International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 267-271
Published Online August 2020 in IJEAST (http://www.ijeast.com)

COVID-19 DATA ANALYSIS AND DATA VISUALIZATION

Harinishri.S
Electronics and Communication
Rajalakshmi Engineering College
Chennai, Tamil Nadu, India

***Abstract*— COVID-19 (coronavirus disease 2019) is a viral infectious disease and presently, World Health Organization (WHO) has declared it as a global pandemic. By viewing the reports of August 2020, nearly 18 million people have been affected on the whole with over 700,000 deaths. As the cases are getting increased day by day, there is a huge demand for data manipulations and data storage. Thereby, data analytics play a vital role in identifying, arranging the data and visualizing in various forms. This can largely help in controlling the diseases in the regions and isolating those regions. Thus, data analytics is a new dawn in the medical field.**

Keywords: Data Analysis, plots, Graphs, Covid-19, Global pandemic, Deaths.

I. INTRODUCTION

Data mining has already established as a novel field for exploring knowledge from hidden lattices in the big datasets. Data Mining can be defined as a non-trivial extraction of implicit, formerly not known and significantly valuable from given data. Breifly, it is a process to analyze the data from different perspective and gather the knowledge from it. This meticulously gathered knowledge can be used in different domains, majorly in healthcare industry. So we will use data mining techniques along with big data and IOT to assist doctors and other people in making decision of virus control in early stages**.**

Big data is quite an old method which is becoming the most primitive method to harness data as many health care organizations are keen in discovering new opportunities for better understanding and predict customer behaviors. It is the kind of data that outflows the processing ability of conventional or the traditional Database Management Systems(DBMS). A patient who gets admitted in a hospital, his/her activities are strictly monitored and is taken into account. This complete set of data is stored under the patients name and is referred when there is a need for a check-up. It is now very crucial to investigate patient care and reducing both mortality and morbidity concerned with covid attacks that can be used to improve or to set up an alert program.

II. METHODS

A) DATA ANALYSIS

The data of Covid-19 affected patients is taken from one of the most reliable sources. The individual data is taken from https://www.covid19india.org/ and the state level data is taken from https://www.mohfw.gov.in/. This data taken from the Indian states and used for the complete analysis of Covid-19 Pandemic. The cumulative analysis is done using Jupyter Notebook. The Covid-19 data taken from 01/03/2020 to 06/08/2020. In dataset1, we have the state-wise Covid attacked patients details and in dataset2, we have the statewise-population and the area of that state.

```
data.head()
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed
0	1	1/3/2020	6:00 PM	Kerala	3	0	0	0	3
1	2	2/3/2020	6:00 PM	Telengana	1	0	0	0	1
2	3	2/3/2020	6:00 PM	Kerala	3	0	0	0	3
3	4	2/3/2020	6:00 PM	Delhi	1	0	0	0	1
4	5	3/3/2020	6:00 PM	Telengana	1	0	0	0	1

Fig 1:First 5 data of dataset1

```
datas.head()
```

	State	Aadhaar assigned as of 2019	Area (per sq km)
0	Delhi	21763471	1483
1	Haryana	28941133	44212
2	Kerala	36475649	38852
3	Himachal Pradesh	7560770	55673
4	Punjab	30355185	50362

Fig 2:First 5 data of Dataset2

III. ACCESSING OF DATA

Data mining is known as the process of extracting valuable data from an enormous amount of any raw data. It also does analysis of data patterns in numerous number batches of data

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 267-271
Published Online August 2020 in IJEAST (http://www.ijeast.com)

with the help of one or more software. Data mining has a wide range of applications in multiple fields, like technology and science. Data mining deals with efficient data collection, storing it as well as processing it. Data mining can also be called as Knowledge Discovery in Data (KDD). In order to view the widespread of virus in a India for a particular day, the below code is used for analysis.

```
data_latest = data[data['Date']=="5/8/2020"]
data_latest
```

This gives the data of people who got affected in all the indian states , people who got cured and also the number of deaths which took place.

4815	4816	5/8/2020	8:00 AM	Andaman and Nicobar Islands	-	-	277	12	928
4816	4817	5/8/2020	8:00 AM	Andhra Pradesh	-	-	95625	1604	176333
4817	4818	5/8/2020	8:00 AM	Arunachal Pradesh	-	-	1105	3	1790
4818	4819	5/8/2020	8:00 AM	Assam	-	-	34421	115	48161
4819	4820	5/8/2020	8:00 AM	Bihar	-	-	40348	347	61788
4820	4821	5/8/2020	8:00 AM	Chandigarh	-	-	715	20	1206
4821	4822	5/8/2020	8:00 AM	Chhattisgarh	-	-	7613	69	10202
4822	4823	5/8/2020	8:00 AM	Dadra and Nagar Haveli and Daman and Diu	-	-	919	2	1325
4823	4824	5/8/2020	8:00 AM	Delhi	-	-	125226	4033	139156
4824	4825	5/8/2020	8:00 AM	Goa	-	-	5114	60	7075
4825	4826	5/8/2020	8:00 AM	Gujarat	-	-	48376	2633	65599
4826	4827	5/8/2020	8:00 AM	Haryana	-	-	31226	448	37796
4827	4828	5/8/2020	8:00 AM	Himachal Pradesh	-	-	1710	14	2879
4828	4829	5/8/2020	8:00 AM	Jammu and Kashmir	-	-	14856	417	22396
4829	4830	5/8/2020	8:00 AM	Jharkhand	-	-	5164	128	13940
4830	4831	5/8/2020	8:00 AM	Karnataka	-	-	69272	2704	145830
4831	4832	5/8/2020	8:00 AM	Kerala	-	-	16299	87	27956
4832	4833	5/8/2020	8:00 AM	Ladakh	-	-	1127	7	1534
4833	4834	5/8/2020	8:00 AM	Madhya Pradesh	-	-	25414	912	35082
4834	4835	5/8/2020	8:00 AM	Maharashtra	-	-	299356	16142	457956
4835	4836	5/8/2020	8:00 AM	Manipur	-	-	1814	7	3018
4836	4837	5/8/2020	8:00 AM	Meghalaya	-	-	330	5	917
4837	4838	5/8/2020	8:00 AM	Mizoram	-	-	282	0	504
4838	4839	5/8/2020	8:00 AM	Nagaland	-	-	659	5	2405
4839	4840	5/8/2020	8:00 AM	Odisha	-	-	24483	216	37681
4840	4841	5/8/2020	8:00 AM	Puducherry	-	-	2537	58	4147
4841	4842	5/8/2020	8:00 AM	Punjab	-	-	12491	462	19015
4842	4843	5/8/2020	8:00 AM	Rajasthan	-	-	32832	732	46679
4843	4844	5/8/2020	8:00 AM	Sikkim	-	-	299	1	783
4844	4845	5/8/2020	8:00 AM	Tamil Nadu	-	-	206784	4349	266285
4845	4846	5/8/2020	8:00 AM	Telengana	-	-	50814	576	70958
4846	4847	5/8/2020	8:00 AM	Tripura	-	-	3725	30	5628
4847	4848	5/8/2020	8:00 AM	Uttarakhand	-	-	4847	95	8006
4848	4849	5/8/2020	8:00 AM	Uttar Pradesh	-	-	57271	1817	100310
4849	4850	5/8/2020	8:00 AM	West Bengal	-	-	56884	1785	80984

FIG 3: DETAILED REVIEW OF COVID-19 CASES IN INDIA

To get a clear view of the collected data, we can using various types of graphs in Python Jupyter notebook. Some of the graphs available are

- Scatter Plot
- Line Graph
- Pie Chart
- Area Chart
- Legend Graph
- Relplot
- Bullet Chart
- Bubble Chart
- Heat Map
- Funnel Chart
- Waterfall Graph
- Stacked Bar Graph

A) GRAPHS IN PYTHON

- **SCATTER PLOT**

To visualize the data, several graphs are used. The most reliable and efficient graph is scatter plot. A scatterplot is one kind of data representation that gives us the relationship between two independent/dependent variables. Each and every value present in the dataset gets plotted, like a dot whose (x,y) coordinates corresponds to the values for the two variables.
If the y-variable increase when the x variable increases, then it is called as a positive correlation between the two variables else the vice versa. A scatter plot must be used either if one continuous variable is under influence of the examiner and the other is dependent on it or when both these continuous variables are independent. The control parameter or the independent variable is plotted in the x-axis if there is a linear increasing/decreasing relationship exist.

```
sp = data_latest[(data_latest['Confirmed']>1000) | (data_latest['Cases/10million']>200)]
plt.figure(figsize=(12,8), dpi=80)
plt.scatter(data_latest['Confirmed'],data_latest['Cases/10million'], alpha=0.5)
plt.xlabel('Number of confirmed Cases', size=12)
plt.ylabel('Number of cases per 10 million people', size=12)
plt.scatter(sp['Confirmed'],sp['Cases/10million'], color="red")
for i in range(sp.shape[0]):
    plt.annotate(sp['State/UnionTerritory'].tolist()[i], xy=(sp['Confirmed'].tolist()[i],sp['Cases/10million'].tolist()[i]),
                 xytext = (sp['Confirmed'].tolist()[i]+1.0, sp['Cases/10million'].tolist()[i]+12.0), size=11)
plt.tight_layout()
plt.title('Visualization to display the variation in COVID 19 figures in different Indian states', size=16)
plt.show()
```

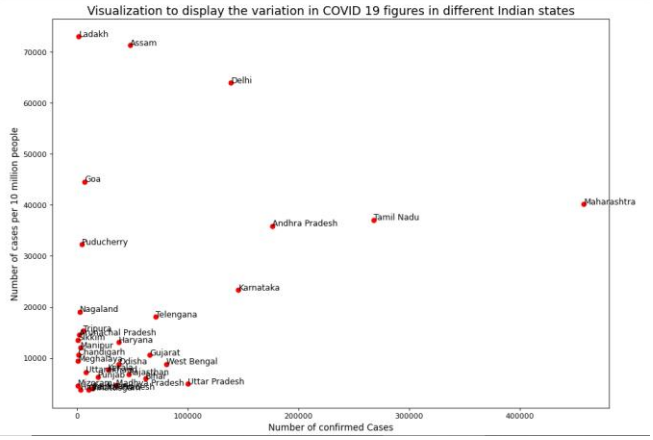

Fig 4: scatter plot of number of covid-19 cases per 10 million people

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 267-271
Published Online August 2020 in IJEAST (http://www.ijeast.com)

From the graph, we can visualize that when we take the state population into account, Maharashtra, Delhi, Tamil Nadu and Punjab have the highest number of covid-19 infected people. Followed by these states, other states like Nagaland, Goa, Chandigarh and also union territory Puducherry have considerable less covid-19 affected cases. From the statistics of Ladakh, 14 out 20 confirmed cases have been recovered.

- **BAR GRAPH**

A bar graph is a graph that represents data in a categorical fashion with rectangular bars having lengths tantamount to the values that contain. These bars can be plotted either in vertical or in horizontal direction. Bar graphs are mainly used in comparison of values between different categories or to make a note of series of changes happening over time. Bar graphs are exceptional when measuring changes over long intervals of time.

Matplotlib API in Python has the bar() function that must be used while plotting bar plots. The State with maximum number of cases has been plotted with bar graph.

```
data_latest = data_latest.sort_values(by=['Confirmed'], ascending = False)
plt.figure(figsize=(12,8), dpi=80)
plt.bar(data_latest['State/UnionTerritory'][:5],data_latest['Confirmed'][:5],
        align='center',color='lightblue')
plt.xlabel('Indian States',size=12)
plt.ylabel('Number of Confirmed Cases',size =12)
plt.title('States with maximum confirmed cases', size =12)
plt.show()
```

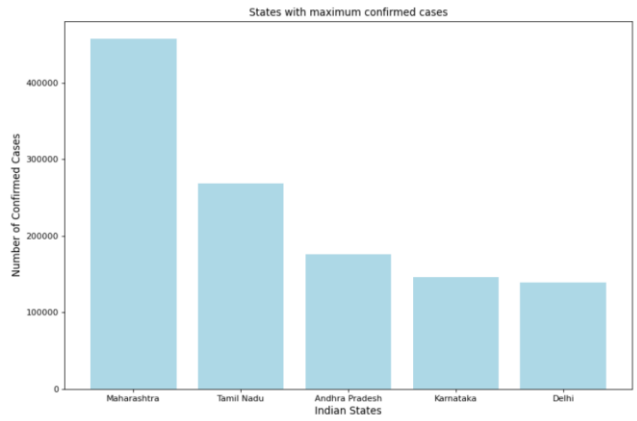

Fig 5: bar graph of number of confirmed cases due to covid-19 statewise

From the above bar graph visualization, we can infer that Maharashtra has the maximum number of confirmed COVID-19 cases as of now. It's peak value is nearly 7000 cases per day. It is an unusual case that no other state in India has even marked upto half of its total cases

Alternatively, the states with maximum number of deaths has been described in the below mentioned bar graph.

```
data_latest =data_latest.sort_values(by=['Deaths'], ascending = False)
plt.figure(figsize=(12,8), dpi=80)
plt.bar(data_latest['State/UnionTerritory'][:10],data_latest['Deaths'][:10], align='center',color='orange')
plt.ylabel('Number of Deaths', size = 12)
plt.xlabel('Indian States',size=12)
plt.title('States with maximum deaths', size = 16)
plt.show()
```

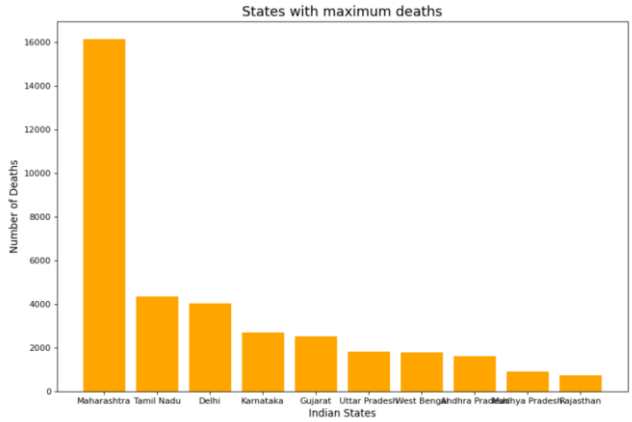

Fig 6: Bar Graph Of Number Of Covid-19 Deaths Statewise

- **RELPLOT**

Relplot is similar to scatterplot, where relplot lets us to create subplots in a single feature. Analysizing, deaths occurred in India using relplot. This relplot is available under the Seaborn library.

```
sns.relplot(x="Date",y="Deaths",data=data)
```

```
<seaborn.axisgrid.FacetGrid at 0xc6a0220>
```

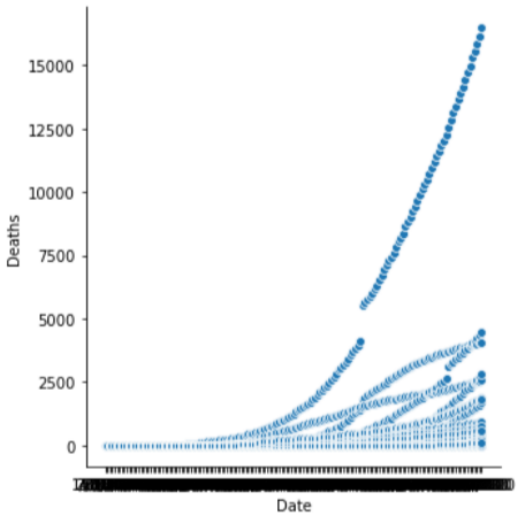

Fig 7: RELPLOT OF DEATHS IN INDIA

- **LEGEND**

A legend is a line denoting the values of the variables in the graph. **Legend()** function is available in the matplotlib library,

International Journal of Engineering Applied Sciences and Technology, 2020
Vol. 5, Issue 4, ISSN No. 2455-2143, Pages 267-271
Published Online August 2020 in IJEAST (http://www.ijeast.com)

By separating the 5 states individually, i.e Maharashtra, Kerala, Delhi, Rajasthan and Gujarat.

The data is firstly segregated and then applied to the legend().

```
plt.figure(figsize=(12,8), dpi=80)
plt.plot(covid19_kerala['Day Count'], covid19_kerala['Confirmed'])
plt.plot(covid19_maharashtra['Day Count'], covid19_maharashtra['Confirmed'])
plt.plot(covid19_delhi['Day Count'], covid19_delhi['Confirmed'])
plt.plot(covid19_rajasthan['Day Count'], covid19_rajasthan['Confirmed'])
plt.plot(covid19_gujarat['Day Count'], covid19_gujarat['Confirmed'])
plt.legend(['Kerala', 'Maharashtra', 'Delhi', 'Rajasthan', 'Gujarat'], loc='upper left')
plt.xlabel('Day Count', size=12)
plt.ylabel('Confirmed Cases Count', size=12)
plt.title('Which states are flattening the curve ?', size = 16)
plt.show()
```

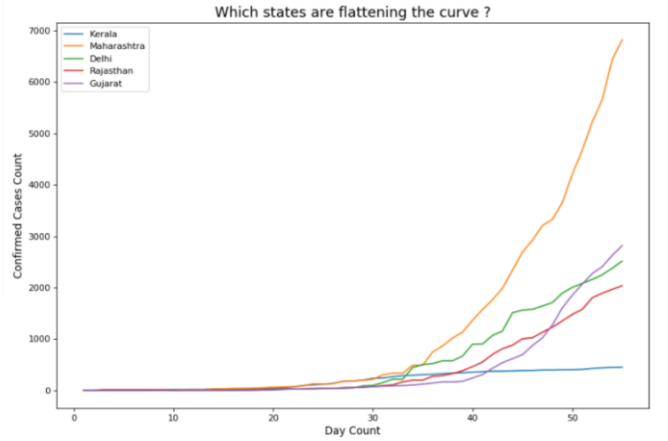

Fig 8: LEGEND OF 5 SELECTED STATES IN INDIA

From the above graph, it is visible that these curves gradually keeps increasing i.e as the severity of the virus increasing day by day, the number of confirmed cases also increases in the same fashion. But as an exceptional case, the southern state, Kerala has contained the virus spread and the amount of cases has become tremendously low and had set them up as an example to other states in India

Whereas the situation in Maharashtra has gone out of hands. The curve keeps increasing drastically and there seems to be no hope or possibility to slow down. Thus, it would be helpful for Maharashtra to consider Kerala as a forerunner in containing the covid-19 outbreak and follow their steps taken for protection. Gujarat's curve was under control for the first 40 days and after that, unfortunately there was a steep increase in the number of confirmed cases.

B) ANALYZING USING THE POPULATION IN INDIA

The data_latest mentioned below has the population count of the number of people in an individual state. The preliminary details were taken from the AADHAR (Identity proof in India) card from the official website controlled by the government of India. By using these figures, it is very efficient to calculate the estimate of the cases accordingly. By using this, Cases per 10million is found out.

```
data_latest = pd.merge(data_latest,datas_new, on="State/UnionTerritory")
data_latest['Cases/10million'] = (data_latest['Confirmed']/data_latest['Population'])*10000000
data_latest.head()
```

Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Population_x	Cases/10million
4835	5/8/2020	8:00 AM	Maharashtra	-	-	299356	16142	457956	114063427	40149.240825
4845	5/8/2020	8:00 AM	Tamil Nadu	-	-	208784	4349	268285	72344821	37084.202613
4824	5/8/2020	8:00 AM	Delhi	-	-	125226	4033	139156	21763471	63940.168367
4831	5/8/2020	8:00 AM	Karnataka	-	-	69272	2704	145830	62462743	23346.717258
4826	5/8/2020	8:00 AM	Gujarat	-	-	48376	2533	65599	62097024	10563.952308

Fig 9: FIRST 5 COVID-19 CONFIRMED CASES/10 MILLION IN INDIA

Decreasing order of cases/10million is found out using the below mentioned code.

```
data_latest.fillna(0, inplace=True)
data_latest.sort_values(by='Cases/10million', ascending=False)
```

	Sno	Date	Time	State/UnionTerritory	ConfirmedIndianNational	ConfirmedForeignNational	Cured	Deaths	Confirmed	Population_x	Cases/10mill
26	4833	5/8/2020	8:00 AM	Ladakh	-	-	1127	7	1534	210156	72993.395
16	4819	5/8/2020	8:00 AM	Assam	-	-	34421	115	48161	8755448	71292.088
2	4824	5/8/2020	8:00 AM	Delhi	-	-	125226	4033	139156	21763471	63940.168
20	4825	5/8/2020	8:00 AM	Goa	-	-	5114	60	7075	1587012	44580.833
0	4835	5/8/2020	8:00 AM	Maharashtra	-	-	299356	16142	457956	114063427	40149.240
1	4845	5/8/2020	8:00 AM	Tamil Nadu	-	-	208784	4349	268285	72344821	37084.202
7	4817	5/8/2020	8:00 AM	Andhra Pradesh	-	-	95625	1604	176333	49145466	35879.817
21	4841	5/8/2020	8:00 AM	Puducherry	-	-	2537	58	4147	1286189	32242.539
3	4831	5/8/2020	8:00 AM	Karnataka	-	-	69272	2704	145830	62462743	23346.717
27	4839	5/8/2020	8:00 AM	Nagaland	-	-	659	5	2405	1262729	19046.050
10	4846	5/8/2020	8:00 AM	Telengana	-	-	50814	576	70958	39184011	18108.916
22	4847	5/8/2020	8:00 AM	Tripura	-	-	3725	30	5628	3672893	15323.905
29	4818	5/8/2020	8:00 AM	Arunachal Pradesh	-	-	1105	3	1790	1229964	14553.271
30	4844	5/8/2020	8:00 AM	Sikkim	-	-	299	1	783	578914	13525.325
12	4827	5/8/2020	8:00 AM	Haryana	-	-	31226	448	37796	28941133	13059.613
25	4836	5/8/2020	8:00 AM	Manipur	-	-	1814	7	3018	2515724	11996.546
23	4821	5/8/2020	8:00 AM	Chandigarh	-	-	715	20	1206	1131522	10658.210
4	4826	5/8/2020	8:00 AM	Gujarat	-	-	48376	2533	65599	62097024	10563.952
28	4837	5/8/2020	8:00 AM	Meghalaya	-	-	330	5	917	978281	9373.584
6	4850	5/8/2020	8:00 AM	West Bengal	-	-	56884	1785	80994	91928327	8809.471
14	4840	5/8/2020	8:00 AM	Odisha	-	-	24483	216	37681	42826828	8798.703
18	4832	5/8/2020	8:00 AM	Kerala	-	-	16299	87	27956	38475649	7264.291
17	4848	5/8/2020	8:00 AM	Uttarakhand	-	-	4847	95	8008	11082791	7225.616
9	4843	5/8/2020	8:00 AM	Rajasthan	-	-	32832	732	46679	68936999	6771.255
11	4842	5/8/2020	8:00 AM	Punjab	-	-	12491	462	19015	30355185	6264.168
13	4820	5/8/2020	8:00 AM	Bihar	-	-	40348	347	61788	102714867	6015.499
5	4849	5/8/2020	8:00 AM	Uttar Pradesh	-	-	57271	1817	100310	203757489	4923.009
8	4834	5/8/2020	8:00 AM	Madhya Pradesh	-	-	25414	912	35082	74770270	4691.971
31	4838	5/8/2020	8:00 AM	Mizoram	-	-	282	0	504	1088577	4625.648

Fig 10: FULL DESCRIPTION OF COVID-19 CONFIRMED CASES/10 MILLION WITH THEIR POPULATION IN INDIA

IV. CONCLUSION

The coronavirus seems to increase in a unpredictable manner and is highly uncontrollable. The ILO's four pillar policy framework has passed the circular in regard to the virus outbreak, about the precautionary measures to be followed and possible ways to cure the affected patients. In order to manage and predict the data, we must arrange them appropriately, thus its visualization becomes efficient and error-free. As data analytics plays a vital role in controlling the spread of the virus by intimating the areas as red zones and green zones

respectively. Analyzing its death- rate and recovery rate. As a whole it is recommended to conduct these analysis to get a clear picture of day-to-day statistics. It is believed these encoded information from the raw sources play a remedial role amongst the Pandemic and helps us to lead a better life.

V. REFERENCE

[1] S. Shreshtha, A. Singh, S. Sahdev, M. Singha and S. Rajput, "A Deep Dissertion of Data Science: Related Issues and its Applications," *2019 Amity International Conference on Artificial Intelligence (AICAI)*, Dubai, United Arab Emirates, 2019, pp. 939-942, doi: 10.1109/AICAI.2019.8701415.

[2] N. S. Godbole and J. Lamb, "Using data science & big data analytics to make healthcare green," *2015 12th International Conference & Expo on Emerging Technologies for a Smarter World (CEWIT)*, Melville, NY, 2015, pp. 1-6, doi: 10.1109/CEWIT.2015.7338161.

[3] S. Liam, "Data, Data Science and the Research University," *2016 5th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI)*, Kumamoto, 2016, pp. 529-532, doi: 10.1109/IIAI-AAI.2016.261.

[4] T. Menzies, E. Kocaguneli, F. Peters, B. Turhan and L. L. Minku, "Data science for software engineering," *2013 35th International Conference on Software Engineering (ICSE)*, San Francisco, CA, 2013, pp. 1484-1486, doi: 10.1109/ICSE.2013.6606752.

[5] L. Erhan et al., "Analyzing Objective and Subjective Data in Social Sciences: Implications for Smart Cities," in IEEE Access, vol. 7, pp. 19890-19906, 2019, doi: 10.1109/ACCESS.2019.2897217.

[6] D. Lande, V. Andrushchenko and I. Balagura, "Data Science in Open-Access Research on-Line Resources," 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), Lviv, 2018, pp. 17-20, doi: 10.1109/DSMP.2018.8478565.

[7] N. W. Grady, "KDD meets Big Data," 2016 IEEE International Conference on Big Data (Big Data), Washington, DC, 2016, pp. 1603-1608, doi: 10.1109/BigData.2016.7840770.

[8] I. B. Hassan and J. Liu, "Embedding Data Science into Computer Science Education," 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 367-372, doi: 10.1109/EIT.2019.8833753.

[9] J. Pearl, "The new science of cause and effect, with reflections on data science and artificial intelligence," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 4-4, doi: 10.1109/BigData47090.2019.9005644.

[10] M. Muniswamaiah, T. Agerwala and C. C. Tappert, "Federated Query processing for Big Data in Data Science," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 6145-6147, doi: 10.1109/BigData47090.2019.9005530.

[11] J. Ming, L. Zhang, J. Sun and Y. Zhang, "Analysis models of technical and economic data of mining enterprises based on big data analysis," 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), Chengdu, 2018, pp. 224-227, doi: 10.1109/ICCCBDA.2018.8386516.

[12] L. Xianglan, "Digital construction of coal mine big data for different platforms based on life cycle," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 456-459, doi: 10.1109/ICBDA.2017.8078862.