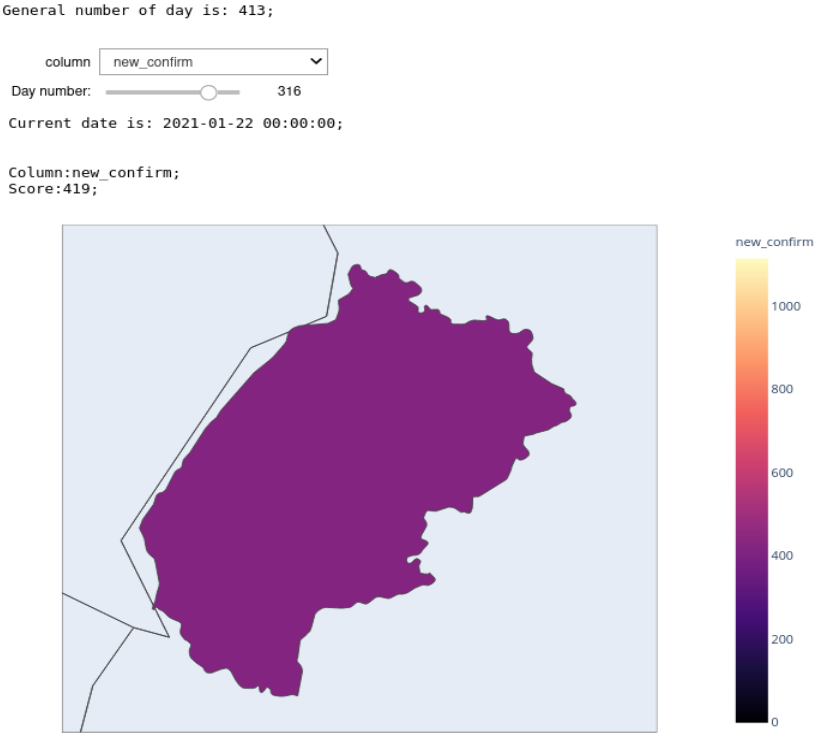


Figure 8. New confirmed cases for Lviv region

Conclusions

Different approaches to time data analysis were analyzed and the use of an RNN neural network, namely LSTM, was used. This choice was also justified by comparison with other approaches. Currently, this option is quite promising for predicting new cases of COVID-19, because it is not limited to the most rigorous type of problem.

During the development, a software product was built with a description of the system itself, taking into account the optimal software. A user guide for working with this product in different situations has also been described.

So, as a conclusion, this system performs the task well enough given the number of factors that affect in one way or another the result and this system is quite relevant for use.

5. Acknowledgements

This work is supported by National Foundation of Fundamental research, Ukraine, Project #103.01.0025.

6. References

[1] I. M. Hall, R. Gani, H. E. Hughes, and S. Leach, “Real-time epidemic forecasting for pandemic influenza,” Epidemiol. Infect., vol. 135, no. 3, pp. 372–385, Apr. 2007, doi: 10.1017/S0950268806007084.

[2] T. Tat Dat et al., “Epidemic Dynamics via Wavelet Theory and Machine Learning with Applications to Covid-19,” Biology, vol. 9, no. 12, p. 477, Dec. 2020, doi: 10.3390/biology9120477.

[3] “Prediction of epidemic trends in COVID-19 with logistic model and machine learning technics,” Chaos Solitons Fractals, vol. 139, p. 110058, Oct. 2020, doi: 10.1016/j.chaos.2020.110058.

[4] D. Zou, L. Wang, P. Xu, J. Chen, W. Zhang, and Q. Gu, “Epidemic Model Guided Machine Learning for COVID-19 Forecasts in the United States,” Epidemiology, preprint, May 2020. doi: 10.1101/2020.05.24.20111989.

[5] P. Gomes, R. Castro, “Wind speed and wind power forecasting using statistical models: autoregressive moving average (ARMA) and artificial neural networks (ANN)”, International Journal of Sustainable Energy Development, vol. 1, no. 1/2, pp 13-28, 2012.

[6] Haider, Abbas, “The COVID-19 Impact on Oil Market and Equity Market Link: An Evidence from ARMA-GJR GARCH-M Model”, Diss. CAPITAL UNIVERSITY, 2021.

[7] D. Benvenuto, M. Giovanetti, L., Vassallo, S. Angeletti, M. Ciccozzi, “Application of the ARIMA model on the COVID-2019 epidemic dataset”, Data in brief, vol. 29, p. 105340, 2020.

[8] F. Milani, “COVID-19 outbreak, social response, and early economic effects: a global VAR analysis of cross-country interdependencies”, Journal of population economics, vol. 34, no. 1, pp. 223-252, 2021.

[9] A. Howell, “Battling Burnout at the Frontlines of Health Care Amid COVID-19”, AACN Advanced Critical Care, vol. 32, no. 2, pp. 195-203, 2021.

[10] Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & San Tan, R. “A deep convolutional neural network model to classify heartbeats”, Computers in biology and medicine, 89, 389-396, 2017.

[11] Zaremba, W., Sutskever, I., & Vinyals, O. “Recurrent neural network regularization”, arXiv preprint arXiv:1409.2329, 2014.

[12] Donkers, Tim, Benedikt Loepp, and Jürgen Ziegler. "Sequential user-based recurrent neural network recommendations." Proceedings of the eleventh ACM conference on recommender systems. 2017.

[13] V. Piven, VasiaPiven/covid19_ua. 2021. Accessed: Apr. 26, 2021. URL:: https://github.com/VasiaPiven/covid19_ua

[14] Al Shorman, Amaal R., et al. "The Influence of Input Data Standardization Methods on the Prediction Accuracy of Genetic Programming Generated Classifiers." IJCCI. 2018.