Journal of
Data Acquisition and Processing
ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7716473

AN ADAPTIVE TECHNIQUE OF COVID-19 PATIENT COUNT PREDICTION

Kishore Mishra,Dr Akash Saxena, Pawan Agarwal

Compucom Institute of Technology and Mangement,Jaipur

ABSTRACT: The virus that causes Covid-19 is quickly spreading around the globe. Approximately every three to four times, waves are created that have a significant effect on people's lives. It's unclear if this situation involves a lack of effective identification of covid illnesses or any other illness. There is no reliable statistics on the total number of covid patients in the nation, and no system exists to track them. This prevents people from receiving necessary care and services. The number of patients in a given dataset may be determined with more precision using AI methods. In this study, we provide an adjustable method for anticipating the number of patients in Covid-19. A piece of software called python spyder is used to run the simulation.
KEYWORDS: Covid, Patient, prediction, dataset, Artificial intelligence.

I. INTRODUCTION

The novel Corona virus SARS-CoV-2 was first seen in December 2019 and is expected to kick off a pandemic of respiratory infection known as Coronavirus in 2020. This virus has already proven itself to be a dangerous pathogen, with the potential to progress from mild to severe, posing a threat to organ disillusionment and even death. There is good reason to be very concerned about the consequences of this viral contamination, given the spread of the pandemic and the increasing number of the altered instances and patients who develop significant respiratory discontent and cardiovascular snares. Reasonable approaches to preventing and resolving Coronavirus-related concerns at the negotiating table have been given considerable study. While fighting an epidemic, researchers and leaders face another enormous challenge: managing the vast amounts of data generated in the process.

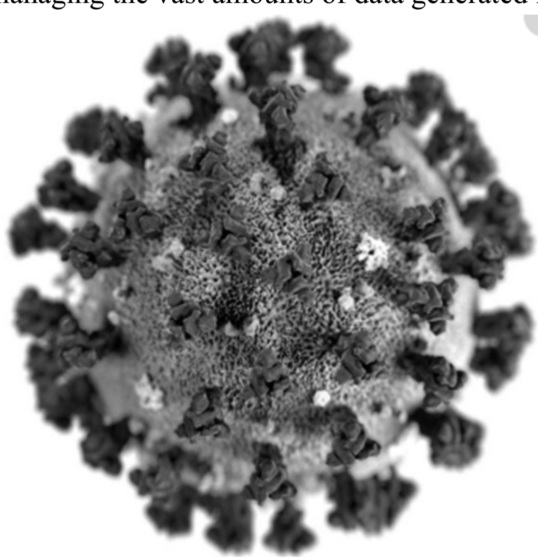


Figure 1: Corona virus

There is a family of RNA viruses called coronaviruses that may infect both mammals and birds. They spread easily between birds and people, and may cause anything from a little cold to death. Some instances of the common cold in humans (which is also caused by other viruses, mostly rhinoviruses) are quite mild, but more severe cases may be caused by SARS, MERS, and COVID-19. Bovine and porcine diarrhoea is caused by these viruses, whereas mouse hepatitis and encephalitis are caused by them. Human coronavirus infections cannot be prevented or treated since no vaccinations or antiviral medications exist at this time.

II. METHODOLOGY

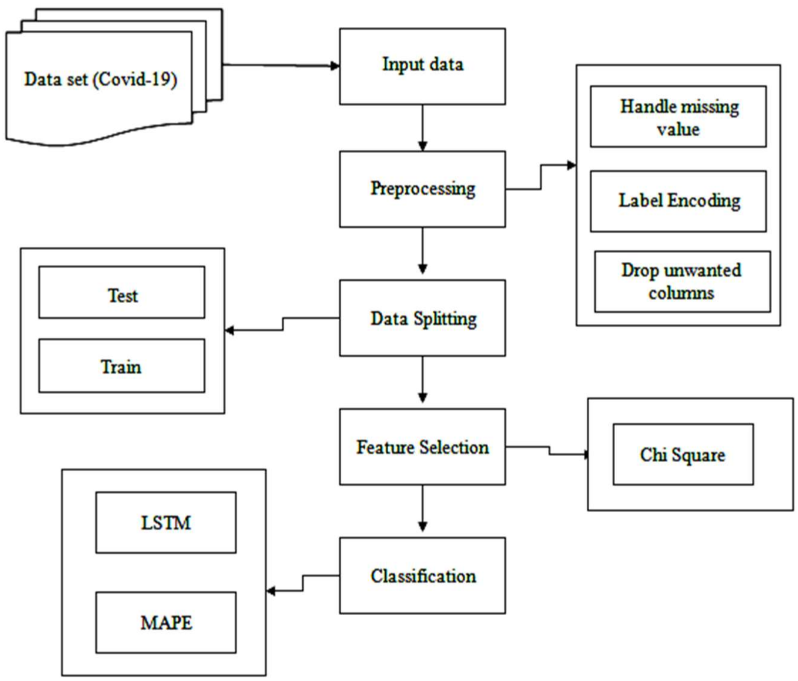


Figure 2: Flow Chart

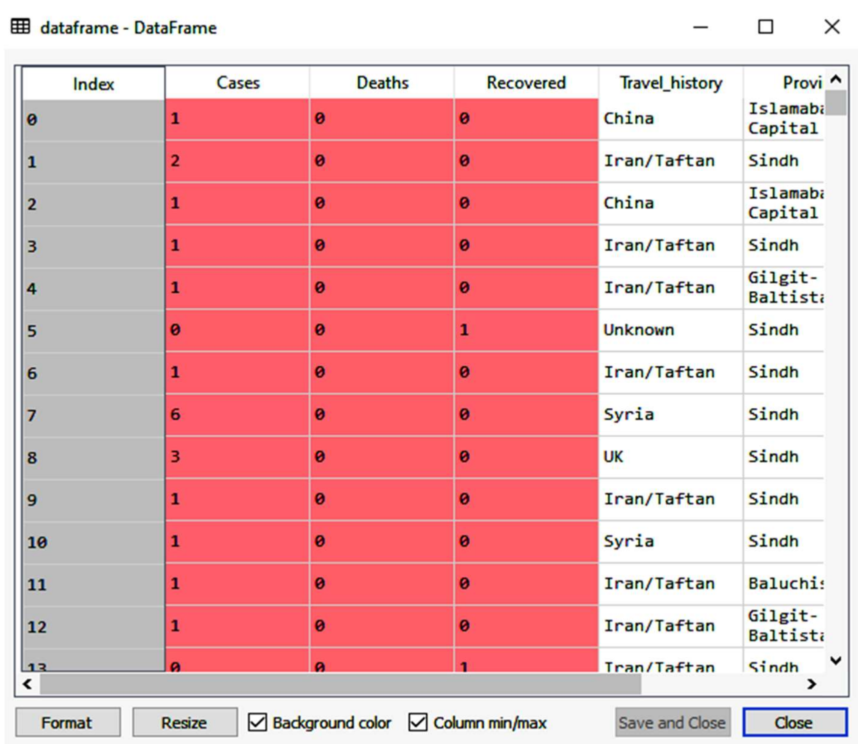
Steps-

1. The first step is to get the covid 19 patient dataset from kaggle, a source of huge datasets and a repository for machine learning.
2. Now, use data preparation techniques, such as filling in blanks, label encoding, and removing columns that aren't needed.
3. The dataset has been partitioned into a training and testing set. Seventy percent of the information is used for training, while the remaining thirty percent is put to the test.
4. Use the dataset's feature selection to construct a forecast.
5. Use a machine learning technique based on long short-term memory (LSTM) to do categorization.
6. Examine and compute the efficiency indicators.

Recurrent neural networks, of which LSTM-Long-Short Term Memory (LSTM) is a subtype, excel in memory. Because of their superior ability to memorise specific patterns, LSTMs often outperform other methods. Like other NNs, LSTM may contain a number of hidden layers, with just the useful information being retained and the rest being destroyed at each cell's iteration. A typical LSTM unit consists of a cell, input gate, output gate, and forget gate. The three gates control the entry and exit of data into and out of the cell, and the cell may store values for an indefinite amount of time. Since there may be gaps of uncertain length between crucial events in a time series, LSTM networks excel in classifying, processing, and making predictions from such data. Long short-term memories (LSTMs) were created as a solution to the vanishing gradient issue that might occur when training regular RNNs. A major benefit of LSTM over RNNs, hidden Markov models, and other sequence learning approaches is its tolerance for gaps of varying lengths.

III. SIMULATION RESULTS

The suggested approach is coded in python spyder 3.7. We are able to take use of the spyder environment thanks to libraries like sklearn, numpy, pandas, matplotlib, pyplot, seaborn, and os.



Index	Cases	Deaths	Recovered	Travel_history	Provi
0	1	0	0	China	Islamaba Capital
1	2	0	0	Iran/Taftan	Sindh
2	1	0	0	China	Islamaba Capital
3	1	0	0	Iran/Taftan	Sindh
4	1	0	0	Iran/Taftan	Gilgit-Baltista
5	0	0	1	Unknown	Sindh
6	1	0	0	Iran/Taftan	Sindh
7	6	0	0	Syria	Sindh
8	3	0	0	UK	Sindh
9	1	0	0	Iran/Taftan	Sindh
10	1	0	0	Syria	Sindh
11	1	0	0	Iran/Taftan	Baluchis
12	1	0	0	Iran/Taftan	Gilgit-Baltista
13	0	0	1	Iran/Taftan	Sindh

Figure 3: Dataset

The KDD dataset is shown in Figure 3. There are a total of 1157 records in this dataset, organised into the following seven columns: Date, Cases, Deaths, Recovered, Travel history, Province, City.

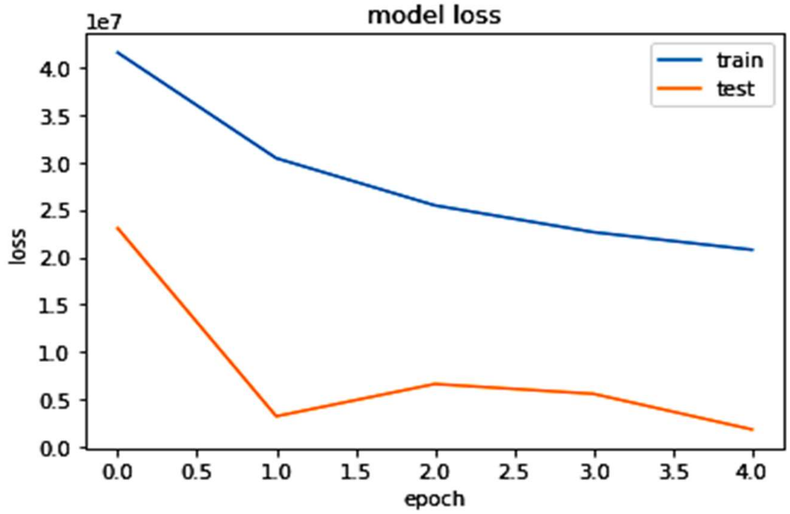


Figure 4: Validation model (Epoch=5)

In figure 4, we see the validation model at epoch 5, together with the corresponding training and testing loss. As can be seen in the diagram, the testing model has lower model loss. As a result, the suggested paradigm is more successful..

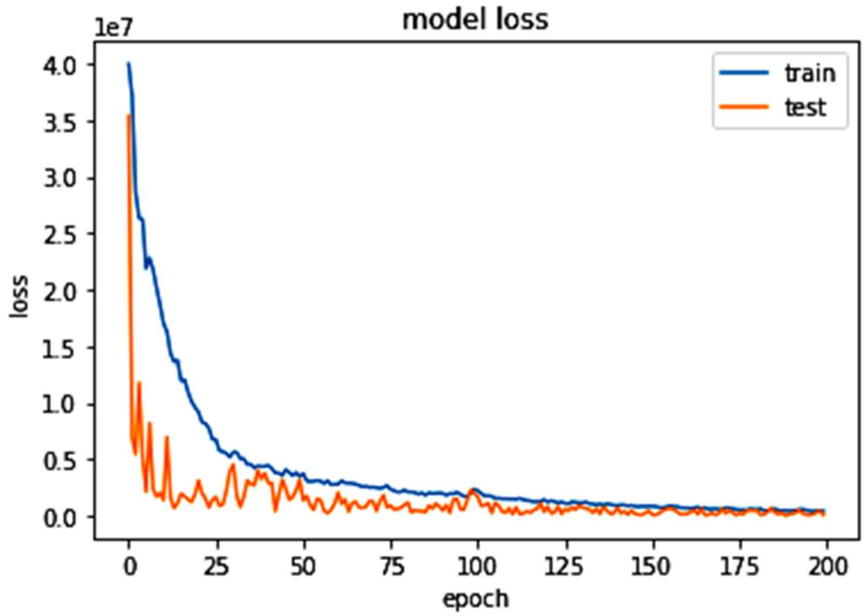


Figure 5: Validation model (Epoch=200)

Figure 5 is showing the validation model at the variable epoch, here epoch value is 100 and 200 respectively.

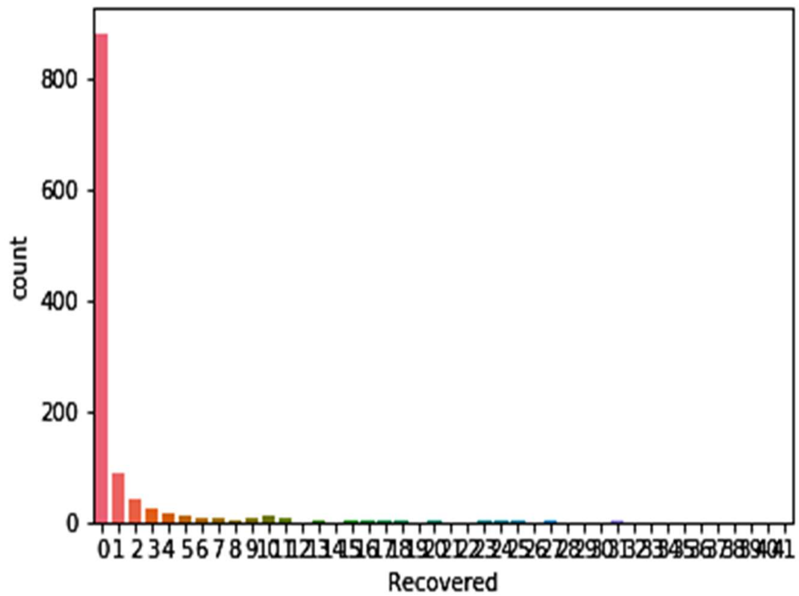


Figure 6: Visualization of patient count

Figure 6 is presenting the visualization of the covid-19 patient count result graph. It is clear from this graph, various patient count with respect to the recovered patient is optimizing.

Table 1: Result Comparison

Sr. No.	Parameters	Previous Work [1]	Proposed Work
1	LSTM Unit	15	10-50
2	Epochs	Upto 100	Upto 200
3	Batch Size	Upto 15	Upto 200
4	Features	NA	Original 5, Reduced 3
5	Mean Percentage Absolute Error	49%	30.55%

IV. CONCLUSION

This paper presents an adjustable method for anticipating the number of patients in Covid-19. Python's spyder environment is used to develop long short term memory (LSTM) methods.

The dataset used here was downloaded from the machine learning competition website kaggle.com. The performance of the suggested method is superior than that of previous efforts. We consider LSTM counts between 10 and 50 in the suggested model. Up to 200 people may be added to a batch. There may be as many as 200 epochs, and the mean percentage absolute error is now 30.55 percent, down from 49 percent in the prior model. The simulation results of the current work and the proposed work show that the suggested work significantly improves upon the existing work.

REFERENCES

1. M. Iqbal et al., "COVID-19 Patient Count Prediction Using LSTM," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 974-981, Aug. 2021, doi: 10.1109/TCSS.2021.3056769.
2. S. Bhardwaj, H. Bhardwaj, J. Bhardwaj and P. Gupta, "Global Prediction of COVID-19 Cases and Deaths using Machine Learning," 2021 Sixth International Conference on Image Information Processing (ICIIP), 2021, pp. 422-426, doi: 10.1109/ICIIP53038.2021.9702560.
3. Z. Alhakeem and H. Hakim, "Mathematical and Engineering Solutions Facing COVID-19 Pandemic: A Review," 2021 7th International Conference on Signal Processing and Communication (ICSC), 2021, pp. 200-204, doi: 10.1109/ICSC53193.2021.9673173.
4. D. Wang, X. Chen, L. Zhang, Y. Fang and C. Huang, "A Blockchain based Human-to-Infrastructure Contact Tracing Approach for COVID-19," in IEEE Internet of Things Journal, doi: 10.1109/JIOT.2021.3138971.
5. Y. Chang, Z. Ren and B. W. Schuller, "Transformer-based CNNs: Mining Temporal Context Information for Multi-sound COVID-19 Diagnosis," 2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2021, pp. 2335-2338, doi: 10.1109/EMBC46164.2021.9629552.
6. P. Ghose, U. K. Acharjee, M. A. Islam, S. Sharmin and M. A. Uddin, "Deep Viewing for Covid-19 Detection from X-Ray Using CNN Based Architecture," 2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2021, pp. 283-287, doi: 10.23919/EECSI53397.2021.9624257.
7. F. Mohsin and W. Elmedany, "A Secure Internet of Healthcare Things for tackling COVID-19," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), 2021, pp. 98-104, doi: 10.1109/3ICT53449.2021.9581819.
8. C. S. Sumanth and R. K. Nayak, "Machine Intelligent Techniques for COVID-19 Detection: A Critical Review and Analysis," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, pp. 01-08, doi: 10.1109/ICCCNT51525.2021.9579858.
9. T. Rahman et al., "Development and Validation of an Early Scoring System for Prediction of Disease Severity in COVID-19 Using Complete Blood Count Parameters," in IEEE Access, vol. 9, pp. 120422-120441, 2021, doi: 10.1109/ACCESS.2021.3105321.

10. L. Famiglini, G. Bini, A. Carobene, A. Campagner and F. Cabitza, "Prediction of ICU admission for COVID-19 patients: a Machine Learning approach based on Complete Blood Count data," 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS), 2021, pp. 160-165, doi: 10.1109/CBMS52027.2021.00065.
11. Sharma, J. Li, D. Mishra, G. Batista and A. Seneviratne, "Passive WiFi CSI Sensing Based Machine Learning Framework for COVID-Safe Occupancy Monitoring," 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021, pp. 1-6, doi: 10.1109/ICCWorkshops50388.2021.9473673.
12. S. Thabasum Aara, A. Pandian K, T. S. Sai Kumar and A. Prabalakshmi, "A Novel Convolutional Neural Network Architecture to Diagnose COVID-19," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 595-599, doi: 10.1109/ICSPC51351.2021.9451701.