IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 9, Issue 7, July 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.542

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

Data Visualization Model and Outlier Detection for COVID-19

Abhishek Koushik B N, Akash Rotti, Amith Vishnu, Anandteerth, Dr Asha T

UG Student, Dept. of CS&E., Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore, India.

UG Student, Dept. of CS&E., Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore, India.

UG Student, Dept. of CS&E., Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore, India.

UG Student, Dept. of CS&E., Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore, India.

Professor and Head, Dept. of CS&E., Bangalore Institute of Technology, VV Puram, K R Pura, Bangalore, India.

ABSTRACT: COVID-19 has caused disruption in various walks of life. The pandemic has been disruptive in all industries including the Software industry. It is important to exercise extreme caution while living with the pandemic-affected world. The pandemic often spikes abruptly with respect to the number of cases reported per day due to various socio-economic reasons. Abrupt spikes are often due to the development / mutation of the virus into a newer version and is most often than not followed by stricter regimen in the society. As the pandemic is a novel problem which has risen recently, there are no existing systems which could help us during our project, but we cannot ignore the invaluable help of the various machine learning algorithms. The machine learning models can be used to learn the regular trend of the virus spread. When a new data point arrives, it is passed through the trained model, and it classifies the data as Outlier/Non-Outlier.

The objective of our project is to detect these abrupt spikes and analyze them against various socio-economic factors and determine when these spikes happened, allowing the health authorities and the municipalities to take the necessary action. It is also our objective to notify the authorities whenever a spike happens in that region. We train Prophet Model for the data as they are better at capturing non-linearity in data. Prophet Model has many tunable parameters which in turn give a lot of flexibility for training. The project solves the problem of misinterpreted preventive measures and allows the authorities ample time to plan their course of action.

KEYWORDS: Prediction, Anomaly Detection, Outlier, Prophet Model, Visualization, Machine Learning

I. INTRODUCTION

The year 2020 has been a disruptive year in all walks of life. The COVID-19 pandemic has brought about a lot of changes in the software industry and other departments of everyday life. This project attempts to solve one of the most glaring problems in the post pandemic world that is, detection and prevention of the spread of the virus.

The COVID-19 virus / pandemic broke out in the year 2020 across all major parts of the world, bringing the world to a standstill, both socially and economically. The world faced one of its worst economic crises during these past 10 months. The detection of this virus became a most important aspect in the fight against it since the virus is found to be a heavily mutable virus. Once the virus is detected in an individual, he needs to be isolated, which leads to isolation of entire neighbourhoods in many cases.

Data sets will almost always have outlier values (points significantly outside the range of the rest of the data), but it's not always included in analysis. This is because outliers can be the result of a simple typo or the result of an extraordinary event happening. It's important to look at outliers to understand and decide if it should be included or excluded from the analysis.

Our problem is to find outliers in COVID data. The outlier in question is a spike in the number of cases on a particular day. We plan to detect these outliers in the data using Machine Learning. This project attempts to solve one of the most glaring problems in the post pandemic world that is, detection and prevention of the spread of the virus.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

II. RELATED WORK

An existing system which is like this project is Arogya Setu. Aarogya Setu is an Indian COVID–19 "contact tracing, syndromic mapping and self-assessment" digital service, primarily a mobile app, developed by the National Informatics Centre under the Ministry of Electronics and Information Technology.

The Drawbacks are:

- Primarily based on Proximity data and not aggregated data.
- Has very high-level aggregated data such as Total cases in Country, State, Etc.
- Does not have diverse visualizations
- Does not make predictions

III. OBJECTIVES

- To Collect COVID-19 data and store in a suitable form.
- To pre-process and extract useful features from data.
- To train a Machine Learning Model on these extracted features.
- To Classify the data-point as an Outlier/Non-Outlier.
- To evaluate model-performance and improve it.
- To perform hyper parameter tuning on components of model.

IV. PROPOSED SYSTEM

Our aim is to detect outliers in the official dataset for the COVID-19 pandemic. We use Machine learning techniques like Prophet Model to detect abrupt spikes in the dataset, thus learning about the trend of the spread of the virus in different parts of the world.

V. PROJECT MODULES

A.*Dashboard Module*

The dashboard module provides a tabular view of the number of cases, both old and the cases of that day, filtered out by both country and state.

B.*India – Time Race Visualization Module*

This page gives a live view of how the number of cases has progressed as time passed since the very first case was recorded.

C.*India – Trajectory view Module*

The trajectory gives a graphical view of the trajectory followed by the virus, i.e., how the number of cases has progressed since the previous day.

D. *India Map Module*

This module gives a cartographical view of the number of cases over the country, filtered by state, identifiable by the position of said state on the map.The user can get a deeper view of the state by clicking on the state.

E. *India – Heat Map Module*

The heat map gives the heat signatures based on the number of cases in a particular state on the map of India.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

F. *India – Gender Module*

The gender view provides an insight into the number of cases in the country, filtered by gender.

G. *India – Containment Zone View Module*

This module gives a view of the containment zones in the country, and the map of India is divided into red, orange, and green zones.

H. *India – Tree Map Module*

The tree map contains a planar state-wise view of the number of cases in each state, allocating the largest area to the state having the greatest number of cases on that day.

I.*Global – World Map Module*

The world map gives a country-wise view of the number of cases, i.e., the number of cases in each country on that day.

J.*Global – Trajectory Module*

The trajectory view in the global section gives a graphical view of how the number of cases has progressed, i.e., how the number of cases today compares to that of yesterday.

K.*Global – Symptoms Module*

The symptoms module gives a statistical insight into the most probable symptoms one might experience due to the pandemic.

L.*Predictions – Forecast Module*

The forecast module predicts how the virus may spread in the forthcoming 365 days and gives an upper and lower limit to inculcate a margin of error. This is the outcome of a machine learning model trained based on the current number of cases.

M.*Predictions – Anomaly Detection Module*

The anomaly detection module gives a graphical idea of the anomalies, i.e., wherever an abrupt spike or a downfall in the number of cases is detected. The datapoint is classified as anomaly or not based on the upper limit and lower limits provided by the forecast module.

N.*Vaccination – State-wise comparison Module*

The state-wise vaccination module gives a state-wise view of the number of people who have been vaccinated to counter the pandemic and people who have not been vaccinated.

O.*Vaccination – Current Vaccination State Module*

The current vaccination module gives various graphs visualizing the number of people who have been vaccinated on that day.

P. *Vaccination – Cowin Module*

This module gives a visualization of the total number of vaccination doses administered to the people of the country. It also gives a comparison between the different vaccines, i.e., how many doses of that vaccine have been administered.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

VI. SYSTEM ARCHITECTURE

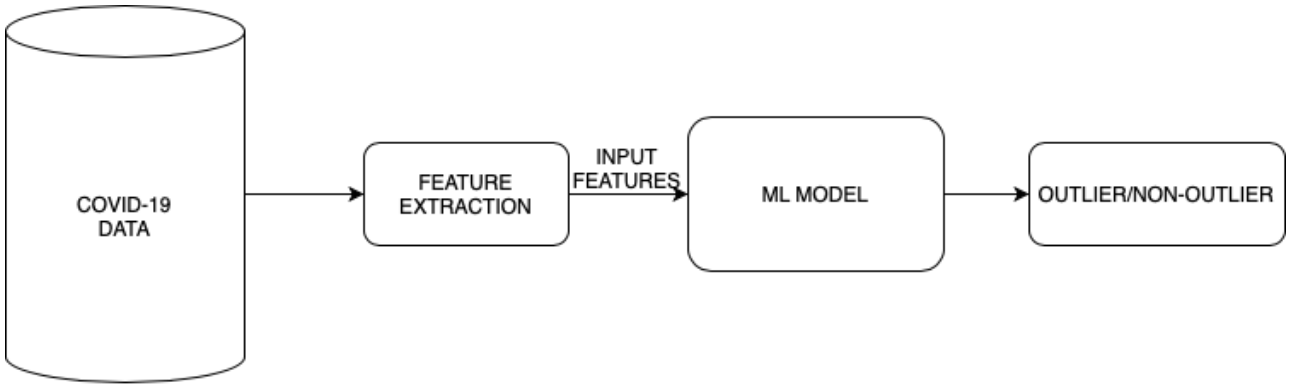

Figure-1Data Flow

The above diagram shows the data flow of the system. The COVID-19 raw data is stored in a database, and it passed through a Feature Extractor. Among all the features in the dataset, only few features are relevant for the prediction. This phase selects the relevant features from the data to maximize the prediction accuracy. These features are then passed to a Machine Learning Model which is trained on large amounts of training data. It uses the weights and parameters it learned to make a prediction for the new data point. Based on the measures of probability, the data point is classified as an Outlier or an Inlier.

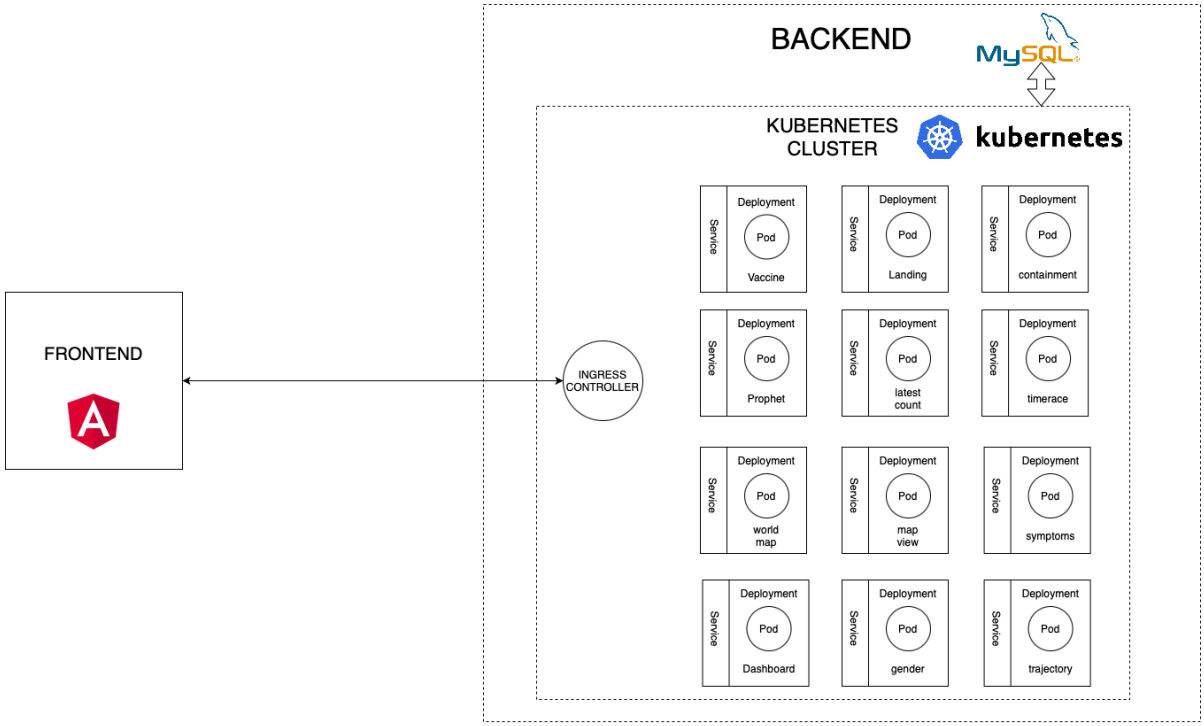

Figure-2 System Architecture

The project is build using Cloud Native Principles like Containers and Kubernetes. Each service is built as an independent microservice. For each microservice, a Kubernetes deployment is created which creates 1 or more pods depending on demand and then a service is created to expose the service to an API endpoint.

The API is exposed to the public using a Kubernetes Ingress Controller in the Data Centre Edge. This also performs the task of load balancing the requests to the specific pods. Each of the pods have access to a shared MySQL database.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

The frontend is built using Angular, A TypeScript Framework which sends HTTP requests to the Ingress Controller and gets data in response.

VII. ALGORITHM

Prophet Forecasting Model

- We use a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + t.$$

- Here g(t) is the trend function which models non-periodic changes in the value of the time series, s(t) represents periodic changes (e.g., weekly, and yearly seasonality), and h(t) represents the effects of holidays which occur on potentially irregular schedules over one or more days. The error term t represents any idiosyncratic changes which are not accommodated by the model; later we will make the parametric assumption that t is normally distributed.
- This specification is like a generalized additive model (GAM), a class of regression models with potentially non-linear smoothers applied to the regressors. Here we use only time as a regressor but possibly several linear and non-linear functions of time as components. Modelling seasonality as an additive component is the same approach taken by exponential smoothing. Multiplicative seasonality, where the seasonal effect is a factor that multiplies g(t), can be accomplished through a log transform.
- We are, in effect, framing the forecasting problem as a curve-fitting exercise, which is inherently different from time series models that explicitly account for the temporal dependence structure in the data. While we give up some important inferential advantages of using a generative model such as an ARIMA, this formulation provides a few practical advantages:
  1) Flexibility: We can easily accommodate seasonality with multiple periods and let the analyst make different assumptions about trends.
  2) Unlike with ARIMA models, the measurements do not need to be regularly spaced, and we do not need to interpolate missing values e.g., from removing outliers.
  3) The forecasting model has easily interpretable parameters that can be changed by the analyst to impose assumptions on the forecast.
  4) Fitting is very fast, allowing the analyst to interactively explore many model specifications, for example in a Shiny application.

VIII. PSEUDO CODE

- Fit the Prophet model for the currently available data
- Generate forecasted cases for desired period using the fit model
- Each point generated by the model will have an Upper Bound (y^ upper) and Lower Bound (y^ lower)
- Compare the actual value(fact) with predicted bounds
- If fact>y^ upper or fact<y^ lower

  Mark the point as an outlier else

  Mark the point as an inlier

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

- Plot the points and color the Outlier points with red to distinguish from the Inlier points

IX. RESULTS

- The application predicts how the trend of the corona virus may look in the future (up to 365 days).
- The application also labels anomalies in the dataset.
- The application gives different visualizations of the trend in the virus.
- The predictions obtained from the application are accurate and comparable to the actual number of cases.

Date	Upper limit	Lower limit	Actual values of cases
03-May-21	44,718	41,567	44,438
04-May-21	45,326	42,394	44,361
31-Mar-21	4,843	1,768	4,225
22-Apr-21	27,543	24,466	25,795
14-Oct-20	9,710	6,678	9,265

Table-1 Results

X. CONCLUSION, APPLICATIONS AND FUTURE WORK

A.*Conclusion*

The proposed system fits a Prophet Outlier Detection Algorithm using the current data and attempts to forewarn the concerned authorities by detecting as early as possible an abrupt spike in the number of cases in a day. This gives the authorities ample opportunity to take the necessary actions.

B.*Applications*

- To understand trends in virus spread.
- To understand the reasons for the spike in the number of cases on a particular day.
- The predictions give a before-hand idea of the trend of the spread of the corona virus, thus giving ample time to prepare for any untoward situations.
- The anomalies help us to understand the socio-economic reasons causing the abrupt spike/downfall in the number of cases.
- The different visualizations of the dataset give us a demographic and semantic view of the dataset, helping us

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 7.542

|| Volume 9, Issue 7, July 2021 ||

| DOI: 10.15680/IJIRCCE.2021.0907162 |

to understand it better.

C.*Future Scope of the Project*

- Migrating to a Cloud Native, Scalable database such as Snowflake to cope-up with ever-increasing data into big data realms.
- Develop iOS and Android Clients for Mobile Access
- Further tune the model for better performance

REFERENCES

1. Y. Cherdo, P. d. Kerret and R. Pawlak, "Training LSTM for Unsupervised Anomaly Detection Without A Priori Knowledge," in Proceedings of IEEE *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 4297-4301, doi: 10.1109/ICASSP40776.2020.9053744.
2. S. Parsai and S. Mahajan, "Anomaly Detection Using Long Short-Term Memory," in Proceedings of IEEE *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 333-337, doi: 10.1109/ICESC48915.2020.9155897.
3.O. I. Provotar, Y. M. Linder and M. M. Veres, "Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders," in Proceedings of IEEE *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, Kyiv, Ukraine, 2019, pp. 513-517, doi: 10.1109/ATIT49449.2019.9030505
4. H. Wang, M. J. Bah and M. Hammad, "Progress in Outlier Detection Techniques: A Survey," in IEEE Access, vol. 7, pp. 107964-108000, 2019, doi: 10.1109/ACCESS.2019.2932769.
5. E. H. Budiarto, A. Erna Permanasari and S. Fauziati, "Unsupervised Anomaly Detection Using K-Means, Local Outlier Factor and One Class SVM," in Proceedings of IEEE 2019 5th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, 2019, pp. 1-5, doi: 10.1109/ICST47872.2019.9166366.
6. J. Kao and J. Jiang, "Anomaly Detection for Univariate Time Series with Statistics and Deep Learning," in Proceedings of IEEE 2019 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2019, pp. 404-407, doi: 10.1109/ECICE47484.2019.8942727.
7. S. Liu, Z. Qin, X. Gan and Z. Wang, "SCOD: A Novel Semi-supervised Outlier Detection Framework," in Proceedings of IEEE 2019 IEEE/CIC International Conference on Communications in China (ICCC), Changchun, China, 2019, pp. 316-321, doi: 10.1109/ICCChina.2019.8855955.
8. J. Zhai, S. Zhang, J. Chen and Q. He, "Autoencoder and Its Various Variants," in Proceedings of IEEE *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Miyazaki, Japan, 2018, pp. 415-419, doi: 10.1109/SMC.2018.00080.
9.W. Lu *et al*., "Unsupervised Sequential Outlier Detection With Deep Architectures," in *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4321-4330, Sept. 2017, doi: 10.1109/TIP.2017.2713048.
10.H. C. Mandhare and S. R. Idate, "A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques," in Proceedings of IEEE *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 931-935, doi: 10.1109/ICCONS.2017.8250601.
11. A. Majumdar and A. Tripathi, "Asymmetric stacked autoencoder," in Proceedings of IEEE *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, 2017, pp. 911-918, doi: 10.1109/IJCNN.2017.7965949
12. N. Abroyan, "Convolutional and recurrent neural networks for real-time data classification," in Proceedings of IEEE *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*, Luton, 2017, pp. 42-45, doi: 10.1109/INTECH.2017.8102422.
13. N. Malini and M. Pushpa, "Analysis on credit card fraud identification techniques based on KNN and outlier detection," in Proceedings of IEEE 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 255-258, doi: 10.1109/AEEICB.2017.7972424.
14. X. Wang, Y. Chen and X. L. Wang, "A Centroid-Based Outlier Detection Method," in Proceedings of IEEE 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2017, pp. 1411-1416, doi: 10.1109/CSCI.2017.247.
15. G. Williams, R. Baxter, Hongxing He, S. Hawkins and Lifang Gu, "A comparative study of RNN for outlier detection in data mining," in Proceedings of IEEE *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, Maebashi City, Japan, 2002, pp. 709-712, doi: 10.1109/ICDM.2002.1184035.
16. Taylor, Sean & Letham, Benjamin. (2017). Forecasting at scale. 10.7287/peerj.preprints.3190v2.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 7.542

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details