EAI Endorsed Transactions
on Bioengineering and Bioinformatics Research Article **EAI.EU**

Automated Pipeline for Continual Data Gathering and Retraining of the Machine Learning-Based COVID-19 Spread Models

S. Baressi Šegota[1,*], I. Lorencin[1], N. Anđelić[1], D. Štifanić[1], J. Musulin[1], S. Vlahinić[1], T. Šušteršič[2,3], A. Blagojević[2,3], Z. Car[1]

[1]University of Rijeka, Faculty of Engineering, Vukovarska 58, 51000 Rijeka, Croatia
[2]Bioengineering Research and Development Center (BioIRC), Prvoslava Stojanovića 6, 34000 Kragujevac, Serbia
[3]University of Kragujevac, Faculty of Engineering, Sestre Janjić 6, 34000 Kragujevac, Serbia

Abstract

INTRODUCTION: The development of epidemiological curve models is one of the key factors in the combat of epidemiological diseases such as COVID-19.
OBJECTIVES: The goal of this paper is to develop a system for automatic training and testing of AI-based regressive models of epidemiological curves using public data, which involves automating the data acquisition and speeding up the training of the models.
METHODS: The research applies Multilayer Perceptron (MLP) for the creation of models, implemented within a system for automatic data fetching and training, and evaluated using the coefficient of determination (R^2). Training time is lowered through the application of data filtering and simplifying the model selection.
RESULTS: The developed system can train high precision models rapidly, allowing for quick model delivery All trained models achieve scores which are higher than 0.95.
CONCLUSION: The results show that the development of a quick COVID-19 spread modeling system is possible.

Keywords: Artificial Intelligence, Bio-engineering, Bio-inspired systems, Bio-inspired models, COVID-19, Epidemiology Curves, Machine Learning, Multilayer Perceptron

Received on 13 March 2021, accepted on 03 May 2021, published on 04 May 2021

doi: 10.4108/eai.4-5-2021.169582

1. Introduction

Coronavirus Disease 2019 (COVID-19) is a contagious disease, which results as an infection by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) [1]. After the first case was recorded over a year ago, COVID-19 has spread worldwide and has been declared a Public Health Emergency of International Concern in January 2020 by World Health Organization (WHO), with its status being raised to a pandemic in March 2020 [2]. Since then, many efforts have been made to combat the spread of COVID-19 across the world. Governments have issued restrictions on public activities, mandatory testing, and lockdowns [3]. Researchers across the world have attempted to assist in the combat against this disease in various ways – either through the development of spread models [4], vaccine development [5,6] or through the modeling of various influences the pandemic may have on society [7,8]. One of the tools that have shown high usability was artificial intelligence models – either for spread prediction [9] or for patient diagnosis [10]. Some examples of the research in modeling the COVID-19 spread follow.

[*]Corresponding author. Email: sbaressisegota@riteh.hr

Melin et al. (2020) [11] have used the method of self-organizing maps for the prediction of spatial spread relationships in the COVID-19 pandemic. The authors successfully grouped countries with similar behaviors, as one of the first papers considering the spatial relations in AI modeling, as opposed to time-wise ones. Rustam et al. (2020) [12] apply four different supervised learning algorithms – namely linear regression, least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), and exponential smoothing (ES) to develop COVID_19 future forecast models, for the number of infected, deceased and recovered patients. The goal of the research was to develop models for 10 days in advance. The results show that the ES achieves the highest scores, followed by LR and LASSO, with SVM performing poorly in the observed prediction task. Farooq and Bazaz apply a deep learning model, with the focus on the development of mortality reduction strategies, applying adaptive incremental learning for the model training. The authors conclude that lowering the number of deaths requires massively available vaccination or, lacking the previous, controlled natural immunization. Mollalo et al. (2020) [14] demonstrate the application of artificial neural network to regress the incidence rates of COVID-19 in the USA. Authors apply the MLP neural network and achieve a high-quality model with Getis-Ord Gi* ($\rho<0.05$). Another example of MLP being applied to the problem of COVID-19 spread prediction is by Car et al. (2020) [15], which applies MLP for the problem of regressing the values of confirmed, deceased, and recovered patients – without observing the epidemiology curve (active cases). The authors state and achieve two goals – developing a high-quality global regression model, and the determination of the optimal hyperparameter combinations. Narrowing the hyperparameter selection and the model creation methodology in the presented work will be based on that paper.

While many papers are presenting the modeling of the various aspects of the COVID-19 pandemic using AI-based methods, most papers do not consider the real-world application of the developed models. Namely the automation of re-training and result delivery, which will be the focus of the presented work.

One of the key issues with modeling of COVID-19 using AI-based algorithms is the time needed for the development of models. While self-learning, most Machine Learning (ML) models are complex to train due to the iterative process of model training, during which the internal parameters of the models are continually adjusted based on the error that is currently achieved by the model [16]. Due to these adjustments consisting of complex mathematical operations to determine the gradient of the neural network error being repeated for each data point time is of great concern [17]. Still, due to the unpredictable nature of COVID-19, the modeling process should be updated as often as possible with the newly collected data – as insights contained in the novel data may introduce information that may improve the models, enabling a higher precision. This may be achieved either using incremental learning paradigm (also known as online learning) [18] in which the model is retrained using newly acquired data or through the full retraining of the models with the entire dataset (including old, and newly acquired data) [19].

In the presented work the authors aim to test the following:

- Can the epidemiological curve be regressed with a satisfying precision from just the data containing the number of active cases?
- Can the process of regressing the epidemiological curve using AI be developed as the continual data gathering and retraining model?
- Can the process of training the ML models be adjusted based on the previous research to speed up the training processes, while still obtaining a high precision model?
- Does the retraining of algorithms with the entire dataset achieve better results than the incremental learning approach for the given case?

The goal of this research is to provide information on the most practical manner of model training in the case of pandemic spread modeling, as the knowledge of such an approach may prove important in the upcoming pandemics for early warning systems. The presented paper provides information on model training practices which will allow to achieve both higher precision epidemiological curves and to develop those models more quickly in case of future pandemics or epidemics.

2. Materials and Methods

In this section, the methodology of the work will be presented. First, the overview of the dataset and the description of data transformation will be given. This section will be followed by the description of the used MLP Regressor methodology, with the description of the automated pipeline being given at the end of the section.

2.1. Dataset

The data used in the presented research has been obtained from the “COVID-19” Data Repository made available by the Centre for Systems Science and Engineering (CSSE) at John Hopkins University (JHU), with support from the ESRI Living Atlas Team and the Johns Hopkins University Applied Physics Lab (JHU APL) [20]. The dataset contains time-series information for three patient groups – confirmed patients, recovered patients, and deceased patients. Starting date in the dataset is 22nd January 2020, and it has been updated daily since, by collecting information from various sources such as the

World Health Organization (WHO), European Centre for Disease Prevention and Control (ECDC), National Health Commission of the People's Republic of China (NHC), and others [21].

The data within the dataset is formatted as time-series. For each province within a country the cumulative number of confirmed, deceased, and recovered patients is marked for each date since the start of the dataset. Dataset collects information for 273 locations worldwide, within 82 countries. The data is regularly curated, with errata being given for data that may have not been precise in the past and has since been updated.

The main benefit of the JHU dataset is that it contains information for the three patient groups – confirmed, deceased, and recovered, which are the three values needed to calculate the number of active cases. An example of the data contained in the dataset is given in Table 1.

Table 1. An excerpt of the data contained in the JHU COVID-19 Dataset

Country	Lat.	Long.	22/01	23/01	24/01	…
Albania	41.2	20.1	0	0	0	
Algeria	28.0	1.6	0	0	2	
Andorra	42.5	1.5	0	0	0	…
Angola	-11.2	17.9	0	0	0	
…				…		

Epidemiology Curve

The epidemiology curve represents the number of active cases through time. Active cases are the group of patients who have been infected by the disease, and remain infected – in other words, not counting the patients that have recovered from the disease or passed away due to it. If the number of confirmed cases is given as N_C, deceased as N_D, recovered as N_R, then the number of active cases, N_A, can be given as [22]:

$$N_A = N_C - (N_R + N_D). \quad (1)$$

The number of active cases is a crucial piece of information for disease tracking as it provides information on the number of patients who are still capable of infecting other people, allowing for the further spread of the disease. The epidemiology curve obtained from the data within the JHU dataset, as well as the values of confirmed, deceased, and recovered patients through time, are given in Figure 1.

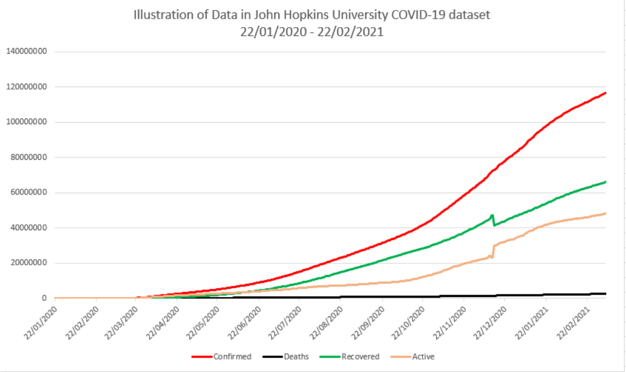

Figure 1. Visualization of Data within the JHU COVID-19 dataset, along with the epidemiology curve calculated from the presented data

Data processing

As previously mentioned, the dataset used in this research is made available in the time-series format. While good for observation, such a format is not appropriate for use with the regressive AI-based methods and needs to be transformed. The first step is the data-filtering. This step is crucial in the development of a continual data gathering and retraining pipeline, the reason being that a large amount of data raises the time necessary for the training [23]. By splitting the dataset, in this case by filtering it by each Country/Province, the individual training can not only be speed-up – but the option of parallelizing calculations for curves of separate countries is possible. This approach allows full utilization of High-Performance Computer (HPC) architectures which are massively parallel [24].

After the filtering, the data needs to be transformed into a regression dataset. This is done to enable the use of AI-based regression methods. For each of the countries, the latitude and longitude are recorded. Then, the number of days since the beginning of the dataset (22nd January 2020) is calculated, with the beginning of the dataset being 0. The corresponding number of patients is then written in a row with the corresponding latitude and longitude of the country the data was recorded in, as well as the number of days elapsed from the start of recording to the day the data was recorded. The example of the transformed dataset is given in Table 2. Note that the name of the country has been removed, as MLP models will take only numeric inputs of latitude and longitude.

In addition to modeling the numbers of confirmed, deceased, and recovered cases the number of active cases will be modeled as well. With modeling of the epidemiological curves in this manner, two ways can be used for the calculation of the number of active cases. First is taking the output from the previously mentioned three models, using Equation (1) – this will be referred to as the "derived" model. The second is to determine the epidemiology curve from the recorded data using Equation (1), and then modeling those values in the same manner as each one – this will be referred to as the "modeled" model. Both approaches will be used in the presented work. For the "modeled" model the data needs to be calculated in advance, and this is done by taking the same approach to transforming the dataset, except that the output is taken from all three datasets and calculated using Equation (1).

Table 2. An excerpt from the transformed dataset

Lat.	Long.	Days	Deaths
28.0	1.6	0	0
28.0	1.6	1	0
28.0	1.6	2	2
28.0	1.6	3	4
…	…	…	…

The four datasets prepared in this manner are now ready to be used for the training of MLP neural network models.

2.2. MLP Regressor

MLP Regressor is an artificial neural network consisting of the input layer, one or more hidden layers, and an output layer; each of which consists of one or more neurons [25]. Neurons in the subsequent layer are connected to ones in the previous layer through weighted connections and act as summators. The input layer consists of the number of neurons equal to the number of inputs – 3 in the presented research (latitude, longitude, and days elapsed). The output layer consists of a single neuron [26]. In cases, such as the presented work, where more than one value needs to be modeled, multiple neural networks need to be trained – with each one regressing a separate value [15].

One of the key issues in the timely development of MLP models, beyond the previously mentioned training process, is determining the optimal model architecture [27]. Model architecture is defined through the hyperparameters which define the number of hidden layers, the number of neurons per layer, solver, activation function of the neurons, learning rate as well as its type, and the regularization parameter value. Common practice is to set possible values of the hyperparameter values based on previous experience with similar problems and datasets and then test all possible combinations of the set hyperparameters – an approach known as grid search [28]. In previous research by the authors [15] a total of 48384 MLP Regressor models were created to determine the optimal model. Part of the paper's goal in that research was determining the optimal hyperparameters of the MLP networks. Due to that, the selected parameters for this research will be based on the ones determined in that paper, with the addition of a single larger network. The larger network, where larger stands for the higher number of neurons per layer, is used as a fallback. The larger networks have a higher chance of successfully regressing a complex task [29]. In general, larger networks of that kind are avoided for two reasons – training times and overfitting issues. In the case of the research presented in this paper overfitting, which is an issue where the network fits the data too well – without generalization, causing it to perform badly on newly introduced data, is addressed using 10-fold cross-validation [30, 31]. The time issue in training the larger network is addressed with a lower number of other possible hyperparameter values, making the model training faster. The hyperparameter values used for the re-training of the neural networks are given in Table 3.

From the above, it can be seen that the total number of the models trained, calculated as the product of the number of values per each hyperparameter, is 54. Such a low number of models, narrowed down by previous research, allows for a significantly faster training process.

Each of the above-mentioned models is trained on 75% of the data points for the given country. Then, the remaining 25% is used for the evaluation. Evaluation is performed using R^2. R^2 signifies the amount of variance that exists in two separate datasets – the real recorded data, and the predicted data in the presented research [32]. The higher R^2 value signifies a higher quality of regression – with the value of 1.0 signifying that all the variance is explained between the two datasets [33].

Table 3. The hyperparameter values used in the training process

Hyperparameter	Possible values
Hidden Layers	4
Neurons per layer	4,8,100
Learning Rate	0.01, 0.1, 0.5
Learning Rate Type	Constant, Adaptive
Solver	LBFGS
Activation Function	ReLU
L2 Regularization Value	0.01, 0.001, 0.0001

2.3. Incremental learning approach

Incremental learning (also referred to as online learning) is a technique that allows for the refitting of already trained AI-based models. This process can allow for previously trained models to be adjusted to newly acquired data without the need to re-train the entire model. This approach can be crucial in speeding up the processes of model adjustment. The goal of its use in the algorithm is to compare the speed and results of such an approach to the process of re-training of the entire model on the problem of COVID-19 spread prediction.

To simulate the online learning the network is trained with a subset of the data, not including the data for the last 30 days. The achieved model is then trained for 30 days, with each increment in learning happening daily, or in other words, the model being incrementally trained for new data daily. This is done to compare to the methodology of re-training the entire model and delivering the updated models daily. The results of the incremental model are then compared to the results of retrained model which has been trained at the end of execution.

2.4. Automated Pipeline for data acquisition, processing, training, and result delivery

The automated pipeline consists of four main parts:

- Data acquisition
- Processing
- Training
- Result delivery

This section will present each of these parts in turn, with examples of code used. The entirety of the code – consisting of a Bash script, and Python codes for transformation, training, and visualization are available in a Github repository [34].

Data acquisition

The data is acquired from the daily updated Github repository. This allows the use of Git to retrieve the data which has been updated by the repository owners. The automated data acquisition script will check if the directory containing the repository “COVID-19” exists. If such the directory does not exist the manuscript will clone the repository using version control software. The same approach is used if the directory exists, but the repository has not been cloned in it. This is checked by moving into the COVID-19 directory and using the command to check if the repository exists. The benefit of using version control software to acquire data instead of directly downloading it is that the local repository is only partially updated. Only the difference between the existing data in the repository and updated data is downloaded – greatly speeding up the process of data acquisition. On the same connection, the download of the entire repository using Git clone takes 182 seconds, downloading the data repository as a ZIP archive and unzipping it takes 178 seconds, and updating the manuscript with one day of new data takes 8.2 seconds (averaged over 10 runs). All timings in the manuscript were measured using the `time` command and are given as real times [35]. The script then extracts the CSV files containing the time-series data that will be used in the further steps of the manuscript. The data acquired in this manner can then be used in the further process of filtering and training.

Data processing and MLP training

Data is processed in the manner described in the previous section Data processing, applying the filtering and transformation as described, using a Python script that filters and transforms data. The script is executed, with the parallel processing of each training script. The parallelization to multiple cores is achieved with the internal functions of used libraries, which also serve as the basis for the implementation of hyperparameter search and testing the resulting models. The final step is visualization. The Python programs for training also store the models as a binary file, along with a text file containing the best performing hyperparameters and the achieved score of those hyperparameters.

Visualization

The script takes the stored models, as well as the dataset, and plots the entire domain using both the data in the original dataset used for modeling and the data obtained from the repository. Then, data is plotted for each day since the 22nd of January to establish the visual comparison of the real and modeled data. Then, the prediction is additionally calculated. By default, prediction is generated for 30 days in advance, but this can be adjusted within the visualization code. The same script also takes the R^2 value calculated during the testing and displays it on the generated graphs. The script generates all the graphs presented in the following section – Results. These graphs can then further be used for publications, webpages (directly if the used server is LAMP enabled) or to be automatically sent to people who can benefit from the information (epidemiologists, government officials, health experts, and others).

The complete overview diagram of the automated data acquisition system is given in Figure 2. The local repository is synced with the one available online – an action which is set to repeat daily – at noon CET, which is an hour later than the standard automated update time for the JHU COVID-19 repository; to allow time for potential delays. Of course, the script can be executed manually at any time by the system administrator. Then, the process of data extraction, filtering, transformation followed by model training, testing, and finally – visualization is performed.

2.4. Used hardware and software

The hardware used for training consists of the Intel Xeon Gold 6240R (2.4 GHz base clock, 24 cores), with 64GB of RAM. Each of the models is trained on 8 cores of the aforementioned CPU, to achieve equal utilization. Machine used for training is based on Linux Ubuntu Server 18.04 LTS (kernel version GNU/Linux),

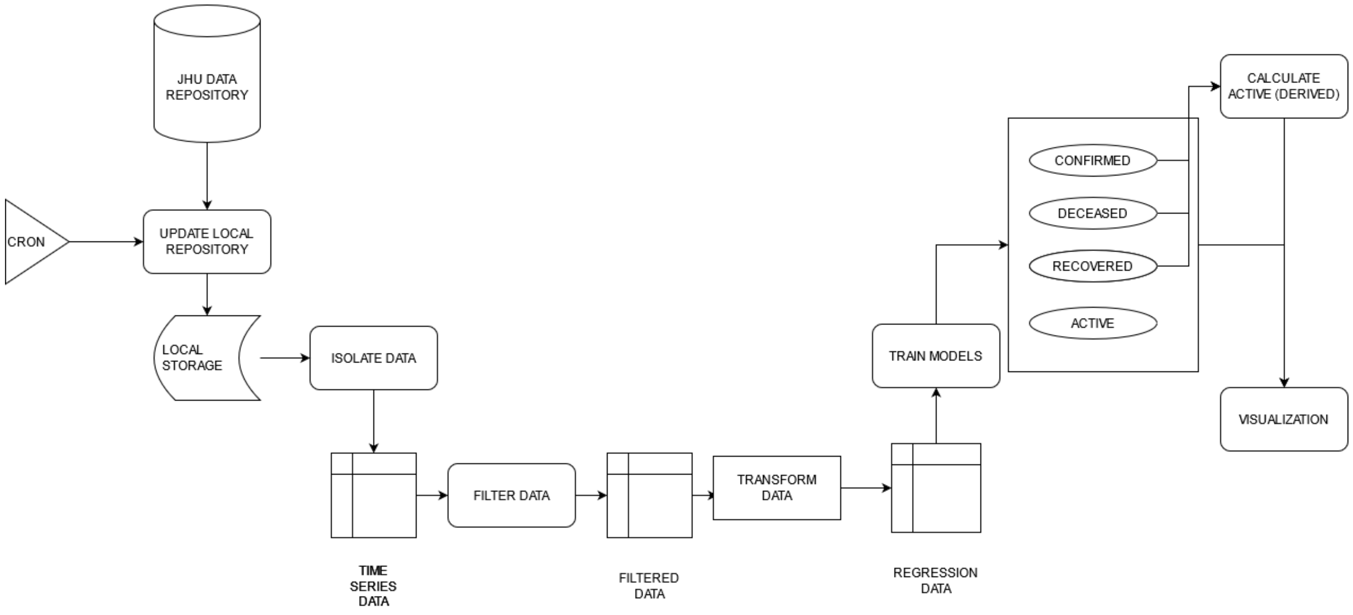

Figure 2. The illustration of the developed pipeline

3. Results

Table 3 demonstrates the average training times over 10 runs for each of the goals. The exception is the derived epidemiology curve which is calculated using the results from previous models). The R^2 scores of the models are also given in Table 3., also averaged over 10 runs.

Table 3. Execution times and R^2 scores of the trained models.

Goal	Training time [m] (N=10)	R^2 score (N=10)
Confirmed	56	0.99998
Recovered	62	0.98873
Deaths	54	0.99999
Epidemiology curve - modeled	55	0.99533
Epidemiology curve - derived	-	0.96052

For the generation of the images data for Croatia has been used, although the script can generate data for any Country in the dataset through the modification of the appropriate variable, defined in the script. For all the Figures 3-7. the modeled data is shown in a dashed line, with real data shown in a full line. The vertical line at x=410 represents the division between the end of the collected data and the 30-day prediction (note that only the predicted data is shown in the graph).

The generated figure for the model trend of the confirmed patients is given in Figure 3. As shown, this model has an extremely high fidelity – which is confirmed visually and through the highest achieved R^2 score.

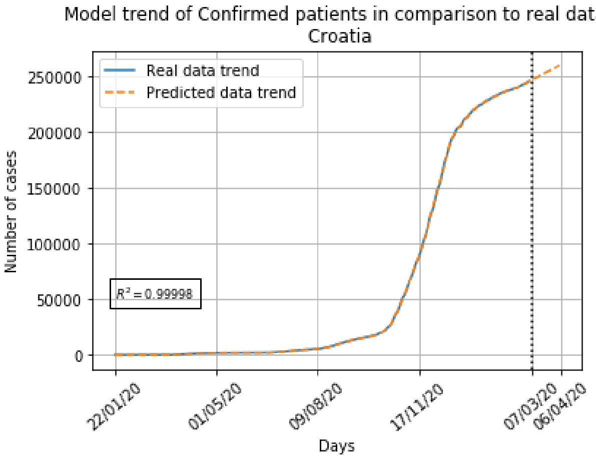

Figure 3. Modeled and predicted data for confirmed patients in Croatia.

In the case of recovered patients, it can be noticed that the regression quality is similarly high, but minor visual differences can be noticed on the test part of the dataset (beyond the 17th November). Still, the R^2 value beyond 0.99 shows that the regression quality is still high enough for the regression to be considered excellent.

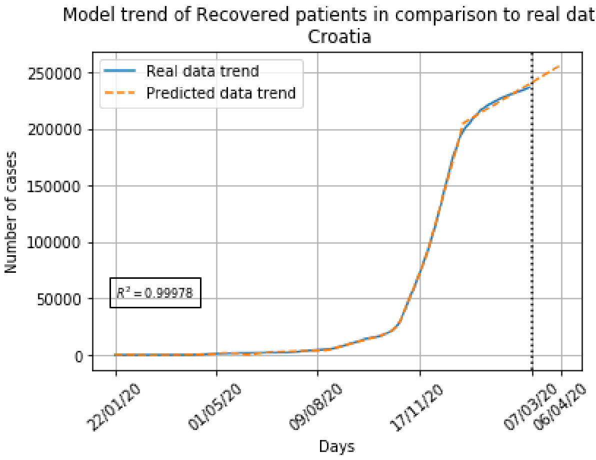

Figure 4. Modeled and predicted data for recovered patients

Figure 5 shows the model for deceased patients which has achieved the highest quality, with the R^2 value tending to one. Visual inspection shows that the test part of the dataset (beyond 17th November 2020) is almost equal for both the model and the real data, confirming the validity of the high R^2 value.

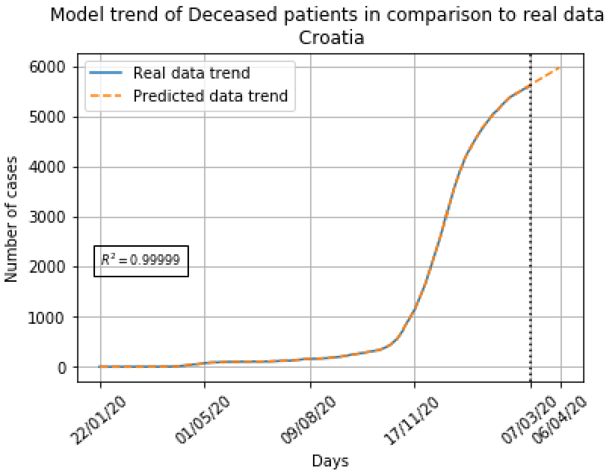

Figure 5. Modeled and predicted data for confirmed patients in Croatia

The modeled and real epidemiology curves are shown in Figure 6. Both the derived and modeled curves are shown, with the model curve being given as a dashed line and the derived curve as a dotted line.

The model curve has a higher regression quality – which can be evidenced by both a higher R^2 score (0.99533) and the visual observation. The derived curve has a lower regression quality – shown by both the lower R^2 score (0.96052), and visually. This is especially noticeable in the second half of the test part of the dataset – where the curve does not follow the real data, and the prediction widely differs from the probable situation which can be assumed from observing the current trend. This difference is probably caused because the errors in the individual models add together in the derived model, causing a larger, perceivable error. Another thing of note is that the visualization script does generate separate images for both the derived and modeled epidemiology curve – but for brevity, the combined image has been presented.

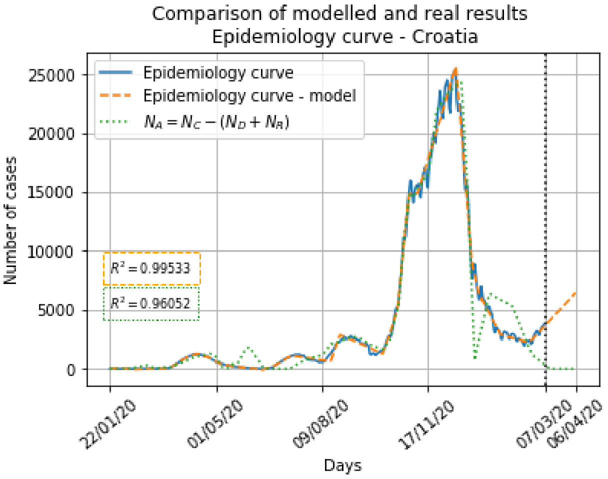

Figure 6. Modeled and predicted data for active patients (epidemiology curve) in Croatia

The final image that the script generates is the image that combines all the models. This is done for easier comparison between the different curves and a more condensed display. An example is given in Figure 7.

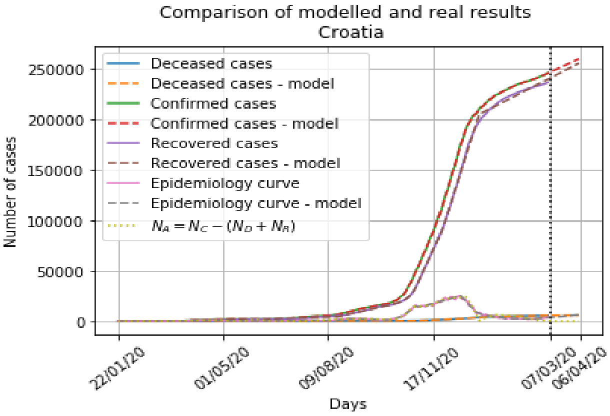

Figure 7. The complete plot of all the generated models

The comparison of scores to the cited research with the scores achieved with the used model – both with online training and the entire model re-training is given in Table 4. The online training model presents the scores which have been updated over the course of one month using re-fitting.

Table 4. R2 Score comparison

Model	Cited work [15]	Model re-training	Online training
Confirmed	0.940	0.999	0.989
Recovered	0.781	0.999	0.997
Deaths	0.986	0.999	0.999
Epidemiology model	-	0.995	0.980
Epidemiology derived	-	0.961	0.959

4. Discussion

The models achieve a high regression quality when using the limited grid search values based on the previous research. The main thing to note is that the training time has been lowered from multiple days in original research [15] using multiple HPC nodes, to below a single hour using a single compute node. The training time of the script is below 90 minutes – accounting for a full dataset download (in the case of the first run or the loss of the data), and the generation of visualizations. This would allow for a quick generation of the models and plots, with updated models being ready for delivery less than 2 hours after the new data has been uploaded to the online repository.

It should be noted how the derived model for the epidemiology curve has a significantly weaker regression than the separate model. For this reason, the modeling of the epidemiology curve should be done only directly, instead of deriving it – due to the errors of individual models stacking and making an apparent error in the derived model [36]. As the modeled version of the epidemiology curve has a higher regression quality it should be noted that if only the epidemiology curve is needed the model training can further be sped up by only directly modeling the number of active cases using more resources, e.g., a full CPU node instead of a quarter of it as it has been utilized in the presented research. While this approach has merits, the authors still propose and have considered in this discussion, the training of all four models. The reason for this is that while the epidemiology curve is sometimes only information needed, the curves for the number of confirmed, deceased, and recovered cases can provide important information. For example – for the same number of confirmed cases, high mortality and low recovery rates will result in a low number of active cases – as will the high recovery rate and low mortality. Obviously, the second case is preferable.

When comparing the results to online learning we can notice that this, incremental learning approach, achieves slightly worse results – but still within the bounds of acceptable models. The model refit times for incremental learning are negligible, so they have not been included.

5. Conclusions

The aim of the research paper has been fully accomplished. First, it was determined that the epidemiological curve can not only be regressed using the data from the dataset, but this approach provides better results in comparison to calculating it from other regressed values. In addition to that, the results show how the MLP algorithm can be applied in the continual data gathering and retraining methodology for the presented problem, through the development of an automated data acquisition, data processing, model training, model validation, and result visualization or through the utilization of online learning. The results also demonstrate that the models can be developed using this script in a relatively short amount of time. It can be noticed that online learning may be used as an alternative to model retraining while achieving comparable results even more quickly. Still, it should be noted that such an approach will not adjust the model to the errata that may be added into the dataset, which may lower the accuracy. A possible solution to this issue is monitoring the used data repository for errata, and only retraining the entire model when necessitated by data errors. The crucial part in developing a faster model training was the utilization of past research by the authors, which allowed for the significant lowering of the number of possible hyperparameter combinations.

Acknowledgements.

This research has been (partly) supported by the CEEPUS network CIII-HR-0108, European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS), project CEKOM under the grant KK.01.2.2.03.0004, CEI project "COVIDAi" (305.6019-20) and University of Rijeka scientific grants uniri-tehnic-18-275-1447 and "Development of intelligent systems for the prediction of medical and economical effect of COVID-19".

References

[1] Velavan TP, Meyer CG. The COVID-19 epidemic. Tropical medicine & international health. 2020 Mar;25(3):278.

[2] Sohrabi C, Alsafi Z, O'Neill N, Khan M, Kerwan A, Al-Jabir A, Iosifidis C, Agha R. World Health Organization declares global emergency: A review of the 2019 novel coronavirus (COVID-19). International journal of surgery. 2020 Apr 1;76:71-6.

[3] Béland LP, Brodeur A, Wright T. The short-term economic consequences of Covid-19: exposure to disease, remote work and government response.

[4] Tsori Y, Granek R. Epidemiological model for the inhomogeneous spatial spreading of COVID-19 and other diseases. PloS one. 2021 Feb 19;16(2):e0246056.

[5] Kim JH, Marks F, Clemens JD. Looking beyond COVID-19 vaccine phase 3 trials. Nature medicine. 2021 Jan 19:1-7.

[6] Lazarus JV, Ratzan SC, Palayew A, Gostin LO, Larson HJ, Rabin K, Kimball S, El-Mohandes A. A global survey of potential acceptance of a COVID-19 vaccine. Nature medicine. 2021 Feb;27(2):225-8.

[7] Štifanić D, Musulin J, Miočević A, Baressi Šegota S, Šubić R, Car Z. Impact of covid-19 on forecasting stock prices: an integration of stationary wavelet transform and bidirectional long short-term memory. Complexity. 2020 Jul 3;2020.

[8] Sahu P. Closure of universities due to coronavirus disease 2019 (COVID-19): impact on education and mental health of students and academic staff. Cureus. 2020 Apr;12(4).

[9] Mohamadou Y, Halidou A, Kapen PT. A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. Applied Intelligence. 2020 Nov;50(11):3913-25.

[10] Lorencin I, Baressi Šegota S, Anđelić N, Blagojević A, Šušteršić T, Protić A, Arsenijević M, Ćabov T, Filipović N, Car Z. Automatic Evaluation of the Lung Condition of COVID-19 Patients Using X-ray Images and Convolutional Neural Networks. Journal of Personalized Medicine. 2021 Jan;11(1):28.

[11] Melin P, Monica JC, Sanchez D, Castillo O. Analysis of spatial spread relationships of coronavirus (COVID-19) pandemic in the world using self organizing maps. Chaos, Solitons & Fractals. 2020 Sep 1;138:109917.

[12] Rustam F, Reshi AA, Mehmood A, Ullah S, On BW, Aslam W, Choi GS. COVID-19 future forecasting using supervised machine learning models. IEEE access. 2020 May 25;8:101489-99.

[13] Farooq J, Bazaz MA. A novel adaptive deep learning model of Covid-19 with focus on mortality reduction strategies. Chaos, Solitons & Fractals. 2020 Sep 1;138:110148.

[14] Mollalo A, Rivera KM, Vahedi B. Artificial neural network modeling of novel coronavirus (COVID-19) incidence rates across the continental United States. International journal of environmental research and public health. 2020 Jan;17(12):4204.

[15] Car Z, Baressi Šegota S, Anđelić N, Lorencin I, Mrzljak V. Modeling the spread of COVID-19 infection using a multilayer perceptron. Computational and mathematical methods in medicine. 2020 May 29;2020.

[16] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning. Cambridge: MIT press; 2016 Nov 18.

[17] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. New York: Springer series in statistics; 2001.

[18] He, J., Mao, R., Shao, Z., & Zhu, F. (2020). Incremental learning in online scenario. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 13926-13935).

[19] Furdek, M., & Natalino, C. (2020, March). Machine learning for optical network security management. In *2020 Optical Fiber Communications Conference and Exhibition (OFC)* (pp. 1-3). IEEE.

[20] "COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University". Accessed on 9th March 2021. Available at: https://github.com/CSSEGISandData/COVID-19

[21] Dong E, Du H, Gardner L. An interactive web-based dashboard to track COVID-19 in real time. The Lancet infectious diseases. 2020 May 1;20(5):533-4.

[22] Anđelić N, Baressi Šegota S, Lorencin I, Mrzljak V, Car Z. Estimation of COVID-19 epidemic curves using genetic programming algorithm. Health Informatics Journal. 2021 Jan;27(1):1460458220976728.

[23] Bhavsar H, Ganatra A. A comparative study of training algorithms for supervised machine learning. International Journal of Soft Computing and Engineering (IJSCE). 2012 Sep;2(4):2231-307.

[24] Hassan HA, Kashkoush MS, Azab M, Sheta WM. Impact of using multi-levels of parallelism on HPC applications performance hosted on Azure cloud computing. International Journal of High Performance Computing and Networking. 2019;13(3):251-60.

[25] Feng X, Ma G, Su SF, Huang C, Boswell MK, Xue P. A multi-layer perceptron approach for accelerated wave forecasting in Lake Michigan. Ocean Engineering. 2020 Sep 1;211:107526.

[26] Seo H, Cho DH. Cancer-Related Gene Signature Selection Based on Boosted Regression for Multilayer Perceptron. IEEE Access. 2020 Apr 3;8:64992-5004.

[27] Huuhtanen T, Jung A. Anomaly Location Detection with Electrical Impedance Tomography Using Multilayer Perceptrons. In2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP) 2020 Sep 21 (pp. 1-6). IEEE.

[28] Al-Fugara AK, Ahmadlou M, Al-Shabeeb AR, AlAyyash S, Al-Amoush H, Al-Adamat R. Spatial mapping of groundwater springs potentiality using grid search-based and genetic algorithm-based support vector regression. Geocarto International. 2020 Jan 24:1-20.

[29] Anil R, Pereyra G, Passos A, Ormandi R, Dahl GE, Hinton GE. Large scale distributed neural network training through online distillation. arXiv preprint arXiv:1804.03235. 2018 Apr 9.

[30] Rice L, Wong E, Kolter Z. Overfitting in adversarially robust deep learning. InInternational Conference on Machine Learning 2020 Nov 21 (pp. 8093-8104). PMLR.

[31] Xiong Z, Cui Y, Liu Z, Zhao Y, Hu M, Hu J. Evaluating explorative prediction power of machine learning algorithms for materials discovery using k-fold forward cross-validation. Computational Materials Science. 2020 Jan 1;171:109203.

[32] Nagelkerke NJ. A note on a general definition of the coefficient of determination. Biometrika. 1991 Sep 1;78(3):691-2.

[33] Anđelić N, Baressi Šegota S, Lorencin I, Jurilj Z, Šušteršič T, Blagojević A, Protić A, Ćabov T, Filipović N, Car Z. Estimation of COVID-19 Epidemiology Curve of the United States Using Genetic Programming Algorithm. International Journal of Environmental Research and Public Health. 2021 Jan;18(3):959.

[34] "COVID-19 MLP Repository". Accessed on 22nd April 2021, available at: https://github.com/RitehAIandRobot/COVID-19-MLP

[35] Åsberg M, Nolte T, Perez CM, Kato S. Execution time monitoring in linux. In2009 IEEE Conference on Emerging Technologies & Factory Automation 2009 Sep 22 (pp. 1-4). IEEE.

[36] Clarke B. Comparing Bayes model averaging and stacking when model approximation error cannot be ignored. Journal of Machine Learning Research. 2003;4(Oct):683-712