Trends in Undergraduate Research (2021) 4(1): e8-17
https://doi.org/10.33736/tur.3201.2021

Visualisation System of COVID-19 Data in Malaysia

REHMAN ULLAH KHAN*, NOR SYAZA SYAMIMI, CLADIA SIMBUT ANAK MAMBANG, IVY ANAK THOMAS, & TZI NI WEE

Faculty of Cognitive Sciences and Human Development, Universiti Malaysia Sarawak, 94300 Kota Samarahan, Sarawak, Malaysia
*Corresponding author: rehmanphdar@gmail.com

ABSTRACT

Pandemics are highly unlikely events, therefore, we need a system to understand the statistics about the pandamic. Machine learning algorithms can analyse the data and then we can plan for handling the pandamic. To date, many people are suffering because of the lack of reliable information system. The problem is that there is no integrated system to use the data and plan for pandemic management to minimise social panic. This study aims to provide a system, using COVID-19 data as a sample to visualise and analyse cases, deaths, discharged ICU cases updates in Malaysia as a whole state wise of COVID-19 daily statistics. The results provide visualisation and case comparison among states in Malaysia to easily and quickly understand the situation. This will help and assist the management in decision-making.

Keywords: COVID-19, Decision support system, Disaster management, Panic, Pandemic

INTRODUCTION

The Coronavirus outbreak which began in Wuhan, China, in December, has expanded to every corner of the globe. Millions of people around the world have been sickened and died. With the pandemic affecting worldwide, it is crucial to look at the numbers to counter the outbreak. World Health Organization states that Coronavirus (COVID-19) is contagious caused by coronavirus (World Health Organization, 2020). This viral disease is caused by a recently discovered coronavirus mutation; SARS-CoV-2; a disease that causes respiratory infections in humans. This virus was brought to our attention through cases reported from Wuhan, China in December 2019. The earliest case of COVID-19 in Malaysia was reported on 24 January 2020.

COVID-19 virus is spread by droplets in a short distance and settles on surfaces when an infected person coughs or sneezes without covering their mouth and nose (Ministry of Health, 2021). There are effective ways of avoiding catching the virus, such as washing your hands regularly with soap and water, using alcohol-based hand sanitiser, avoid touching your face and eyes, maintaining a social distance of six feet apart, wearing two-ply face masks and staying up-to-date with COVID-19 virus news.

Alternative infection prevention and control measures can be applied in the healthcare environment such as wearing personal protective equipment to help prevent the spread of infectious diseases. People must meet proper standards of healthcare, the equipment must be worn correctly in the appropriate context (Ministry of Health, 2021). A study by MacIntyre and Hasanain (2020), wearing a mask may protect people from becoming infected and prevent transmission of infection from infected people because a surgical mask can filter three micrometres droplets. Hence, wearing a mask is important because new research on face coverings shows that the risk of infection to the wearer is decreased by 65 percent (Chu *et al*., 2020). Besides, during this COVID-19 crisis, it is necessary to curb this pandemic.

The need to build up a system to analyse and visualise the COVID-19 crisis as a step to counter the pandemic is crucial for the nation. The government can take further action and deliberate on timely planning to control the pandemic by implementing a plan for targeted high-risk states, high-risk individuals and various health

backgrounds. By using the programming tool, python, via an open-source web application Jupyter Notebook, we aim to visualise and analyse the cases, deaths, discharged cases, and current ICU status within a particular time in Malaysia or the specifications of each state with the COVID-19 outbreak daily statistics. Besides, the system also focuses on the total number cases in each state and visualises it at a specific time. Finally, the system also seeks to compare cases between states in Malaysia. With the following specific tasks, the system will target to complete each aspect to get the best results from the system.

The next section relates to studies that provide literature reviews in the field. Materials and methods explain the methodology of how the system was designed and developed. The last section is about the results and conclusion.

Related Studies

Time series prediction

Machine learning (ML) has recently emerged as one of the key computing technologies and is increasingly applied in day-to-day life and various industrial domains (Deparday, Gevaert, Molinario, Soden, & Balog-Way, 2019). ML is an artificial intelligence application that uses algorithms that work on characteristics of available data to make further predictions. In the era of emerging technologies such as unmanned aerial vehicles, internet of things, and satellite-based technology, the network is becoming more autonomous. Such systems require several local decisions to be made, such as bandwidth selection, data rate selection, power control, and user association to a base station. We can use ML algorithms to address these issues and lower human intervention in uncertain and stochastic environments.

Researchers are using different machine learning algorithms to detect or predict COVID-19 cases. One technique is time series, which uses information from past data, values, and patterns to predict future activity. Time series is a set or series of data points ordered in time whereby the independent variable would be the time, and the forecast for the future would be the goal for the time series. It is often modelled via stochastic process, Y(t), whereby in a forecasting setting, the method Y(t+h) would be used with what information is available during that particular time and setting (t) would be applied (Kerrigan *et al*., 2019). The difference between time series data and cross-sectional data is the fact that time-series data is collected from various points in time. In contrast, cross-sectional data would collect data at a single point in time. Some well-known examples of forecasting models used in time series would be ARIMA, TBATS, Prophet, LSTM, ANFIS, MNETAR, and GARCH. The application of time series can be found in various contexts such as daily weather temperature, allocation of resources, business planning, and stock price forecasting (Erica, 2021). Univariate is called when a time series data contains records of a single variable. When the dataset has more than one variable, it is multivariate data. Time series can be either continuous or discrete. The observations are calculated at every instance of time in an ongoing time series. However, a discrete-time series contains observations measured at distinct points of time (Adhikari & Agrawal, 2013).

LSTM network is a special kind of recurrent neural network (RNN) that could learn long-term dependencies proposed by Hochreiter and Schmidhuber (1997). LSTM is explicitly designed to avoid the long-term dependency problem. The architecture of this network is shown in Figure 1 (Franklin, 2018).

Bouhamed (2020) proposed deep learning nested sequence prediction models with LSTM to monitor the infection and recovery process of the Covid-19 cases continuously. This research used COVID data from 79 countries. This model is capable of controlling the Covid-19 pandemic by making the right decisions. Yudistira (2020) compared the LSTM model with the precedent model of RNN. This study involved 100 countries data cases from 22 Jan 2020 until 1 May 2020. LSTM was concluded as a promising tool to predict the COVID-19 pandemics by learning from big data and can potentially predict future outbreaks.

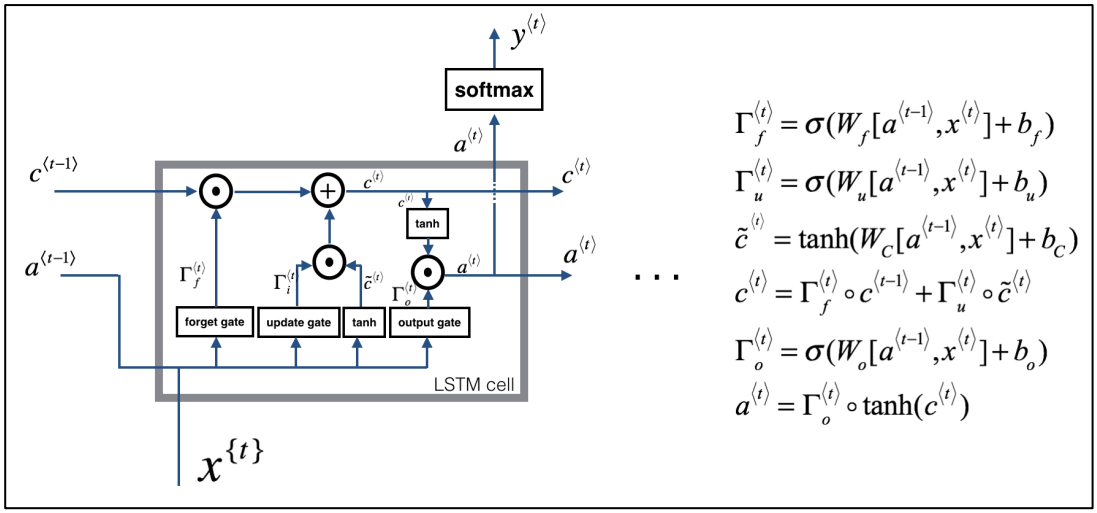

Figure 1. LSTM-cell. This track and updates a "cell state" or memory variable c(t) at every time-step, which can be different from a(t)

Linear Regression (LR) is a linear approach to modelling the relationship between a scalar response and one or more explanatory variables (also known as dependent and independent variables). An LR line has a condition of the structure:

$$y = \mathrm{Ax} + \mathrm{B} \qquad (1)$$

Where *y* is the independent number while x is the dependent variable. The slope of the line is A, and B is the intercept, which is the value of y when x equals zero (Yadav, 2020). LR is to predict response with a linear function of predictors as follows:

$$y = \Delta 0 + \Delta 1 \times 1 + \Delta 2 \times 2 + \cdots \Delta n \times n + \epsilon \quad (2)$$

Where x 1 , x 2 ,..., x n are predictors, and y is the result of the prediction. At the same time, Δ 0 , Δ 1 , Δ 2 ,…, Δ n are parameters and ϵ for errors (Uyanık & Güler, 2013).

Yadav (2020) studied the spreading rate and forecast the cases of the Covid-19 by developing the regression analysis models using COVID-19 Indian data from 1 March 2020 to 11 April 2020 was used. The models' performances were evaluated with sum of squared errors, degree of freedom for error, R^2 and adjusted R^2. This model performed well such as actual case were 8.1k on 1st day, 17.8k on 7th day and its predicted cases were 8.5k and 17.6k . Ayyoubzadeh, Ayyoubzadeh, Zahedi, Ahmadi, and Kalhori (2020) studied predicting COVID-19 incidence through analysis of Google trends data in Iran using data mining and deep learning. They used daily COVID-19 cases from 15 February 2020 to 18 March 2020.

Face mask detection

The studies by Jiang and Fan (2020), and Loey, Manogaran, Taha, and Khalifa (2021) combine machine-learning algorithms to detect face masks. The authors used one algorithm for feature extraction and the other for classification purpose. The authors also used more than one dataset. To increase the versatility and accuracy of the system, the authors combined two datasets to train their algorithms.

Cakiroglu, Ozer and Gunsel (2019) implemented an existing model called Mask RCNN. They trained the model by small datasets and combined segmentation with an object bounding box detection. Similarly, Li, Wang, Li and Fei (2020) used YOLOv3 for face detection and changed the detection layer to detect smaller faces. They also replaced the logistic classifier with Softmax to maximise the difference of inter-class features and decreasing the dimension of features on detection layers to improve the speed. The system was trained on the WIDER FACE database and the CelebA database and tested on the FDDB database. Nagrath *et al*. (2021) used

deep learning, TensorFlow, Keras, and OpenCV to detect faces. The detected faces were cropped and passed to the MobilenetV2 classifier to classify the faces into two classes, mask and no mask.

In the study by Loey *et al*. (2021), the datasets were classified into "with masks" and "without masks". The study by Jiang and Fan (2020) on the other hand, classified their datasets into "face with masks", "face without masks", "faces without and without a mask in the same image", and "confusing images without a mask". This classification was made possible due to the author's dataset containing not only images of faces and people wearing masks, but also images with faces masked by hand or other objects. Both authors successfully implemented their face mask detection system as most of their algorithms have a high detection accuracy.

MATERIALS & METHODS

The programming language that we used for developing this expert system was Python. To perform more productivity and easy collaboration, we used the Jupyter Notebook as the main platform for using the programming language.

While on the other side, the method that we used in this project was the Waterfall method. The Waterfall method widely used in software development and IT as a linear project management approach (Andrei, Casu-Pop, Gheorghe, & Boiangiu, 2019). It involves five phases which are requirements analysis, design, implementation, verification, and maintenance as shown in Figure 2 below. In the first phase of requirements, we gathered the requirements for the system which emphasise elements that we want to implement. This includes the decision of specific tasks in machine learning. In another way, the requirement phase is one of the components in the planning stage of this whole project. The next phase is the design phase of breaking down the logical design and physical design. The logical design consists of finding possible solutions with theories and brainstorming while the physical design consists of ideas that are constructed before being made into concrete specifications. In this phase, we also included searching for source codes and datasets to find examples of the actual codes for this machine learning in a few different websites like kaggle.com and github.com. Overall, the design stage involved us providing support, improvement and solidity to the intentions that we collect in the requirement phase.

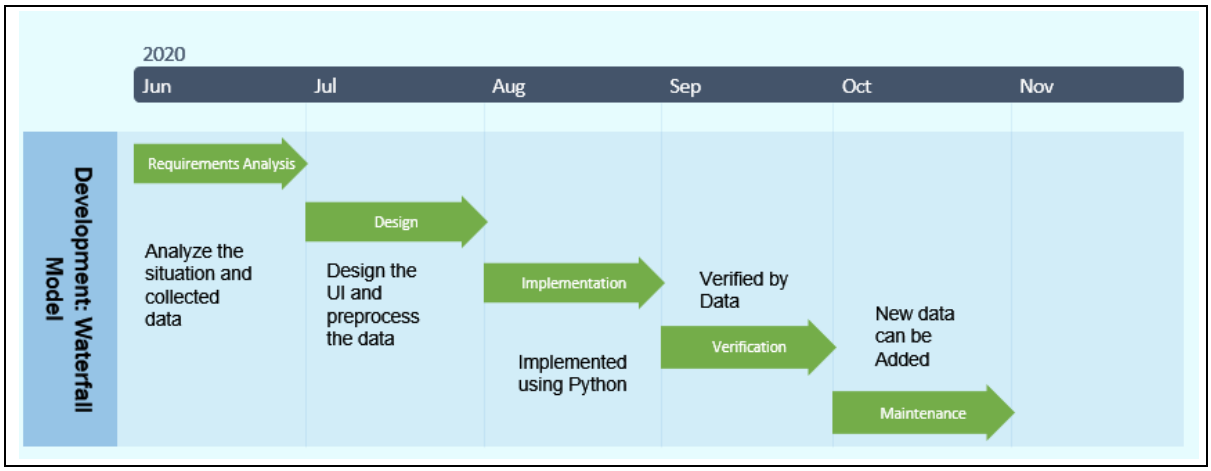

Figure 2. The Waterfall Model.

Next is the implementation phase. In this phase, the programmers had to specify and implement the requirements to produce an actual code for machine learning. The authors developed a system and implemented the data of every state and generally Malaysia as a whole. Eventually, the next stage is the verification phase. In this phase, we tested the whole machine learning and checking the specific task together to make sure everything

fulfils the requirements that were laid out in the first phase. As we finish the fourth phase of the Waterfall method, the project comes to an end. Finally, the last phase is the maintenance phase. In this phase, we have to improve machine learning with feedback from the fourth stage and create a new version based on feedback including bugs, misinformation, and errors during production.

Machine Learning Library

The library used in our project includes pandas, NumPy, time delta, urlopen, matplotlib, seaborn as well as plotly.express. Pandas datetime is needed to make it easier to plot the data. This project requires updated data from the source. Hence, there is a need to import urlopen to open the required online dataset by its URL. Instead of just using matplotlib as the basic visualisation in python, we decided to choose plotly.express to help our visualisation be more interactive.

Datasets

Two datasets were used in this project. As mentioned before, this project aims to show the current trend of COVID-19. Thus, an updated dataset is required. For both datasets, the URL is inserted into our machine learning. https://raw.githubusercontent.com/ynshung/covid-19-malaysia/master/covid-19-malaysia.csv is the dataset used for COVID-19 cases in Malaysia.

For the second dataset, the same method to insert the dataset was used. However, empty values and NaN can be found in the dataset that may affect the visualisation. To overcome this, we replace the '-' symbol with None, which can be found under the 'wp-putrajaya' column. Then, the function dropna() helps to drop any NaN and None values in the dataset.

Pre-processing of Data

Below is the code to show the latest update for total cases, death, discharge and ICU for a particular time. To make the data analysis easier, the data was reshaped using temp.melt() function. This function is useful in formatting the column to identifier variables (id_vars= 'date'), and measured variables (value_vars=['cases', 'discharged', 'death']). Note that, in the temp() function, the tail() value of the dataset was taken, which means the final current data is taken, hence fulfilling the goal to visualise the current trend of Covid-19 in Malaysia. In this visualisation, px.treemap was used to show hierarchical data using nested rectangles. Each of the rectangles was defined by labels that can be clicked to zoom in or out, and when hovered, it will show the path bar on the left corner of the treemap.

RESULTS

Latest Update for Covid-19 Active, Death, Discharged and ICU Cases

Using the datasets and by applying the algorithms, it shows that the current total cases were 8734, 8526 for total discharge and 122 for total death as shown in Figure 3 below. The total death can be barely seen in the screenshot, due to the small number compared to the total cases and discharged. As can be seen from the output below, the data for the updated case of death is shown when hovered. No label showed for ICU current data as no cases for the current time.

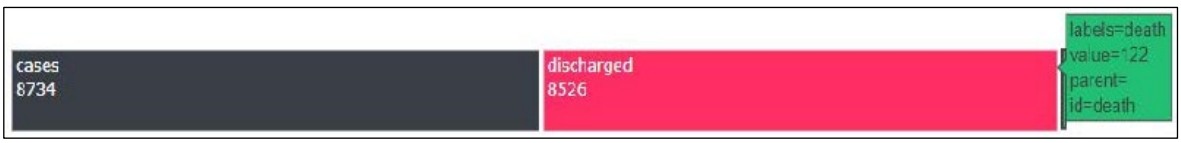

Figure 3. Covid-19 total cases, discharge and death.

Visualising Covid-19 Active, Death, Discharged and ICU Cases

Before visualising each data, plot_df_Msia was defined first to make it easier to call the function later. Inside this function, plotly.express was used to make an interactive bar chart with a colour sequence that shows the increasing amount of data from the beginning until the current time. To visualise the data, function plot_df_Msia was called

with the variable name inserted inside the function, followed by the colour palette that was defined earlier. As a result, the total number of cases is shown in Figure 4, deaths shown by Figure 5, discharged shown by Figure 6, and the total number of patients in ICU shown in Figure 7.

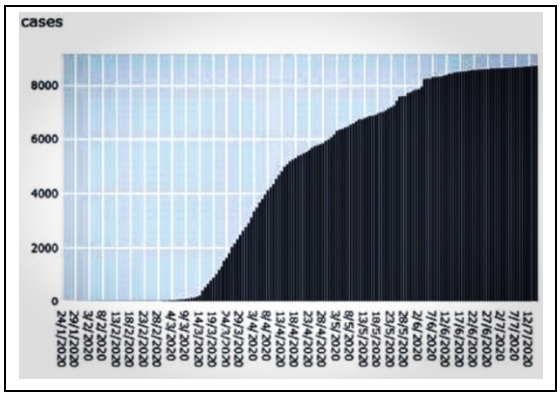

Figure 4. Malaysia's Covid-19 cases.

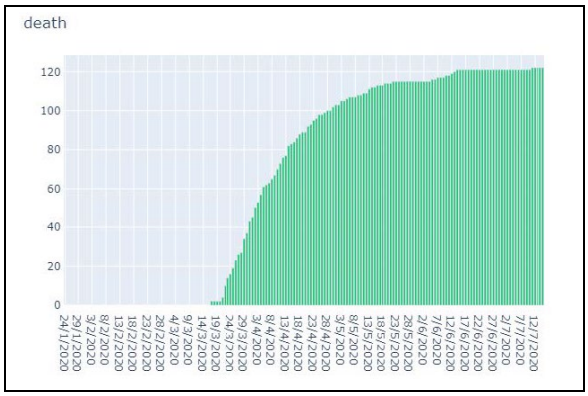

Figure 5. Malaysia's Covid-19death cases.

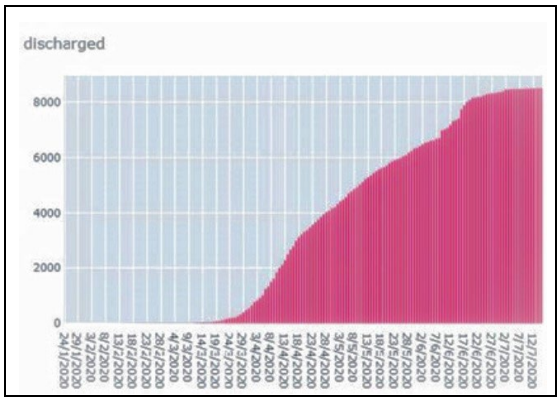

Figure 6. Malaysia's Covid-19 discharged cases.

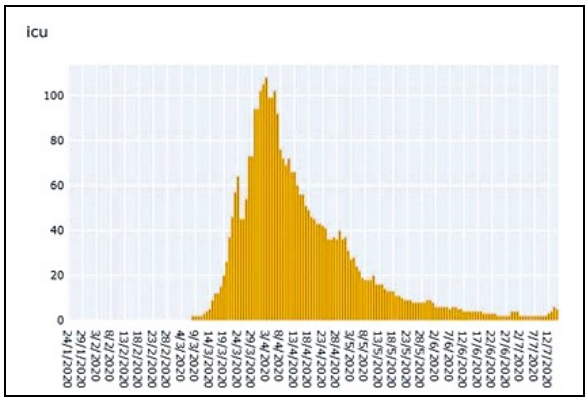

Figure 7. Malaysia's ICU Covid-19 cases

Visualising Trend in Covid-19 Discharged and Death Cases

Trend is a future direction and is very important for best planning. temp.melt() is used again to make the analysis easier. This time, the temp() function takes all data inside the dataset, and not only the tail as mentioned before. The measured variables (value_vars) only include cases, discharge and death. The analysis is then visualised using px.line, as shown in Figure 8.

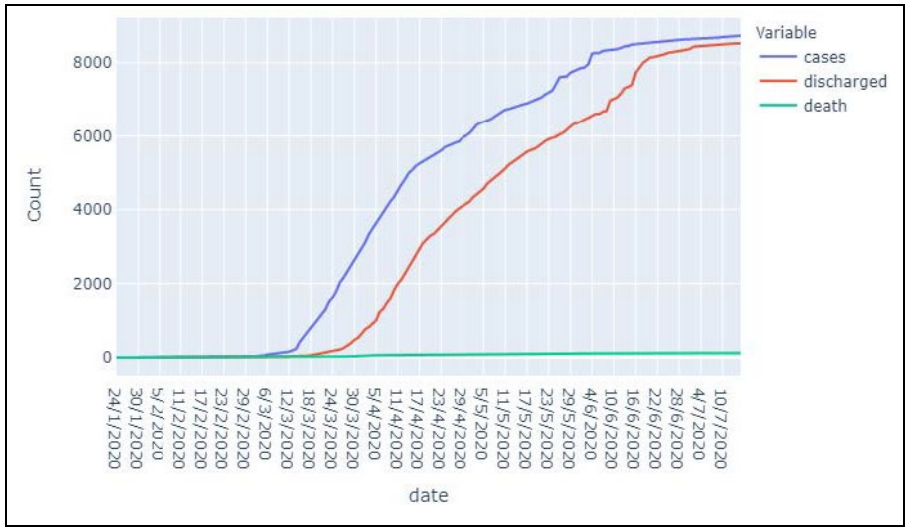

Figure 8. Comparison of Covid-19 death, and discharge cases.

Latest Update of Covid-19 Cases Based on State

The latest updated system can help in proper planning and management. Below is the code to show the latest update for each state in Malaysia for a particular time. The data was reshaped by using temp.melt() function to make the analysis data easier. This function is useful in formatting the column to identifier variables (id_vars= 'date'), and measured variables (value_ vars = ['perlis', 'kedah', 'pulau-pinang', 'perak', 'selangor', 'negeri sembilan', 'melaka', 'johor', "pahang', 'terengganu', 'kelantan', 'sabah', 'sarawak', 'wp-kuala-lumpur','wp-putrajaya', 'wp-labuan']). In the temp() function, the tail() value of the dataset was taken, which indicates that the final current data is taken from the dataset, to visualise the current trend of Covid-19 in Malaysia. In visualisation, px.treemap was used to show hierarchical data by using nested rectangles. Each rectangle is defined by labels that can be zoomed in or out, and when hovered, it will show the path bar on the left corner of the treemap. Based on Figure 9, it shows the total case for each state. The total case in Kuala Lumpur is 2447, 2094 in Selangor, 1027 in Negeri Sembilan, 701 in Johor, 503 in Sarawak, 380 in Sabah, 365 in Pahang, 258 in Melaka and Perak, 157 in Kelantan, 111 in Terengganu, 121 in Pulau Pinang, 99 in Kedah, 98 in Putrajaya, 18 in Perlis and 17 in Labuan. The highest total case is in Kuala Lumpur with 2447, the lowest total case is Labuan with 17 (Figure 9).

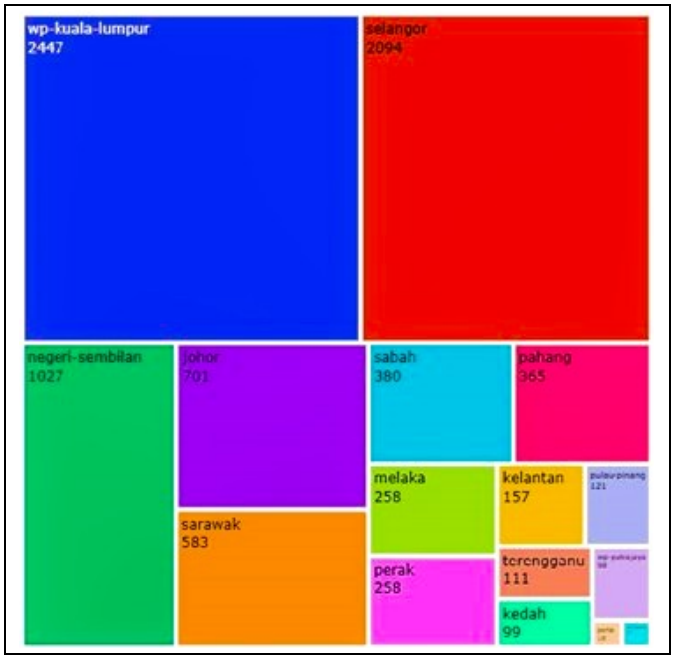

Figure 9. Comparison of Covid-19 cases among States.

Detailed Visualisation of Covid-19 Cases Based on State

We have calculated detailed visualisation of each state, but only one example of Perlis is shown here in Figure 10. The same method was applied to all other 12 states and 3 federal territories by changing the variable name for each state, such as plot_stacked(‘kedah’), plot_stacked(‘pulau-pinang’), plot_stacked(‘perak’), plot_stacked (‘selangor’), plot_stacked(‘negeri sembilan’), plot_stacked(‘melaka’), plot_stacked(‘johor’), plot_stacked (‘pahang'), plot_stacked('terengganu'), plot_stacked('kelantan'), plot_stacked('sabah'), plot_stacked ('sarawak'), plot_stacked(‘wp-kuala-lumpur'), plot_stacked('wp-putrajaya'), plot_stacked('wp-labuan'). Before visualising the

data, plot_stacked() function and plot_line() function was defined to make it easier to call the function later. In plot_stacked() function, px.bar() was used to represent each data in rectangular mark. In the plot_line() function, px.line() was used to represent the data in vertex of a polyline mark in 2D space.

Comparison of Covid-19 Cases Among States

Comparative analysis among states is very useful to point out high-risk states and manage the situation. We used plt.figure to compare cases between states in Malaysia. The purpose of using plt.figure is to create figure objects. The whole figure is regarded as the figure object. It is important to use plt.figure( ) specifically when we want to change the size of the figure and when we want to add several objects to the Axes in a single figure. For the figure size object plt.figure(15,7), it only has six figures possible because it cannot be done until fig.add_subplot(237). We can see in Figure 11, an upward trend, then changing the trend horizontally showing that the COVID-19 cases between states are being controlled.

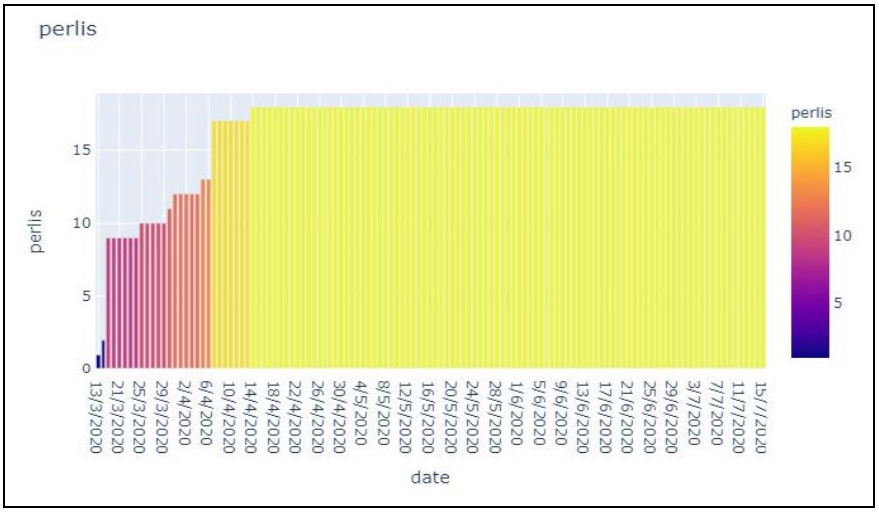

Figure 10. Covid-19 cases in Perlis

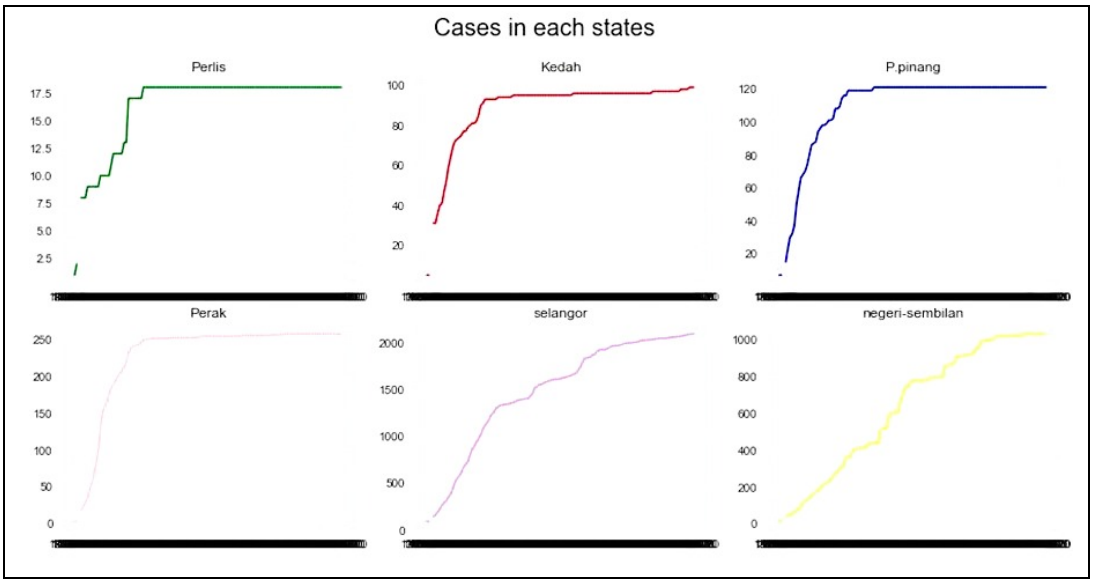

Figure 11. Covid-19 cases based on States

CONCLUSION

This study provided a system, using COVID-19 data as a sample, visualise, and analyse the cases, deaths, discharged ICU cases updates at a particular time in Malaysia as a whole or with the specification of each state with the COVID-19 outbreak daily statistics. The results provided visualisation and comparison of cases among states in Malaysia. This will assist the management in decision making. Therefore, the pandemic can be managed by proper planning and utilising advanced technologies such as machine learning.

Limitation and Recommendation

The main limitation of this system is to get real-time data from all the important sources. The real real-time data will predict the real situation. Therefore, it is recommended to integrate such a system with data sources to get real-time data.

REFERENCES

Adhikari, R., & Agrawal, R. K. (2013). An introductory study on time series modeling and forecasting. *arXiv preprint arXiv*:1302.6613.

Andrei, B. A., Casu-Pop, A. C., Gheorghe, S. C., & Boiangiu, C. A. (2019). A study on using waterfall and agile methods in software project management. *Journal of Information Systems & Operations Management*, *13*(1), 125-135.

Ayyoubzadeh, S. M., Ayyoubzadeh, S. M., Zahedi, H., Ahmadi, M., & Kalhori, S. R. N. (2020). Predicting COVID-19 incidence through analysis of google trends data in Iran: Data mining and deep learning pilot study. *JMIR Public Health and Surveillance*, *6*(2), e18828.

Bouhamed, H. (2020). Covid-19 cases and recovery previsions with deep learning nested sequence prediction models with long short-term memory (LSTM) architecture. *Int. J. Sci. Res. in Computer Science and Engineering*, *8*(2).

Cakiroglu, O., Ozer, C., & Gunsel, B. (2019, April). Design of a deep face detector by mask R-CNN. In *2019 27th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.

Chu, D. K., Akl, E. A., Duda, S., Solo, K., Yaacoub, S., Schünemann, H. J., El-harakeh, A., Bognanni, A., Lotfi, T., & Reinap, M. (2020). Physical distancing, face masks, and eye protection to prevent person-to-person transmission of SARS-CoV-2 and COVID-19: A systematic review and meta-analysis. *The Lancet*, *395* (10242), 1973-1987.

Deparday,V., Gevaert,C. M., Molinario,G. M., Soden, R. J. & Balog-Way, S. A. B.(2019). *Machine Learning for Disaster Risk Management.* Washington, D.C. : World Bank Group. Retrieved June 16, 2021, from http://documents.worldbank.org/curated/en/503591547666118137/Machine-Learning-for-Disaster-Risk-Management.

Erica (2021). Introduction to the fundamentals of time series data and analysis. Retrieved June 10, 2021, from https://www.aptech.com/blog/introduction-to-the-fundamentals-of-time-series-data-and-analysis/

Franklin, M. (2018). Introduction to Sequence Models - RNN and LSTM. Retrieved June 16, 2021, from https://franklinwu19.github.io/2018/08/27/rnn-lstm/.

Hochreiter, S., & Schmidhuber, J. (1996). LSTM can solve hard long time lag problems. In *Proceedings of the 9th International Conference on Neural Information Processing Systems* (pp. 473-479).

Jiang, M., Fan, X., & Yan, H. (2020). Retina mask: A face mask detector. *arXiv preprint arXiv*:2005.03950.

Kerrigan, J., Plante, P. L., Kohn, S., Pober, J. C., Aguirre, J., Abdurashidova, Z., ... & Zheng, H. (2019). Optimizing sparse RFI prediction using deep learning. *Monthly Notices of the Royal Astronomical Society*, *488* (2), 2605-2615.

Li, C., Wang, R., Li, J., & Fei, L. (2020). Face detection based on YOLOv3. *4th International Conference on Intelligent Computing, Communication and Devices, ICCD 2018; Guangzhou; China; 7 December 2018 through 9 December 2018.* Volume 1031 AISC, 2020, Pages 277-284.

Loey, M., Manogaran, G., Taha, M. H. N., & Khalifa, N. E. M. (2020). A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. *Measurement: Journal of the International Measurement Confederation*, *167*, 108288-108288.

MacIntyre, C. R., & Hasanain, S. J. (2020). Community universal face mask use during the COVID 19 pandemic - from households to travellers and public spaces. *Journal of Travel Medicine*, *27*(3), 056.

Ministry of Health (2021). COVID-19: Use of masks and face coverings in the community. Retrieved June 16, 2021 from *https://www.health.govt.nz/our-work/diseases-and-conditions/covid-19-novel-coronavirus/covid-19health-advice-public/covid-19-use-masks-and-face-coverings-community.*

Nagrath, P., Jain, R., Madan, A., Arora, R., Kataria, P., & Hemanth, J. (2021). SSDMNV2: A real time DNN-based face mask detection system using single shot multibox detector and MobileNetV2. *Sustainable Cities and Society*, *66*, 102692.

Uyanık, G. K., & Güler, N. (2013). A study on multiple linear regression analysis. *Procedia-Social and Behavioral Sciences, 106*, 234-240.

World Health Organization. (2020). Rational use of personal protective equipment for Coronavirus disease (COVID-19)*:* Interim guidance, 27 February 2020 (No.WHO/2019-nCov/IPCPPE_use/2020.1). World Health Organization. Retrieved June 14, 2021, from https://www.who.int/publications/i/item/rational-use-of-personal-protective-equipment-for-coronavirus-disease-(covid-19)-and-considerations-during-severe-shortages.

Yadav, R. S. (2020). Data analysis of COVID-2019 epidemic using machine learning methods: a case study of India. *International Journal of Information Technology, 12***,** 1321–1330.

Yudistira, N. (2020). COVID-19 growth prediction using multivariate long short term memory. *arXiv preprint arXiv*: 2005.04809.