Jadavpur University

Faculty of Engineering and Technology

Department of Computer Science & Engineering

GraphSAGE-LSTM Model

for Spatiotemporal Forecasting Based on COVID-19 Data

Submitted by: Arkapriya Ghosh

Roll No: 002110503050

Registration No: 160155 of 2021-2022

Exam Roll No: MCA2340013

Under the supervision of

DR. ANASUA SARKAR

Jadavpur University

Project submitted in partial fulfilments of the requirements for the degree of MASTER OF COMPUTER APPLICATION

Jadavpur University

Faculty of Engineering and Technology

Department of Computer Science & Engineering

CERTIFICATE:

This is to certify that the project "**GraphSAGE-LSTM Model for Spatiotemporal Forecasting Based on COVID-19 Data**" has been completed by **Arkapriya Ghosh** [Class Roll Number: 002110503050, Exam Roll Number: MCA2340013, Registration Number: 160155 of 2021-2022] as a part of curriculum of Master Computer Application Degree of the Department of Computer Science and Engineering, Jadavpur University. This work is carried out under the supervision of Dr. Anasua Sarkar, Assistant professor of Jadavpur university. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Dr. Anasua Sarkar
Project supervisor
Computer Science & Engineering
Jadavpur University

Countersigned:

Dr. Nandini Mukherjee
Head of the department
Computer Science and Engineering
Jadavpur University

Prof. Ardhendu Ghoshal
Dean
Faculty of Engineering and Technology
Jadavpur University

Jadavpur University

Faculty of Engineering and Technology

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project entitled as "**GraphSAGE-LSTM Model for Spatiotemporal Forecasting Based on COVID-19 Data**" Has been completed and submitted by **Arkapriya Ghosh** [Class Roll Number: 002110503050, Exam Roll Number: MCA2340013, Registration Number: 160155 of 2021-2022], for partial fulfilment of the requirements for completion of the degree Master of Computer Application under Department Of Computer Science and Engineering, Jadavpur University During the session 2021-2023. This work has been carried out under my supervision and this work is not submitted elsewhere for obtaining a degree.

EXAMINERs:

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION OF ORIGINALITY
AND
COMPLIANCE OF ACADEMIC ETHICS

This is to declare that the project entitled as "GraphSAGE-LSTM Model for Spatiotemporal Forecasting Based on COVID-19 Data" contains all original work done by the undersigned candidate, as part of her Master of computer Application studies, under the guidance of Dr. Anasua Sarkar. This resource contains all the resources identified to the best of knowledge and contains no unacknowledged material those are not original to this paper.

Name: Arkapriya Ghosh
Roll No: 002110503050
Exam Roll No: MCA2340013
Registration Number: 160155 of 2021-2022
Project Title: GraphSAGE-LSTM Model for Spatio-Temporal Forecasting Based on COVID-19 Data

----
Signature with date

ACKNOWLEDGEMENTS

Hereby, I would like to convey my gratitude to every person who have helped to complete this work successfully. I would like to convey my sincere thanks to my guide Dr. Anansua Sarkar, Assistant Professor, Department of Computer Science & Engineering, Jadavpur University, for her guidance and suggestions from the very beginning of this work. Her constant patience, help and support helped me to learn and evolve through this work in the entire period. My thanks to Dr. Nandini Mukherjee, HoD, Department of Computer Science & Engineering, Jadavpur University for all her support. My thanks to Jadavpur University for providing and maintaining the appropriate environment for research works. My gratefulness to all my teachers for always enlightening us with their valuable knowledge. Thanks to my classmates for all their help, support and appreciation. Lastly, gratitude for my parents and family members for their tireless effort towards my betterment and for believing in me.

Regards,
Arkapriya Ghosh
Roll Number: 002110503050
Exam Roll Number: MCA2340013
Registration number: 160155
Batch: 2021-2023

ABSTRACT

Covid 19 first appeared as a pneumonia type disease in China, at the end of 2019. By the middle of 2020 it has spread all over the world, taking shape of a deadly pandemic. The disease being extremely contagious and the virus mutating fast the infection rate were hard to reduce. This led to a prolonged lockdown for several months in stretch, some precautionary measures in the local level and in the country level, borders were sealed, immigrations were restricted. Healthcare system struggled to accommodate such huge number of patients, although maintaining the prrecautionsry measures and treating such unknown disease. Millions of people lost their lives, life and livelihood came in front of a tough challenge. In such situation prediction of the infection rate was important, as it could be helpful in devising the next steps to control and prevent the disease. A lot of researches were happening on predictions with various models of machine learning and deep learning. Many machine-learning models have been proposed and successfully applied for predictions. Here we collected global data from COVID-19 till date, containing daily new cases, new deaths, cumulative cases, and cumulative deaths country wise, state wise, region wise. Motivated from those we have chosen a version of graph neural network for the analysis of the data. The countries are represented as a big graph and daily data were chosen for the time series analysis.

CONTENTS

Chapter 1

1.Introduction:

1.1. Background Study:

1.1.1. Epidemiology: [1]

Epidemiology is the scientific, systematic, and data-driven study and analysis of the distribution (in terms of frequency or specific pattern) and determinants (driving factors, risk factors) of health-related states and events (not just diseases) in specified populations (locality, city, state, country, global). Epidemiological analyses are totally scientific and are usually based on systematic and thorough analysis of data collected from the target population or effected community. [2]

Among all the factors of epidemiology Geographical factors come most significant. The distribution of different factors in different region provides a great variability and community health factors tend to depend on distribution and variability of those factors. With variability comes the temporal factor of epidemiology. Different factors tend to change in pattern or distribution over time. So, consideration of time series in epidemiology opens up an all-new horizon of features affecting the community health over a period of time.

Since, spatial and temporal factors tends to pair up and are inseparable, a combined spatiotemporal analysis has become an inherent demand in epidemiology. Added to it, spatiotemporal analysis is capable to capture different environmental events and their effects over the stipulated period (seasonal analysis is also in play for handling seasonal changes).

1.1.2. Covid 19 as Pandemic:

In past few years the whole world has seen the breakout of Corona Virus Disease all over the world, which has put the animal and human kind in front of new sets of challenges with little known virology.

Coronaviruses are human and animal pathogens, extremely contagious, can be spread by air or droplets, causes pneumonia type respiratory syndrome.[3] A novel coronavirus was first identified at the end of 2019 identified as the cause of a clustered pneumonia cases in Wuhan, a city in the Hubei Province of China. It rapidly spread, resulting in an epidemic throughout China, followed by a global pandemic. In February 2020, the World Health

Organization designated the disease COVID-19, which stands for coronavirus disease 2019 [4]. The virus that was mainly behind COVID-19 is designated as severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2).

The primary transmission mode of SARS-COV-2 was respiratory transmission from person to person[5]. SARS-CoV-2 can also be transmitted longer distances through the airborne route (through inhalation of particles that remain in the air over time and distance), but the extent to which this mode of transmission has contributed to the pandemic is uncertain [6,7,8,9]. Scattered reports of SARS-CoV-2 outbreaks (e.g., in a restaurant, on a bus) have highlighted the potential for longer distance airborne transmission in enclosed, poorly ventilated spaces [10,11,12,13].

Almost all countries around the globe have implemented some preventive measures, according to WHO Standards, considering the ease of spread of covid 19. The respiratory transmission predominantly occurs within a diameter of 6 feet of the infected person. Therefore, using of mask and maintaining social distance was the most primitive way of prevention. Also, hygiene becomes a main issue in contagious disease. According to WHO Standards washing hands with soap, or using a hand sanitiser with at least 70% alcohol in case of no visible dirt was considered safe. Almost all countries implemented multiple instances of lockdown or quarantine.

In spite of all these measures the fatality of Covid 19 was not less. Till date there are around 766 million cases of infection and 6.9 million deaths as of May 2023[14]. There were evidently two waves of infection. The RNA virus got mutated several times, with more transmitting power. The spread finally came under control, but Covid is not gone. Such pandemics needs to be foreseen to prevent them and minimise the fatality and loss.

1.2. Motivation:

Covid 19 has shown an enormous effect on the whole world with to waves of spread, highly mutant virus and great transmission and resisting potential. Starting from preventive measures to treatment and vaccination, everything was under research. A number of times in several countries the healthcare system broke down, people suffered, people lost their lives. Economies dropped, because industries were closed for a long time. Market values of products declined. People lost their jobs in some sectors.

Its intuitive that such situations will be hard to afford if any other pandemic arrives again. Some precautionary measures can be deployed beforehand if the rate of infections or fatalities can be estimated beforehand.

The covid data is a record of daily cases, deaths and their cumulative values. This data contains several variations. The timeline of outbreak of covid 19 for all the countries were not same. The severity of Covid for all countries were different at different times. Based on the severity and rate of infection the government of various countries imposed different rules to check the infection. Countries like India Were far more affected than other countries.

Considering the facts like, varied time span of Covid phases in different countries, their different approaches towards lockdown and preventive measures, bases different rate of outbreaks different type of rules and actions the Covid data are varying throughout the time line. In this context every country is different and needs to be analysed differently. Again, countries are just geographical boundaries, there must be some dependence in terms of certain parameters like transportation, mobility, climate etc which affected the pattern of the disease. The time series also reflected the effect of preventive measures. Hence a macro-scale prediction needs to accommodate these changes. Microscale data are better fitted region wise but these are not recorded so thoroughly. Therefore, macroscale data will fit well for the deep learning models with incorporating proper parameters.

1.3. Literature Review:

A thorough literature review is performed to gain a good knowledge about the field; like, methods to deal with covid 19 data and time series analysis. From the beginning of this pandemic a lot of research work has been done to predict the covid cases, to analyse the diagnosis of covid, evaluate comorbidity etc. Since our field is limited to the prediction of covid 19 cases and deaths, we tried to summarise some popular works of this field in the following passages.

For forecasting the spread of covid 19 three types of approaches have been encountered. First is epidemiological, second one is population based and the third one is pure machine learning based approach. The pandemic years was a time was a time of crisis and every research finding tried to help to make up and prevent the losses. Due to our limitation, we may not be able to review all the papers but every work has great contribution towards the benefit of preventing covid and getting back to normal life with controlling and reducing the infection rates.

1.3.1. Epidemiological Models

Epidemiological models are traditionally used in the field of disease prediction and they are really helpful in certain cases. The SIR model is one of the simplest and most popular models. It is a set of three ordinary differential equations which try to describe the rate of change in relation to three different compartments in a particular population: Susceptible (*S*), Infected and infectious (*I*), and recovered and neither able to be infected again nor to spread the disease (*R*). During an epidemic episode some individuals move from *S* to *I*, and then, to *R*. The equations are intended to predict how the number of individuals in each compartment changes as epidemics evolve:

$$\frac{dS}{dt} = -\beta SI$$

$$\frac{dI}{dt} = \beta SI - \gamma I$$

$$\frac{dR}{dt} = \gamma I$$

The parameters *β* and *γ* are the "infectivity rate" and the "recovery rate", respectively. *β* depends on the number of contacts an infected individual has per time unit (*κ*) and the "transmission rate", that is, the probability of transmitting the infection to susceptible contacts (*τ*). Then, $\beta = \frac{\kappa \cdot \tau}{N}$. And $\gamma = \frac{1}{D}$, where *D* is the duration of the infection, measured in units of time (days, for instance). The SIR model calculates the number of infected people in a closed community. [15]

Zhihua Liu, Pierre Magal, Ousmane Seydi, Glenn Webb used SLR model with a little variation to predict the covid cumulative cases in China in the early days of epidemic (Feb, 2020) [16]. This model formulates the following features: (1) the importance of the timing and magnitude of the implementation of major government public restrictions designed to mitigate the severity of the epidemic; (2) the importance of both reported and unreported cases in interpreting the number of reported cases; and (3) the importance of asymptomatic infectious cases in the disease transmission.[16]

Ian Cooper, Argha Mondal, and Chris G. Antonopoulos implemented SLR model in across places and countries and communities. This model can give insights into the time evolution of the spread of the virus that the data alone does not. It can be applied to communities, can adjust to accommodate new

data, given reliable data are available. This model can also predict the success and failure of preventive measures implemented.[17]

Another approach is possible by using the SEIR model. SEIR is an epidemiological model used to predict infectious disease dynamics by compartmentalizing the population into four possible states: Susceptible [S], Exposed or latent [E], Infectious [I] or Removed [R]. N. Bannur, H. Maheshwari, S. Jain, S. Shetty, M. Srujana and A. Rava implemented SEIR model on the historical data of Covid and used Bayesian optimisation. This model expertise on a localised prediction of Covid.[18]

H. Gupta, S. Kumar, D. Yadav, O. P. Verma, T. K. Sharma, C. W. Ahn, et al developed an SIRD model as shown in the following figure. Here the parameters are susceptible to seasonal changes and can predict with up to 97% accuracy. [19]

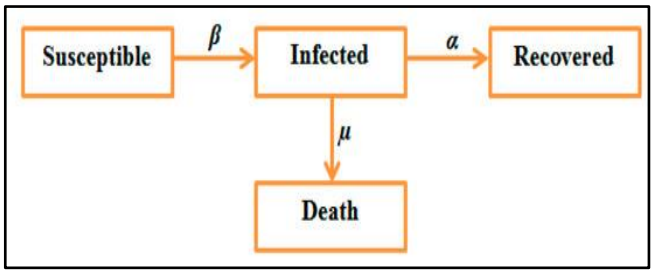


Fig 1.1: Schematic of SIRD model.[19]

The paper published by S. He, Y. Peng and K. Sun, a SEIR model is proposed for the COVID-19. Parameters of the system are estimated by the Particle Swarm optimisation algorithm, and dynamics of the system is investigated. Finally, how the parameters affect the dynamics of the system is discussed and the control strategies are presented. [20]

F. S. Lobato, G. B. Libotte and G. M. Platt presented a SIDR i.e., Susceptible, Infectious, Dead, and Recovered model for predicting the covid 19 behaviour in China. The parameters of this model are learned by solving both robust and nominal inverse problems. Stochastic Fractal Search algorithm is used to solve the nominal inverse problem. Robustness is incorporated in order to check perturbation of parameters while incorporating dynamisms in the model. [21]

L. Russo, C. Anastassopoulou, A. Tsakris, G. N. Bifulco, E. F. Campana, G. Toraldo, et al introduced to an all-new concept of tracing day 0 of the pandemic, and to find out the reasons why it started. This was done in a localised approach. Also, they performed a thorough tracing of asymptomatic cases. This paper was published in Oct,2020 and standing in the rising phase of pandemic this research gave better insights for modelling the preventions. [22]

[23] have used an extension of SEIR model, i.e. the SEI-HCRD compartmental model – Susceptible (S) → Exposed(E) → Infectious (I) → Removed (Hospitalized (H), Critical (C), Recovered (Rec), Dead (D)) and implemented a supervised deep learning model for simulating the pandemic situation and predicting the results, so that they can be used o plan an “Covid exit strategy”.

[24] shows a compartmentalised study of Covid 19 data and prediction based on SIRD (susceptible-infected-recovered-dead) model with implementing a networking model to incorporate the mobility data. This model used the data of France for the analysis and prediction.

1.3.2. Population or Agent Based Models

These models are basically SEIR based models that divided the whole population into compartments and uses machine learning or heuristic studies.

F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, D. Gutiérrez-Avilés, L. Melgar-García, R. Pérez-Chacón, et al. proposed a combined study model. This model uses pre-set parameters according to covid statistics and those are not changed after several iterations and proposes a multi-virus version to accommodate the evolution of corona virus and finally uses time series forecasting for prediction[25]. [26] is a similar model incorporating the climate factor.

[27] shows a model based on multiple epidemiological methods for prediction of covid 19 in United States and [28] shows the effect of time dependent parameters used to flatten the curve (reduce infection rate) of Covid in Kazakhstan.

1.3.3. Machine Learning and Deep Learning based Approaches

Several works have been done in machine learning and deep learning for analysing and revealing different aspects of covid 19. In this paper we are restricting the studies to only the covid data predictions.

The Auto-regressive Integrated Moving Average (ARIMA) model in machine learning is a model for time series analysis, based on RNN. [29] shows the use of ARIMA model in time series prediction of Covid 19 in European countries. [30] is a similar study based on Canada, using multi-layer perceptron model with a LSTM model at baseline.

[31] assessed the performance of 3 machine learning models, which are hidden Markov chain model (HMM), hierarchical Bayes model, and long-short-term-memory model (LSTM) using the root-mean-square error (RMSE) metric for calculating error. The LSTM model showed the smallest error rate and the hierarchical Bayes model provided capability of showing a plateau point in the infection growth curve.

[32] produces a study based on small datasets over six countries namely, Italy, Spain, France, China, USA, and Australia. The model uses comparative study of simple Recurrent Neural Network (RNN), Long short-term memory (LSTM), Bidirectional LSTM (BiLSTM), Gated recurrent units (GRUs) and Variational AutoEncoder (VAE) algorithms for global forecasting of COVID-19 cases.

A multiple time-series based analysis was published in [33] using eXogeneous variables of Covid 19. This paper shows three approaches. The first one is spatiotemporal graph neural networks with accommodating time series and mobility data. The second one is based on a statistical model Seasonal ARIMA and eXogenous variables. This one compares the SARIMA model with another model called 3.2 Minimax Concave Penalty (MCP). Third one is the implementation of GCN-LSTM which shows drastic improvement in accuracy, and also give comparable results to Augmented Neural network.

Furthermore, [34] demonstrates an approach using a multi- time series deep LSTM network trained on daily cases, mobility data and other parameters which produces predictions comparable to a multi-method ensemble model.

With the research trend of predicting covid 19 graph neural network and time-series models have gained extreme popularity in machine learning and with further improvements they are showing very accurate performance in analysis and predictions.

1.4. Objective:

Using country wise data of infections and fatalities all over the world we investigate the macro scale infection rate all over the world in macro scale. For this purpose, two macro scale machine learning models are implemented with appropriate techniques and architecture, to capture the geographical boundaries and connections, and variance of the data with time.

The main focus area of this project is the spatiotemporal analysis of the data and predicting the future. For the analysis of spatial dependencies, the data

is represented as a graph depicting the interconnections among the countries, with each country (nodes in graph) having their time series data as features.

Our model consists of two neural networks, one is the GraphSAGE convolution network and the other is the Long Short Term Memory Model. These two models will be discussed in depth in the methodology section, with analysis of the architecture and the obtained results.

Chapter 2

2. Methodology

2.1. Neural Networks

Neural Networking is a concept in machine learning that is inspired by the information flow through brain neurons. The most basic building block of a neural network is known as perceptron. The architectural view of a perceptron resembles a brain neuron consisting of an input unit, processing unit, and output unit. The figure shows the similarity of brain neurons and a perceptron.

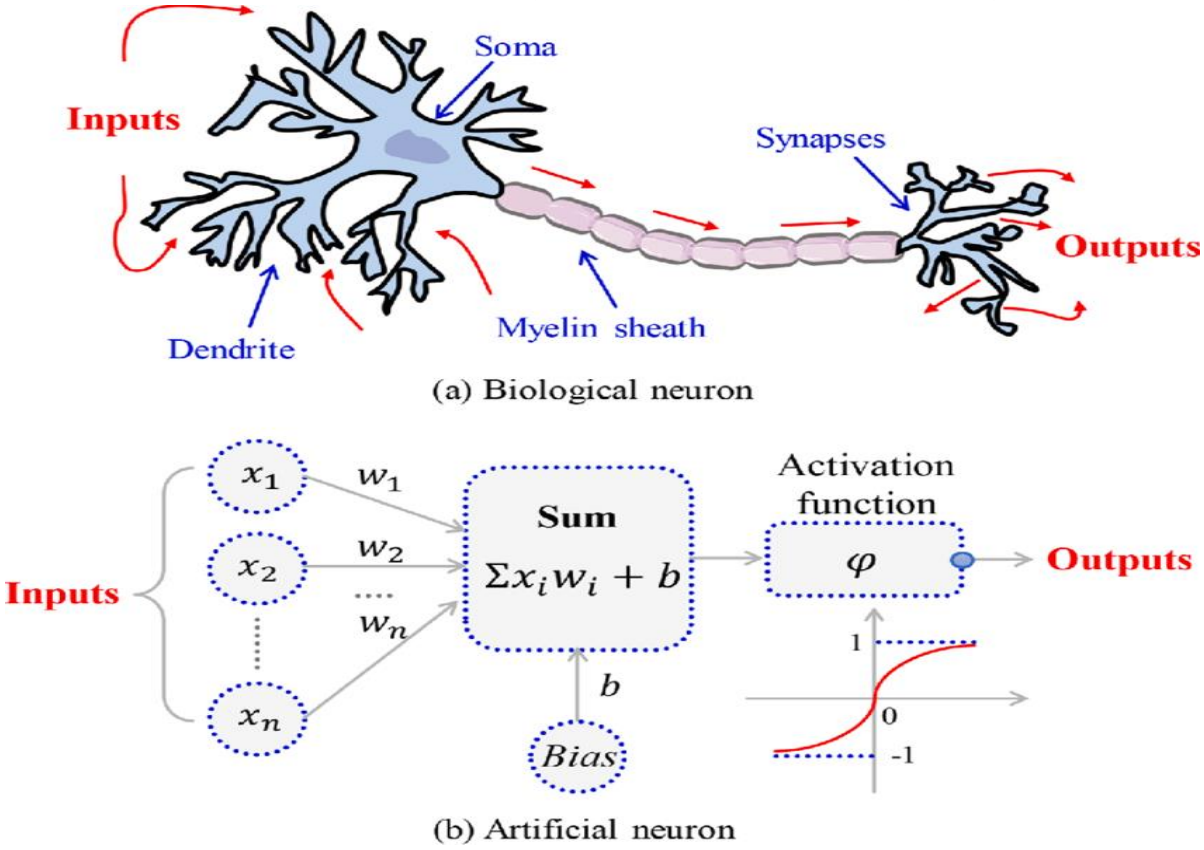


Fig 2.1: Resemblance of neuron and perceptron[35]

With clustering and layering of perceptrons with appropriate functionalities neural networks aim to map the inputs to it to the outputs comparable to the reaction of brain to various sensory inputs. Neural networks implement various computational functions with forward and backward passing of data and variables throughout the network. There can be several input and output pairs. Each pass of data variables through the neural network is known as one epoch.

Deep learning models that are similar to the ones implemented in this paper implement Feedforward neural networks (FNNs). FNN tries to optimiser

and implement a function f that tries to map the input variable x to the output variable y, by feeding forward the input data through the network until it reaches y. A FNN can be typically represented as:

$$f(x) = f_1 (f_2 (... ...f_{n-1} (f_n).....))$$

This formula shows a n layer Neural Network where i^{th} layer functionality is shown by f_i. Each layer contains a set of nodes, takes in the output from the previous layer or input and produces the output. Inside each layer the input is multiplied by a set of weights and biases which are learned by the network while approximating the target function.

2.1.1. Graph Convolution

The concept of graph convolution is to apply the functionalities of Convolutional Neural Network to the graphs. A convolution neural network is invariably a neural network with some special property to analyse spatial data. CNN is mainly used in image processing. The typical architecture of CNN consists of an input layer, several convolution layers, at least one ReLU layer (to avoid flattening to linear data), one fully connected layer to produce the output(classification). In CNN the neurons are not connected to all their previous and next layer neurons. They are rather connected to the neurons relatively similar or closer to each other. Thus, the network divides the data into smaller parts for optimized processing. The main characteristic of the convolution layer is parameter sharing. In CNN an image is looked at as a 2D vector of pixels. A kernel vector is moved over the image and a feature vector of a different size is generated (convolution). Within the kernel vector the feature of each pixel is influenced by its surrounding pixels.

CNNs by architecture are meant to operate on structured data, i.e., Euclidean graphs whereas graphical representation of real-world data are hard to fit in the Euclidean structure. The connectivity varies and the nodes are usually unordered, there may be some nodes or edges with special features, graphs may have forest like structure, or may be just randomly generated as subgraphs. Graph Neural Networks (GNNs) are more generalised versions of CNN that can be applied on graph datasets. One variant of GNN is known as Graph Convolution Network (GCN) which performs the similar task like CNN by learning the features through the inspection of neighbouring nodes.[36]

GCN uses a convolution of facts derived from classifiers and features. It does this through a concatenation of inputs from node features and graph adjacency matrices. The GCN takes in feature inputs X from N nodes across D features and a graph adjacency matrix A. Its output Y therefore, is given mathematically as:

$$Y = G_{M,A}^{B}(X) = Re\,LU(A \times B + \bar{C})$$

Where M is the number of output nodes, B is the state transition weight between input and output, and C is the prediction bias (also known as the graph structural offset). GCNs are adapt at dealing with static graphically structured information. However, it lacks the capability to account for temporal changes of these structures through time. Thus, dynamic vertex and edge information cannot be effectively represented using a GCN framework.[37]

Two major drawbacks of GCN are, firstly GCN is memory intensive as the whole graph need to be present in memory, secondly the graph structure should be known beforehand.[38]

2.1.2. GraphSAGE

The GNN model used in this project is GraphSAGE (Graph SAmple and aggreGatE). GraphSAGE was developed by Hamilton, Ying, and Leskovec (2017) and built on top of the GCN[38]. GraphSAGE is a general, inductive framework that leverages node feature information (e.g., text attributes) to efficiently generate node embeddings for previously unseen data. Instead of training individual embeddings for each node, it generates node embeddings by sampling and aggregating features from a node's local neighbourhood. [39]

GraphSAGE is based on Sampling and Aggregation. Unlike GCN, GraphSAGE does not need to process the whole graph together. It rather implements methods to sample a neighbourhood for each node, then aggregates the features of neighbouring nodes to modify the current node features.

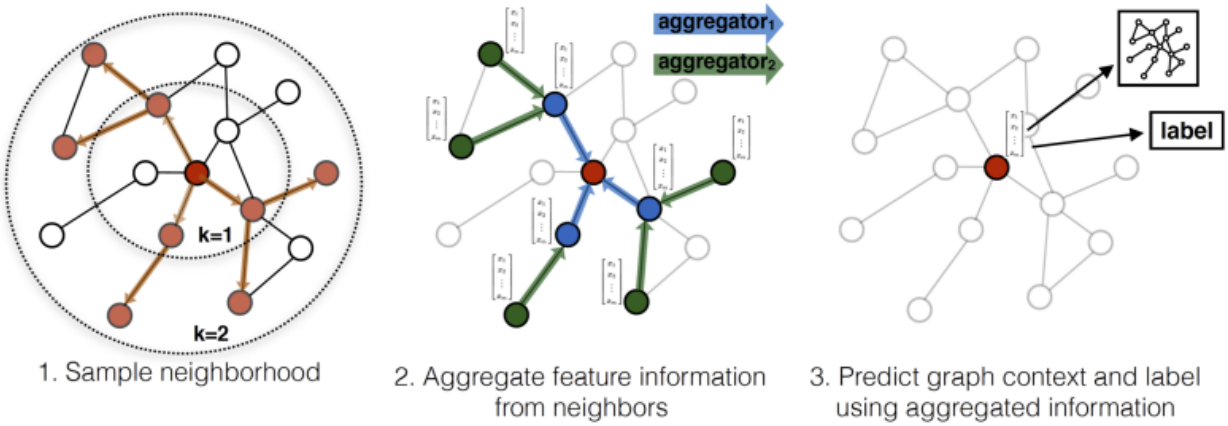


Fig 2.2: Illustration of Sampling and Aggregation in GraphSAGE[39]

The GraphSAGE model proposed by Hamilton, Ying, and Leskovec consists of three main steps. Those described as follows:

i) Embedding generation: This is the forward propagation step where it is assumed that the parameters of K aggregator functions (denoted AGGREGATE_k, $\forall k \in \{1, ..., K\}$) is already learned. This function aggregate information from node neighbours, as well as a set of weight matrices W^k, $\forall k \in \{1, ..., K\}$, which are used to propagate information between different layers of the model or "search depths".

Algorithm 1: GraphSAGE embedding generation (i.e., forward propagation) algorithm

Input : Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; input features $\{\mathbf{x}_v, \forall v \in \mathcal{V}\}$; depth K; weight matrices $\mathbf{W}^k, \forall k \in \{1, ..., K\}$; non-linearity σ; differentiable aggregator functions $\text{AGGREGATE}_k, \forall k \in \{1, ..., K\}$; neighborhood function $\mathcal{N} : v \rightarrow 2^{\mathcal{V}}$

Output : Vector representations $\mathbf{z}_v$ for all $v \in \mathcal{V}$

$\mathbf{h}_v^0 \leftarrow \mathbf{x}_v, \forall v \in \mathcal{V}$;
for $k = 1...K$ **do**
  for $v \in \mathcal{V}$ **do**
    $\mathbf{h}_{\mathcal{N}(v)}^k \leftarrow \text{AGGREGATE}_k(\{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\})$;
    $\mathbf{h}_v^k \leftarrow \sigma\left(\mathbf{W}^k \cdot \text{CONCAT}(\mathbf{h}_v^{k-1}, \mathbf{h}_{\mathcal{N}(v)}^k)\right)$
  end
  $\mathbf{h}_v^k \leftarrow \mathbf{h}_v^k / \|\mathbf{h}_v^k\|_2, \forall v \in \mathcal{V}$
end
$\mathbf{z}_v \leftarrow \mathbf{h}_v^K, \forall v \in \mathcal{V}$

Fig 2.3: Original embedding generation algorithm [39]

ii) Learning the parameters of GraphSAGE: In unsupervised settings a graph based loss function, z_u, $\forall u \in V$ is applied, and the weight matrices, W^k , $\forall k \in \{1, ..., K\}$, and parameters of the aggregator functions are tuned via stochastic gradient descent. The graph-based loss function encourages nearby nodes to have similar representations, while enforcing that the representations of disparate nodes are highly distinct:

$$J_{\mathcal{G}}(\mathbf{z}_u) = -\log\left(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)\right) - Q \cdot \mathbb{E}_{v_n \sim P_n(v)} \log\left(\sigma(-\mathbf{z}_u^\top \mathbf{z}_{v_n})\right)$$

Fig 2.4: Equation of loss function [39]

Here, v is a node that co-occurs near u on fixed-length random walk, σ is the sigmoid function, P_n is a negative sampling distribution, and Q defines the number of negative samples. Importantly, unlike previous embedding approaches, the representations z_u that we feed into this loss function are generated from the features contained within a node's local neighbourhood, rather than training a unique embedding for each node (via an embedding look-up) [39].

iii) Aggregator: The purpose of the aggregator function is to learn the features of the neighbouring nodes. Ideally an aggregator function should be symmetric but graph nodes have no natural ordering hence, aggregator should be capable to fit on unordered vectors. In the paper "Inductive Representation Learning on Large Graphs" the authors have proposed 3 types of aggregators.

Mean Aggregator: Takes the element wise mean of the vectors in

$$\{h_u^{k-1}, \forall u \in N(v)\}.$$

The mean aggregator is nearly equivalent to the convolutional propagation rule used in the transductive GCN framework.[40]

$$\mathbf{h}_v^k \leftarrow \sigma(\mathbf{W} \cdot \text{MEAN}(\{\mathbf{h}_v^{k-1}\} \cup \{\mathbf{h}_u^{k-1}, \forall u \in \mathcal{N}(v)\}$$

Fig 2.5: Expression for Mean Aggregator [39]

LSTM Aggregator: LSTMs have larger expressive capability. But LSTMs are not inherently symmetric (i.e., they are not permutation invariant), since they process their inputs in a sequential manner. To operate on an unordered set, LSTMs are applied to a random permutation of the node's neighbours.

Pooling Aggregator: In this pooling approach, each neighbour's vector is independently fed through a fully-connected neural network; following this transformation, an elementwise max-pooling operation is applied to aggregate information across the neighbor set:[39]

$$\text{AGGREGATE}_k^{\text{pool}} = \max(\{\sigma\left(\mathbf{W}_{\text{pool}}\mathbf{h}_{u_i}^k + \mathbf{b}\right), \forall u_i \in \mathcal{N}(v)\}),$$

Fig 2.6: Expression for Max Aggregator [39]

Our model has used the mean aggregator, sampling has been done in reference to the adjacency matrix of our graph, and the modified feature vector is used for further analysis.

2.2. Time Series Analysis

The main challenge of the project is capturing the time series variation of data. Graph based algorithms capture the spatial dependencies, but are unable to process time series data. Time Series Data Analysis is a way of studying the characteristics of the response or target variable with respect to time as the independent variable. Time variable act as the point of reference based on which target variables are forecasted. A Time-Series represents a series of time-based orders that can be Years, Months, Weeks, Days, Hours, Minutes, and Seconds. It is an observation of data in form of a sequence of discrete time of successive intervals.[41]

Time series analysis adds an explicit order dependence on data. Analysis can be of two types - namely, Descriptive and Forecasting.

In descriptive modelling, a time series is modelled to determine its components in terms of seasonal patterns, trends, relation to external factors etc. In contrast, time series forecasting uses the information in a time series (perhaps with additional information) to forecast future values of that series.[42]

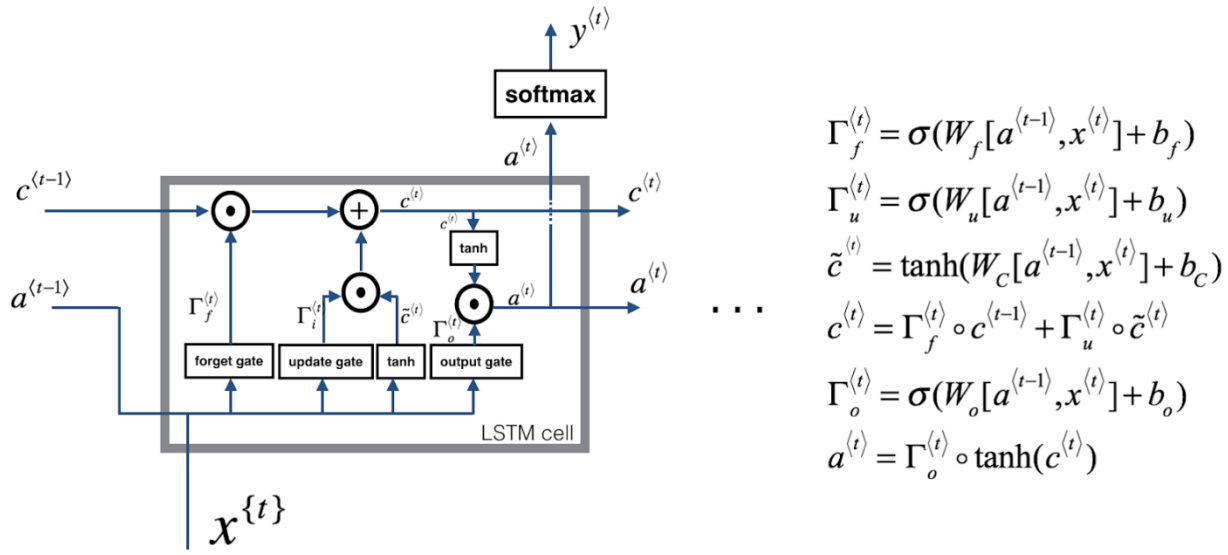
The objective of this project is based on time series forecasting. Forecasting is done by fitting a model on a set of past data to learn the pattern and dependencies to be able to estimate the future.

2.2.1. Long Short Term Memory

Long short term memory or LSTM is a type of Recurrent Neural Network (RNN) that can process sequential data. LSTM finds its use in case of sequential analysis and forecasting. Being a Recurrent Neural network, a LSTM node takes

the parameters from a previous step or input data to produce the next step and then in the next cycle the produced output is again passed into the network along with the next input. The speciality of LSTM nodes is working memory area known as state, which learns and store the history, hyperparameters from the input data and gets updated in each step. A LSTM node also contains gates to control flow of input data inside the node. These are weights, biases, activation functions.

The state of the LSTM network is also known as cell state which holds the main functionality of LSTM. A basic LSTM unit is composed of a cell, an input gate, an output gate and an forget gate. The cell remembers the history and the gates control the flow of information in the unit. Thus, the long short term Memory concept is implemented with memory of the cell. The forget gates are used to remove biases towards the recent5 events but this holds risk of inclination towards recent data.[43]



$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$
$$\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$$
$$\tilde{c}^{\langle t \rangle} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$
$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$
$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$
$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Fig 2.7: LSTM unit with functional formulas [44]

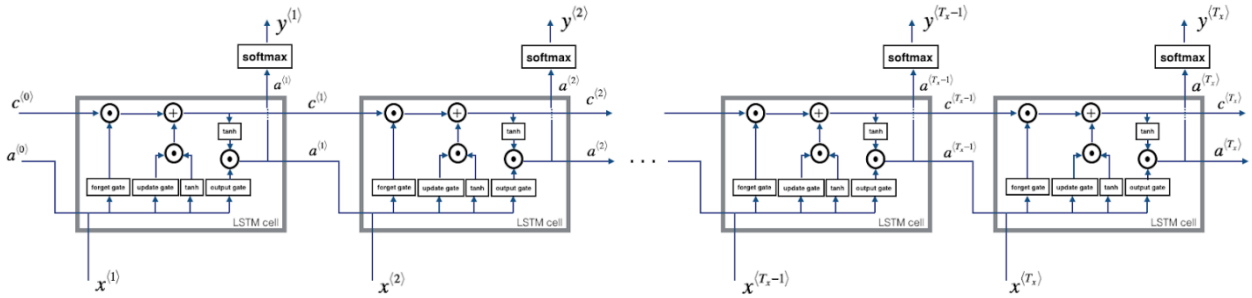


Fig 2.8: LSTM forward pass [44]

2.3. Activation Function

Activation function or transfer functions are used in Neural Networks to map the weighted input to the required range of outputs. Activation functions creates a weighted sum of inputs and other parameters and adds up the bias to it to produce the output. Activation function adds the non- linearity to the neural network, otherwise a neural network will be just a linear regression model. Based on the inaccuracy of output the weights and biases are modified, this modification proceeds backwards the neural network. This is known as back propagation. Activation function provides thew gradient and error thus eases the process of backpropagation.

2.3.1. ReLU

ReLU stands for rectified linear unit. This is a linear unit activation function that is able to interpret only the positive part of data. The functional formula of ReLU is given by

$$f(x) = \max(0, x) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Here, $f(x)$ ranges from [0, ∞]. The output of ReLU is the positive data of the input, if the input is negative the output is zero.

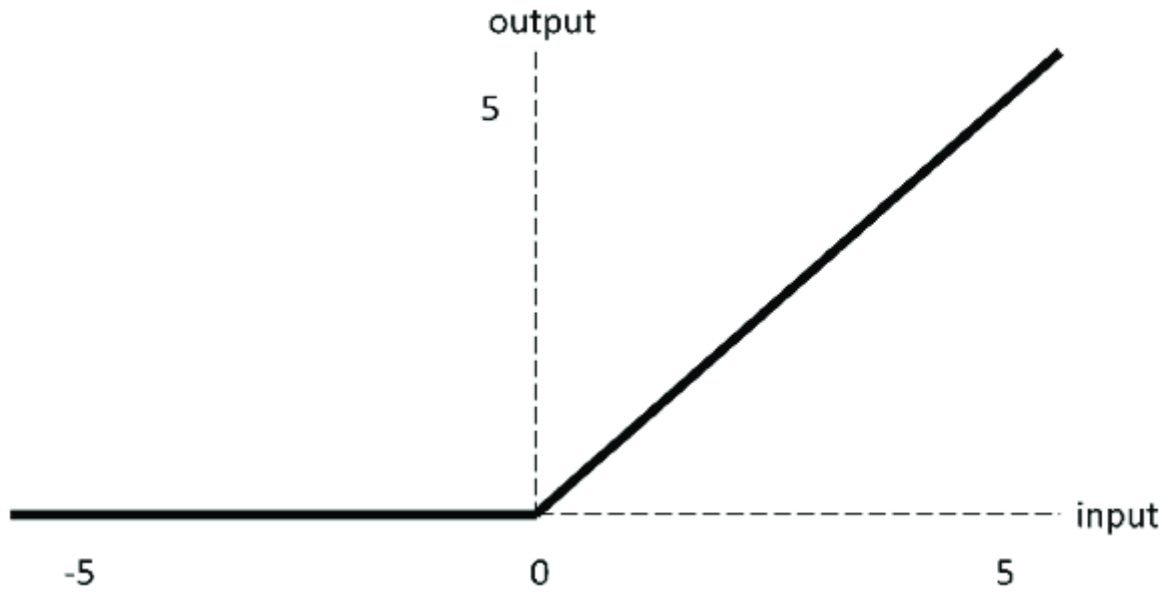


Fig 2.9: ReLU activation function[45]

ReLU is differentiable everywhere except 0. Thus, ReLU solves the vanishing gradient problem. The gradient of ReLU is 1 everywhere except 0. ReLU has less computational complexity compared to sigmoid and tanh. ReLU learns

the input parameters in minimum time and incorporates minimum processing. It is usually applied in the hidden layers. After applying ReLU only a few neurons produce required output, others are turned off. Thus, ReLU promotes optimized and efficient processing. Also, the easy implementation of ReLU makes backpropagation easier.

2.3.2. Sigmoid

Sigmoid function another activation function with a characteristic "S" shaped curve, popularly used in probabilistic and classification problems. Sigmoid function maps the input data to a range of [0,1]. The function of sigmoid function is given by:

$$f(x) = \frac{1}{1+e^{-x}}$$

The graph of the Sigmoid function is given below.

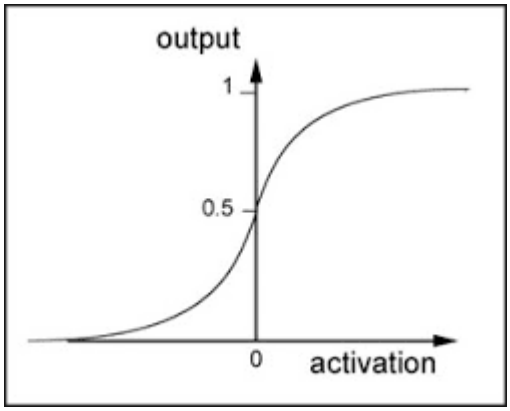


Fig 2.10: Sigmoid Activation function[46]

It is usually applied in the output layers. The result of sigmoid function is approximated to 0 if it is <0.5, and approximated to 1 if it is >0.5. The main issue with sigmoid function is when the function reaches the horizontal end of the curve the gradient does not exist. This is known as vanishing gradient problem.

2.3.3 tanh

Another activation function that is common in deep learning is the tangent hyperbolic function simply referred to as tanh function. It is calculated as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{2x}}{1 + e^{2x}} = \frac{2 - (1 + e^{-2x})}{1 + e^{-x}} = \frac{2}{1 + e^{-x}} - 1$$
$$= 2.Sigmoid(2x) - 1 = 2(\,Sigmoid(2x) - \frac{1}{2}\,)$$

The above calculation evidently shows that tanh is a modified and shifted version of sigmoid activation function. Plot of tanh when the input is in the range [-10,10]:

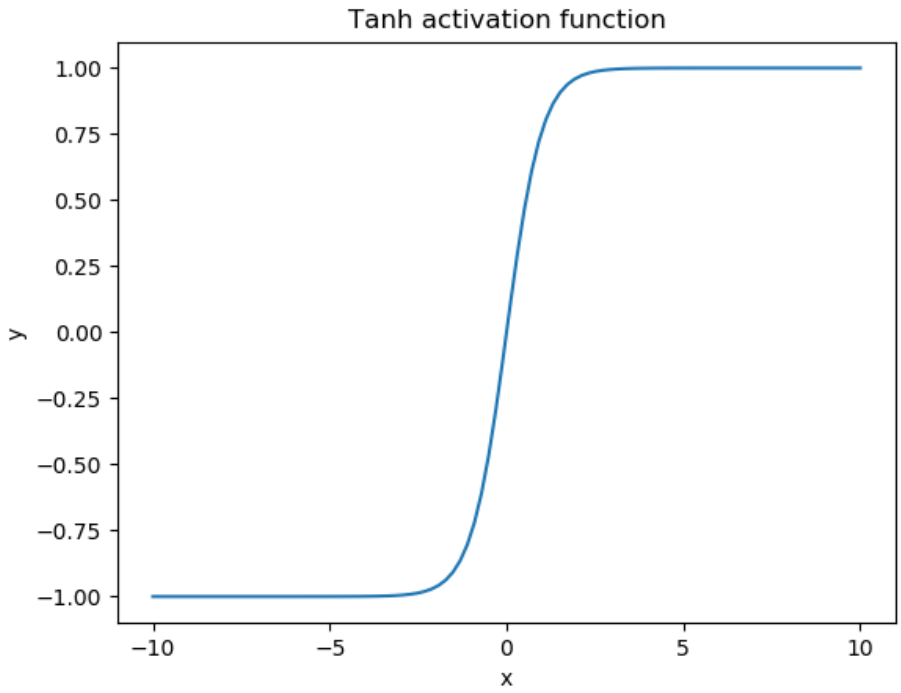


Fig 2.11: tanh Activation function [49]

The output range of the tanh function is [-1,1] and shows similar characteristics like sigmoid function. The main difference is the fact that the tanh function pushes the input values to 1 and -1 instead of 1 and 0.

2.4. Loss Function

Loss function is a measure of the performance of any machine learning model. The Loss function computes the error by comparing the expected and predicted value. This generated error is used by the neural networks in the backpropagation procedure to learn data features, modify weights and biases. Just like activation function there are different loss functions available. They are chosen based on the problem and solution approach.

2.4.1. Mean Square Error (MSE)

Mean squared error is the simplest and most popular loss function in machine learning. This error computes the squared difference between the observed and predicted values. The formula is given by:

$$MSE = \frac{1}{n}\Sigma(actual - predicted)^2$$

MSE is a parabolic function. Higher value of MSE denotes data points are widely spread and away from the central or target value. Lower value denotes data points are close and clustered towards target. Lower value of MSE indicates better performance of the model. Since the difference of actual output and expected output is squared, the outliers have a lot of impact on the loss computation.

Thus, backpropagation procedure gives more importance to the outliers.

2.5. Optimisation Algorithms

Optimisation are algorithms used by the neural network during the learning step. The purpose of optimisers is to modify the parameters such as weights, biases, learning rate in order to reduce the loss. Gradient descent is the most commonly used optimisation in deep learning. It gradually changes the parameters in the way so that the loss function reaches minima. Learning rate determines the rate of change of these parameters.

2.5.1. Stochastic Gradient Descent

In our model we have used the Stochastic Gradient Descent technique. This model updates the network weights after each training example. This results in faster training. But the most challenging part is determining the appropriate learning rate. A rate that is too low will not converge, while a rate that is too large would cause too many variations, slowing down the learning process.

2.5.2. Adam Optimiser

Adam stands for adaptive moment estimation works with momentum of first and second order. This is to slow down the rate of learning parameters for a careful search. Adam stores the exponentially decaying average of past gradients and average of past squared gradients. It is a form of Stochastic Gradient Descent. [47]

Chapter 3

3. Importing Datasets and Preprocessing

3.1. Datasets

The data set that we have chosen for obtaining the records of covid infectious and deaths is the WHO Covid 19 Global Dataset available at the official website in updated version. This dataset contains the daily records of new cases, cumulative cases, new deaths and cumulative deaths of all countries worldwide starting from January 2020.

1	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
2	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
5	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
6	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0
7	2020-01-08	AF	Afghanistan	EMRO	0	0	0	0
8	2020-01-09	AF	Afghanistan	EMRO	0	0	0	0
9	2020-01-10	AF	Afghanistan	EMRO	0	0	0	0
10	2020-01-11	AF	Afghanistan	EMRO	0	0	0	0
11	2020-01-12	AF	Afghanistan	EMRO	0	0	0	0
12	2020-01-13	AF	Afghanistan	EMRO	0	0	0	0
13	2020-01-14	AF	Afghanistan	EMRO	0	0	0	0
14	2020-01-15	AF	Afghanistan	EMRO	0	0	0	0
15	2020-01-16	AF	Afghanistan	EMRO	0	0	0	0
16	2020-01-17	AF	Afghanistan	EMRO	0	0	0	0
17	2020-01-18	AF	Afghanistan	EMRO	0	0	0	0
18	2020-01-19	AF	Afghanistan	EMRO	0	0	0	0
19	2020-01-20	AF	Afghanistan	EMRO	0	0	0	0
20	2020-01-21	AF	Afghanistan	EMRO	0	0	0	0
21	2020-01-22	AF	Afghanistan	EMRO	0	0	0	0
22	2020-01-23	AF	Afghanistan	EMRO	0	0	0	0
23	2020-01-24	AF	Afghanistan	EMRO	0	0	0	0
24	2020-01-25	AF	Afghanistan	EMRO	0	0	0	0
25	2020-01-26	AF	Afghanistan	EMRO	0	0	0	0

Fig 3.1: WHO Covid 19 Global Data [14]

This dataset marks the countries with their names and a specific country code. To obtain the geographical distance among all countries a second dataset was used. This dataset is derived from the world_country_and_usa_states_latitude_and_longitude_values dataset. The original dataset is downloaded from kaggle.com [48]. This dataset contains the latitude and longitude values of all country capitals, countries are marked by their names and unique codes. This dataset also contains latitude longitude data of USA states but those were redundant for the project, hence are removed. After this the dataset looks as following:

1	country_code	latitude	longitude	country
2	AD	42.546245	1.601554	Andorra
3	AE	23.424076	53.847818	United Arab Emirates
4	AF	33.93911	67.709953	Afghanistan
5	AG	17.060816	-61.796428	Antigua and Barbuda
6	AI	18.220554	-63.068615	Anguilla
7	AL	41.153332	20.168331	Albania
8	AM	40.069099	45.038189	Armenia
9	AN	12.226079	-69.060087	Netherlands Antilles
10	AO	-11.202692	17.873887	Angola
11	AQ	-75.250973	-0.071389	Antarctica
12	AR	-38.416097	-63.616672	Argentina

Fig 3.2: Latitude and Longitude data of all countries [49]

With these two datasets the graph of our model is build and further analysis are done.

3.2. Data Pre-Processing

3.2.1. Normalisation

The GraphSAGE - LSTM model demands a graphical representation with the required data being embedded to the graph in appropriate relations. The spatial feature is captured in the graph as distances between countries. An adjacency matrix is used to record all these distances. The latitudes and longitudes are used to compute the distance between countries. The distances are normalized using min-max scaling, so that the adjacent nodes are proportional in terms of relation with other nodes. The following formula is applied for the normalisation process.

$$X_{norm} = \frac{(X - X_{min})}{(X_{max} - X)}$$

The network for our model is a connected graph structure. Each country is a node to the graph. The distances between the countries After min-max scaling the distances between the countries are rescaled so that they fall within the range of [0,1]. Now, the network contains the connected graph of all the countries in the world.

After all cleaning and processing and removing the erroneous values from the datasets, we get 227 countries, 51302 total distances, i.e., 25651 total edges in the undirected graph. This total spatial information is given as input to the GraphSAGE algorithm.

0	0.2111776941	0.3118602991	0.6934145332	0.2859237848	0.3643686446	0.6122955275	0.6125098521
0.2111776941	0	0.1116022849	0.8450967206	0.07729834629	0.2919757746	0.4075642169	0.4063293496
0.3118602991	0.1116022849	0	0.9153127627	0.08086587378	0.2354801716	0.3325271659	0.3287791386
0.6934145332	0.8450967206	0.9153127627	0	0.8405083147	0.8543184424	0.6146999704	0.6203657426
0.2859237848	0.07729834629	0.08086587378	0.8405083147	0	0.3108153403	0.330328153	0.3292315301
0.3643686446	0.2919757746	0.2354801716	0.8543184424	0.3108153403	0	0.4740259284	0.4659347488
0.6122955275	0.4075642169	0.3325271659	0.6146999704	0.330328153	0.4740259284	0	0.009348557369
0.6125098521	0.4063293496	0.3287791386	0.6203657426	0.3292315301	0.4659347488	0.009348557369	0
0.7851803825	0.6139660784	0.5023858258	0.5215235737	0.5586942199	0.4259418252	0.315602518	0.3092736761
0.1061608549	0.1050170957	0.2088797321	0.7795562773	0.1803399119	0.3183035574	0.5093632408	0.5088461972
0.6605283921	0.4563562359	0.3792950481	0.5724419672	0.3791263058	0.5034321467	0.04879920202	0.05076197594
0.4804137986	0.6888124164	0.7655448533	0.298386195	0.7655607029	0.6002903839	0.9074786084	0.9101479508

Fig 3.3: Clip from the adjacency matrix

Currently, our focus lies on the daily count of new COVID-19 cases. To analyse this information, we handle a dataset that includes figures for both cases and deaths. The dataset is structured in a way that it comprises 236 columns representing different countries and 704 rows corresponding to dates spanning from March 3rd, 2020 to February 4th, 2022. Once we complete the processing, the resulting data is presented in the following manner:

1	2020-03-03	2020-03-04	2020-03-05	2020-03-06	2020-03-07	2020-03-08	2020-03-09	2020-03-10	2020-03-11	2020-03-12
2	0	0	0	0	0	0	0	0	0	0
3	7671	7673	7673	7675	7675	7675	7675	7676	7676	7676
4	2192	2204	2210	2216	2227	2235	2241	2247	2256	2265
5	0	0	0	0	0	0	0	0	0	0
6	6874	6874	6874	6875	6875	6875	6875	6875	6875	6875
7	0	0	0	0	0	0	0	0	0	0
8	16	17	17	21	22	24	26	26	28	29
9	153	153	153	153	153	153	153	153	153	153
10	591	594	596	600	602	603	609	618	622	628
11	0	0	0	0	0	0	0	0	0	0
12	9	9	9	9	9	9	9	9	9	9
13	42	42	42	42	42	42	42	42	42	42
14	583	607	622	634	646	662	687	706	729	747
15	128973	128973	128973	128994	128994	128994	128994	128994	128994	128994

Fig 3.4: Country wise daily records of Cumulative deaths of Covid 19

3.2.2. Graphical View

A graph data structure shows the relation among the vertices or nodes by connecting them through ages. In order to show the spatial dependency among countries world wide they are represented in form of a graph. As mentioned earlier our graph contains 227 vertices and 25651 edges in total. This is certainly a huge graph to study. But the graph being densely connected, gives a chance to accommodate features from distantly related countries if feasible. Hence, potential of this graph can be extended even out of the scope of this project. A visualisation of this graph is given below.

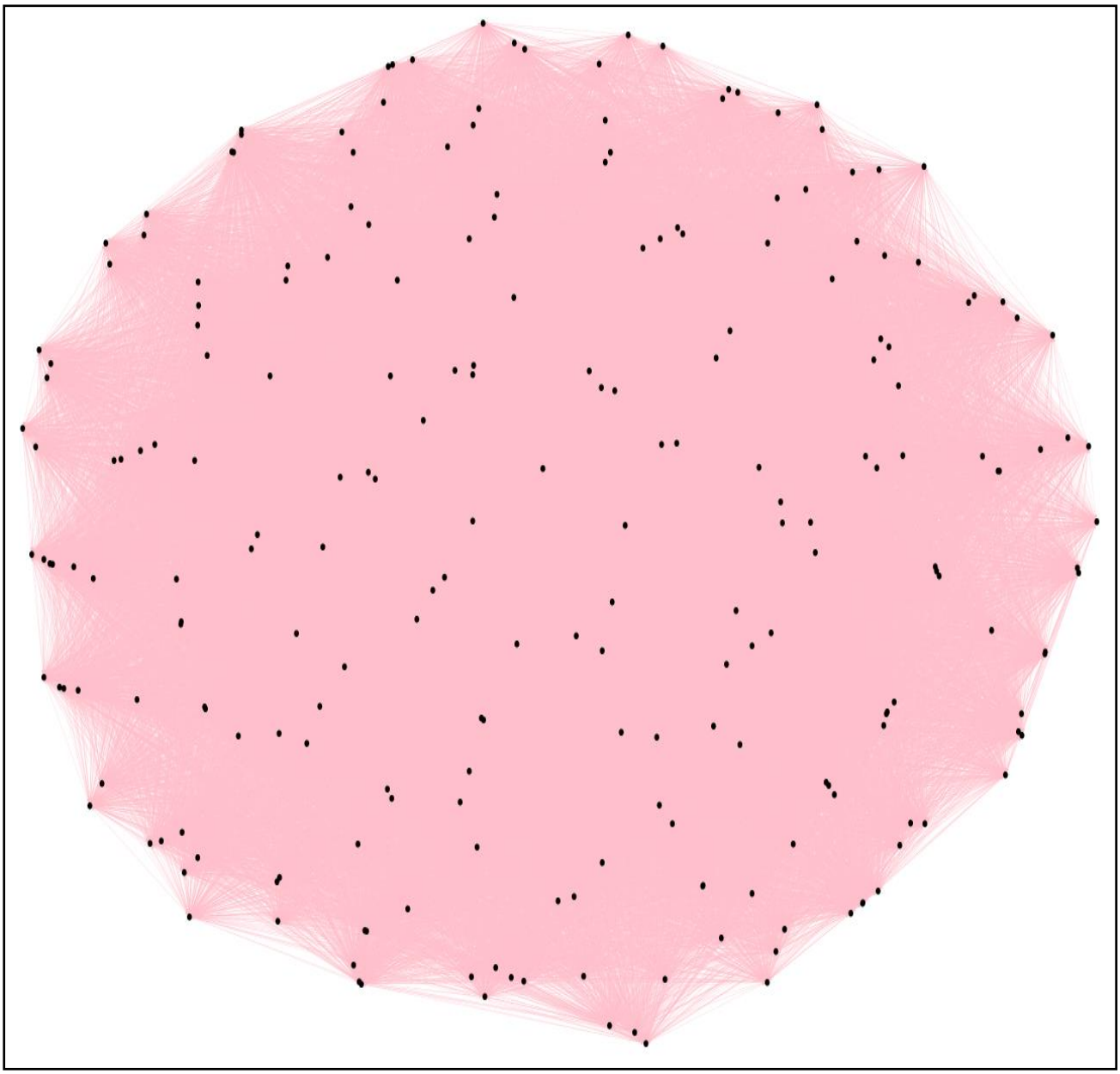


Fig 3.5: Connected graph of 227 countries

Chapter 4

4. Experiments

4.1. GraphSAGE-LSTM Model

The GraphSAGE-LSTM model is implemented in python. For our project the model is implemented as a 3-layer GraphSAGE and 3-layer LSTM. The GraphSAGE trains the graph, processes the features after that the output GraphSAGe is given to LSTM to prowess and learn the sequential data. The detailed architecture along with all the parameters are given in the following figure.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 227, 5)]	0
tf.expand_dims (TFOpLambda)	(None, 227, 5, 1)	0
reshape (Reshape)	(None, 227, 5)	0
graph_sage_convolution (GraphSAGEConvolution)	(None, 227, 32)	51916
graph_sage_convolution_1 (GraphSAGEConvolution)	(None, 227, 32)	52780
graph_sage_convolution_2 (GraphSAGEConvolution)	(None, 227, 32)	52780
reshape_1 (Reshape)	(None, 227, 32, 1)	0
permute (Permute)	(None, 32, 227, 1)	0
reshape_2 (Reshape)	(None, 32, 227)	0
lstm (LSTM)	(None, 32, 64)	74752
lstm_1 (LSTM)	(None, 32, 32)	12416
lstm_2 (LSTM)	(None, 16)	3136
dropout (Dropout)	(None, 16)	0
dense (Dense)	(None, 227)	3859

Total params: 251,639
Trainable params: 97,052
Non-trainable params: 154,587

Fig 4.1: Summary of GraphSAGE – LSTM model

4.2. Training And Testing

The data we used are labelled for spatiotemporal analysis but these labels are not completely usable by our model. Hence, a semi-supervised learning is implemented for node classification and time series forecasting. The data is split for training and testing. The training data is used to train the model and prediction were performed on testing data to check the performance of the model.

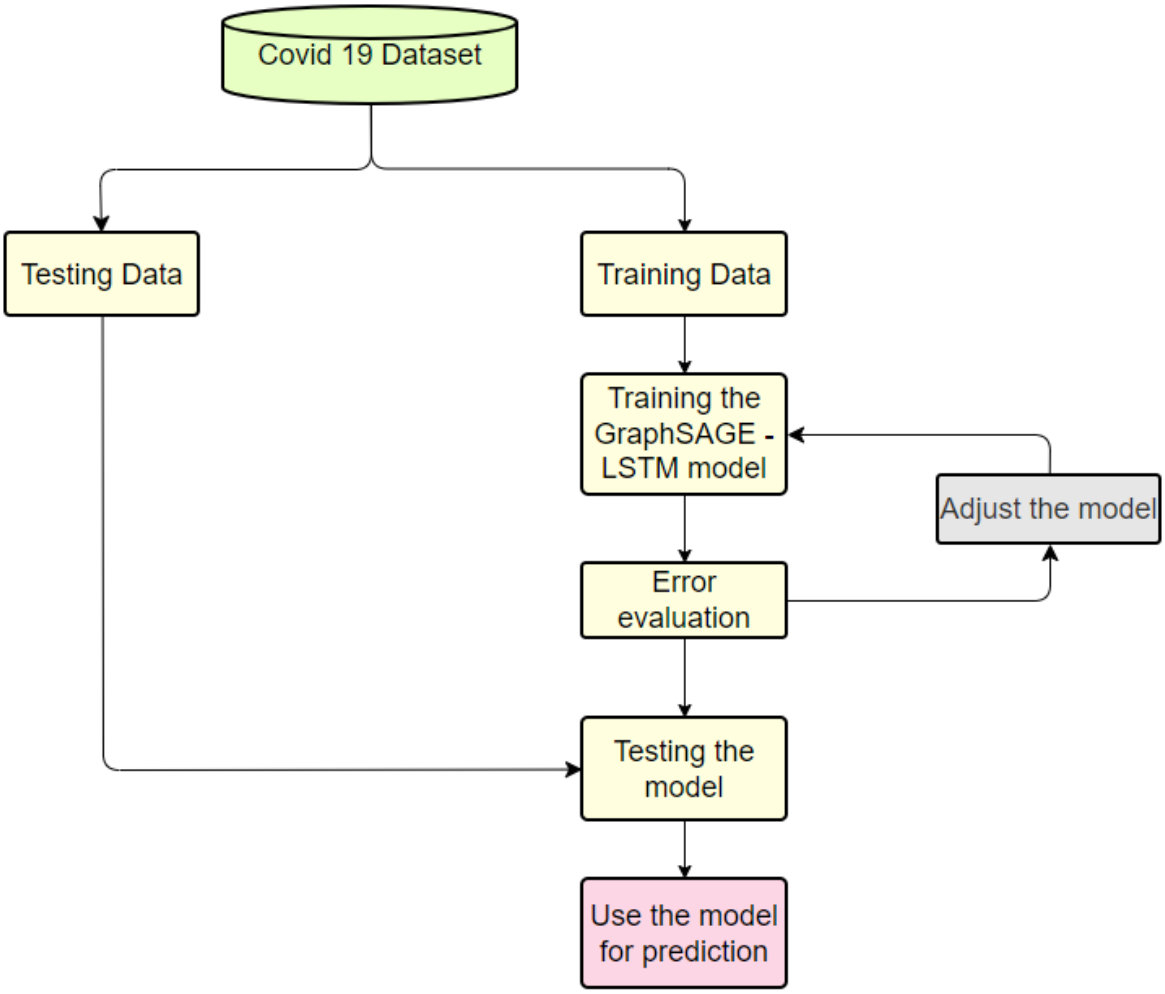


Fig 4.2: Work flow of GrahSAGE-LSTM model

4.3. Hyperparameters

Hyperparameters play a great role in the training process of each model. Hyperparameters are essentially the parameters that the neural network highly depends on. Hyperparameters controls the learning process the learning rate the, the learning procedure, the batch size, even the performance of the model. The main challenge is to make the right choice of parameters. The parameter choosing is indeed a procedure which is done by running the model in a loop of by several trials until the desired output is obtained, or the results are satisfactory. Different hyperparameters are required depending on the requirement of the analysis, whether it is a deep learning model or a classical machine learning model.

The parameters used in this model are learned after several trials and rebuilding. The final values of the parameters which produced the most satisfactory values are discussed below.

The GraphSAGE -LSTM is adjusted based on our dataset and feature definition with the help of certain hyperparameters. The parameters required by GraphSAGE are neighbourhood size for each node to which the neighbours of the node will be sampled. Also, the type of activation function to be used and the sizes of each layer is given as parameters. For the LSTM layers size of each layer, sequence length, prediction length and activation functions are given as input. The sequence length refers to the number of prior days which will be utilised to predict the data of the coming days. The number of days the data will be predicted is specified in prediction length. An optimiser needs to be defined for the model so that it can optimise the results, loss and learning. For the optimiser learning rate act as a hyperparameter and it is an important hyperparameter to control the performance of the model by controlling the gradient descent. While compiling the model a loss function needs to be specified that will compute the error and readjust the model.

The sequence length is set to 5 and prediction length is set to 100. ReLU is used as an activation function for GraphSAGE. Sigmoid and tanh are used in LSTM. For optimisation Adam optimiser is used, with a learning rate of 0.001. For loss function we used Mean Absolute Error, Mean Squared error and Mean Squared Logarithmic Error because we are trying to reduce a statistical function in this model.

The number of epochs is set to 100, and the amount required to make the loss curve converge is determined. Finally, the stochastic gradient descent algorithm's batch size is set to 10.

Hyperparameter	**Values**
Sequence length	10
Prediction length	1
GraphSAGE activation function	ReLU
LSTM activation function	tanh, Sigmoid
Optimiser	Adam
Learning Rate	0.001
Loss	MSE
Epoch	100
Batch Size	40

Table 4.1: Hyperparameters with their values

Chapter 5

5. Results

5.1. Runtime

The runtime of a model is highly crucial in performance analysis. Although the runtime is highly dependent on the device hardware that is being employed for the project. For any model or algorithm, we try to optimise it based on space and time complexity. The computer used for the implementation has the following specifications.

Processor: 12th Gen Intel(R) Core (TM) i5 12400 2.5GHz
RAM size: 16GB
System type: 64-bit operating system, x64-based processor
Operating System: Windows 10 Pro
Version: 21H2

Runtime are also important in terms of usability of the model. A very complex model with great space and time complexity is not much widely usable. Hence, checking and optimising the runtime is important as far as feasible.

The runtime of the GraphSAGE-LSTM model is = **3 min 10 sec**

Although this looks quite long but considering the limitations of personal computer, the editor and compiler, and more over the huge complexity of the model with 3 layers of GraphSAGE and 3 big layers of LSTM the runtime quite satisfactory. The accuracy this model shows is much better and this outweighs the long runtime.

5.2. Model Performance

As mentioned earlier, we have chosen the Mean Squared Error (MSE) as our loss function for the task at hand. Our goal is to predict the number of new COVID-19 cases worldwide using daily data specific to each country. To achieve this, we utilize COVID-19 case counts from March 3rd, 2020 to September 17th, 2021 as the training data. Subsequently, our model is tested on data from September 18th, 2021 to February 4th, 2022 to evaluate its predictive capabilities. Throughout the training process, we employ the Adam optimizer, as previously mentioned. In addition to optimizing the loss, we also measure the training and testing accuracy of the model when trained with COVID-19 data. The model iterates over 100 epochs, calculating the MSE error for both the training and testing datasets. After each calculation, the model is adjusted to

optimize the loss and improve performance. The Figure displays the all the errors, as well as the training and testing accuracy, for the last few epochs.

```
- MSE: 0.0477 - MAE: 0.1980 - MSLE: 0.0369 - accuracy: 0.8784 - val_loss: 0.1847 - val_MSE: 0.0373 - val_MAE: 0.1847 - val_MSLE: 0.0301 - val_accuracy: 1.0000 -
- MSE: 0.0456 - MAE: 0.1930 - MSLE: 0.0353 - accuracy: 0.8693 - val_loss: 0.1827 - val_MSE: 0.0365 - val_MAE: 0.1827 - val_MSLE: 0.0296 - val_accuracy: 1.0000 -
- MSE: 0.0459 - MAE: 0.1935 - MSLE: 0.0356 - accuracy: 0.8845 - val_loss: 0.1808 - val_MSE: 0.0358 - val_MAE: 0.1808 - val_MSLE: 0.0290 - val_accuracy: 1.0000 -
- MSE: 0.0436 - MAE: 0.1885 - MSLE: 0.0339 - accuracy: 0.8845 - val_loss: 0.1788 - val_MSE: 0.0350 - val_MAE: 0.1788 - val_MSLE: 0.0285 - val_accuracy: 1.0000 -
- MSE: 0.0437 - MAE: 0.1880 - MSLE: 0.0340 - accuracy: 0.8784 - val_loss: 0.1770 - val_MSE: 0.0343 - val_MAE: 0.1770 - val_MSLE: 0.0279 - val_accuracy: 1.0000 -
- MSE: 0.0406 - MAE: 0.1802 - MSLE: 0.0317 - accuracy: 0.8845 - val_loss: 0.1751 - val_MSE: 0.0336 - val_MAE: 0.1751 - val_MSLE: 0.0274 - val_accuracy: 1.0000 -
- MSE: 0.0416 - MAE: 0.1838 - MSLE: 0.0325 - accuracy: 0.8875 - val_loss: 0.1733 - val_MSE: 0.0329 - val_MAE: 0.1733 - val_MSLE: 0.0269 - val_accuracy: 1.0000 -
- MSE: 0.0413 - MAE: 0.1830 - MSLE: 0.0323 - accuracy: 0.8845 - val_loss: 0.1714 - val_MSE: 0.0323 - val_MAE: 0.1714 - val_MSLE: 0.0264 - val_accuracy: 1.0000 -
- MSE: 0.0389 - MAE: 0.1768 - MSLE: 0.0305 - accuracy: 0.8845 - val_loss: 0.1696 - val_MSE: 0.0316 - val_MAE: 0.1696 - val_MSLE: 0.0259 - val_accuracy: 1.0000 -
- MSE: 0.0413 - MAE: 0.1830 - MSLE: 0.0323 - accuracy: 0.8875 - val_loss: 0.1679 - val_MSE: 0.0310 - val_MAE: 0.1679 - val_MSLE: 0.0254 - val_accuracy: 1.0000 -
- MSE: 0.0385 - MAE: 0.1763 - MSLE: 0.0303 - accuracy: 0.8875 - val_loss: 0.1661 - val_MSE: 0.0304 - val_MAE: 0.1661 - val_MSLE: 0.0249 - val_accuracy: 1.0000 -
```

Fig 5.1: Model training and testing through epochs

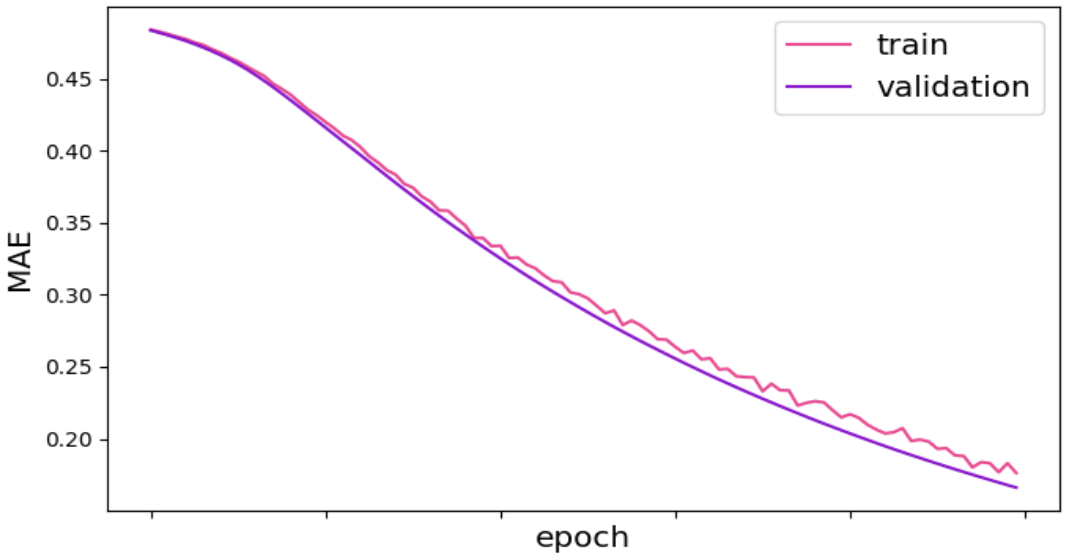


Fig 5.2: plot of Mean Absolute Error with pink line for training results and purple line for validation results

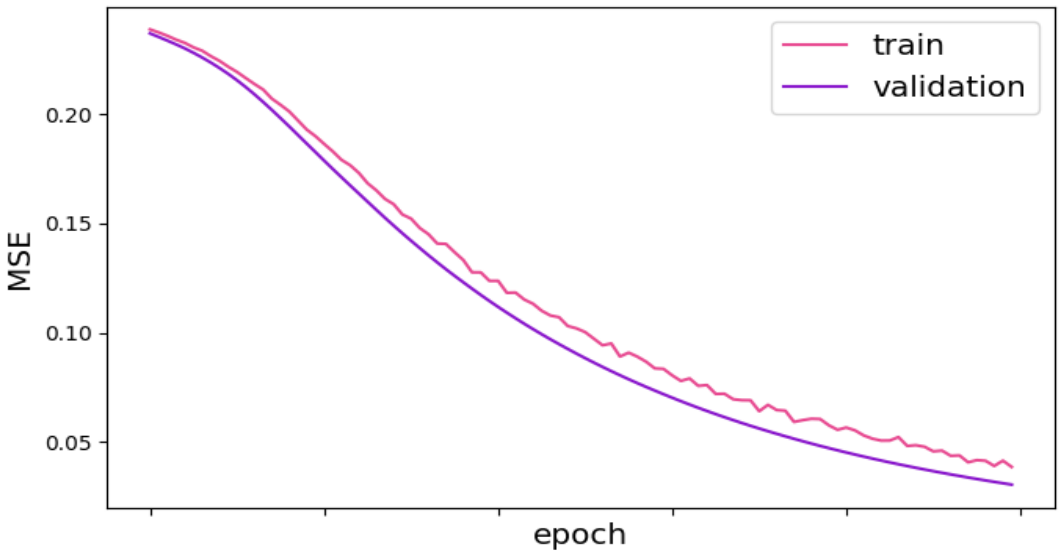


Fig 5.3: plot of Mean Squared Error with pink line for training results and purple line for validation results

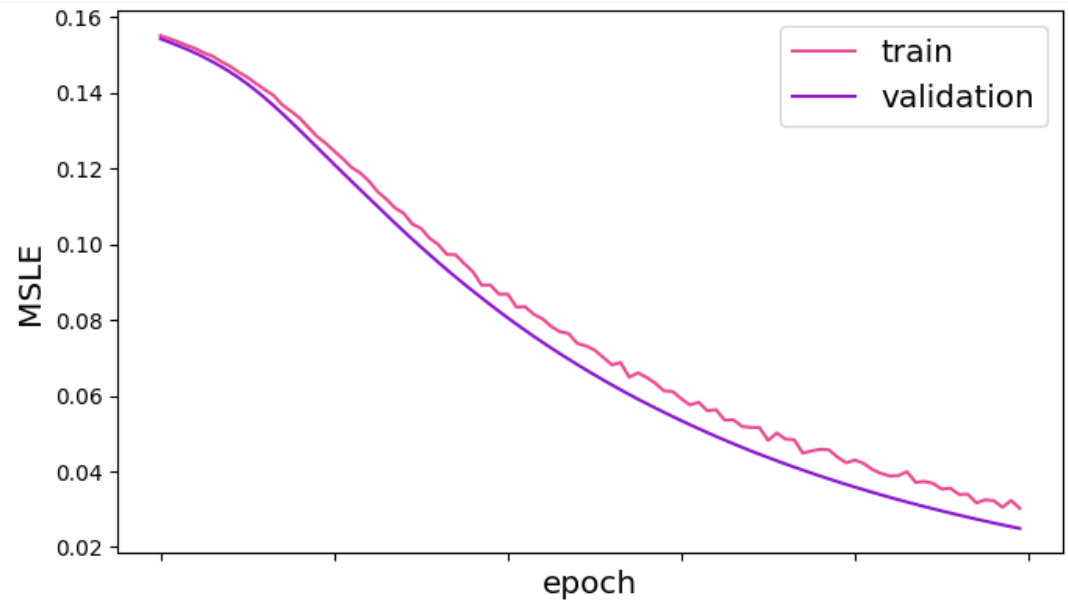


Fig 5.4: plot of Mean Squared Logarithmic Error with pink line for training results and purple line for validation results

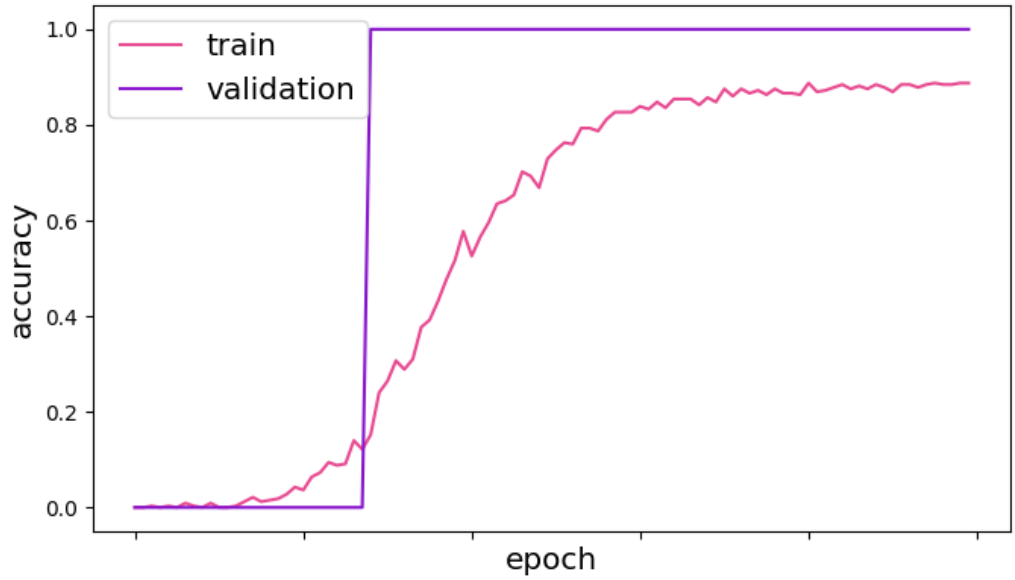


Fig 5.5: plot of accuracy with pink line for training results and purple line for validation results

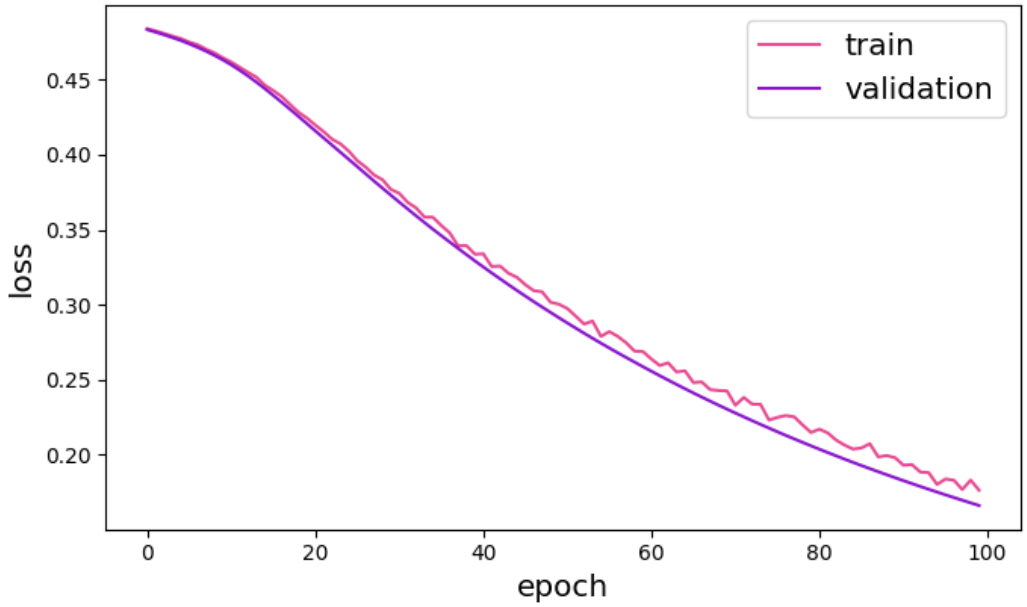


Fig 5.6: plot of Loss with pink line for training results and purple line for validation results

5.3 Evaluation Metric

As mentioned earlier the loss function or evaluation function play an importa role in the training of the model. So before going to the performance analysis it is required to take alook at the loss functions. We have used total 3 types of loss functions or error functions. All 3 are of regressor type since the problem is a classification problem.

5.3.1 RMSE (Root Mean Squared Error)

The Root Mean Squared Error (RMSE) i.e. RMSE is given by:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i' - y_i)^2}{n}}$$

Where y' represents the actual observations time series, y represents the estimated time series, and n is the number of observations. The sum of squared distance between the observed and predicted values are calculated and the sum is divided by the number of observations. Till this step this is the mean squared error (MSE) stated in Section 2.3. It then takes the square root of this sum, effectively equating the RMSE to the residuals' standard deviation.Hence, a smaller RMSE denotes improved performance and more precise forecasts.

5.3.2 MAE (Mean Absolute Error)

The absolute difference between calculated and real values is calculated using Mean Absolute Error. Because it calculates inaccuracy in observations taken on the same scale, it's also known as scale dependent accuracy. It's a statistic for evaluating regression models in machine learning. It calculates the differences between actual and model-predicted values. It is used to forecast the machine learning model's accuracy.

The Mean Absolute Error is calculated as:

$$MAE = \frac{1}{n}\Sigma|y_i - x_i|$$

Where y_i is the actual value of the i-th observation and x_i is the calculated value of i-th observation.

5.3.3 MSLE (Mean Squared Logarithmic Error)

The mean squared logarithmic error (MSLE) is a measurement of the difference between true and anticipated values. MSLE now only cares 34 about the relative difference between the true and predicted values, or in other words, the percentual difference between them. This means that MSLE will consider tiny discrepancies between true and predicted values in the same way as large disparities between true and anticipated values are treated.

$$MSLE\ (y, \hat{y}) = \frac{1}{n} \Sigma_{i=0}^{n}\ (log(y_i + 1) - log(\hat{y}_i + 1))$$

5.3.4. Values of Error Metrics

The resultant metrics of Cumulative Deaths from training and testing data shown in the following table:

Error Metric	Training data	Testing data
MSE	0.0397	0.0310
MAE	0.1796	0.1675
MSLE	0.0312	0.0245
Accuracy	88.75%	97.783%

Table 5.1: Values of Error Metrics

5.4 Visualization

In the table below, we compare the actual and predicted numerical values of Covid-19 new cases all over the world.

Date	Actual Cumulative deaths	Predicted Cumulative deaths
18/09/21	117050	12718
19/09/21	117061	12725
20/09/21	117068	12728
…….	…….	……..
…….	…….	……

The visualization of the curve fitting for the features “Cumulative Deaths” and “New Cases” respectively are shown in the next figures. The x-axis marks the timeline in terms of day numbers, and y-axis plots the features.

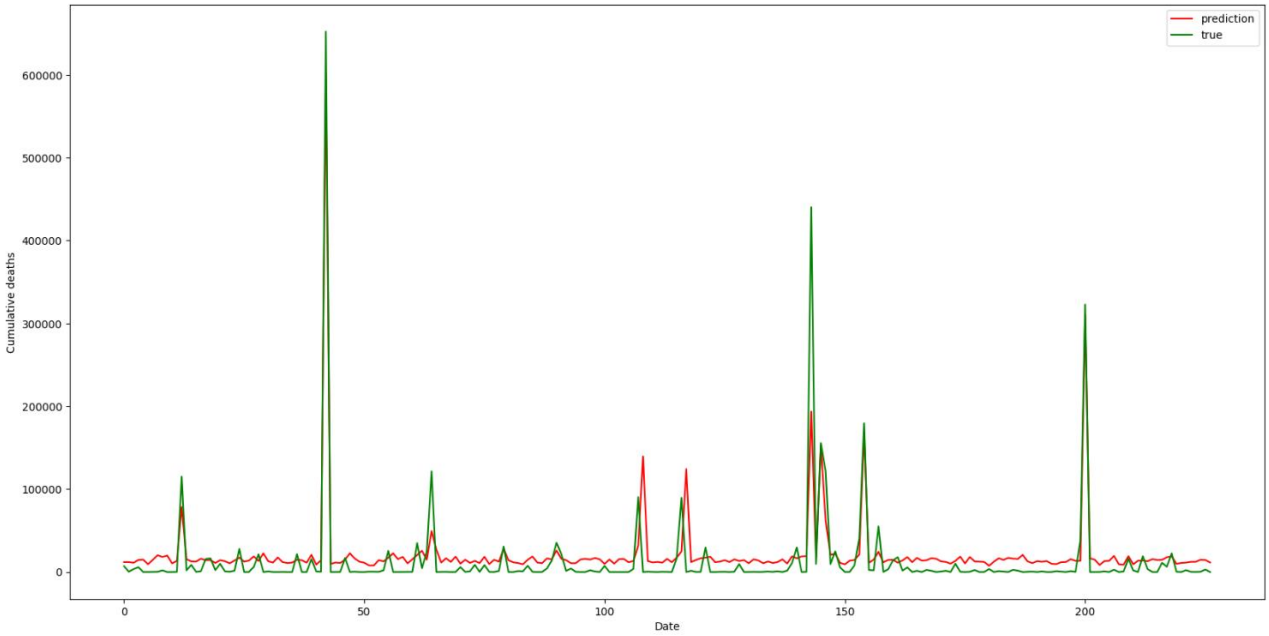


Figure 5.7: Globally Covid-19 cumulative deaths prediction, with x-axis as Dates and y-axis as Cumulative Deaths count. The green line indicates actual value and the red line indicates predicted value

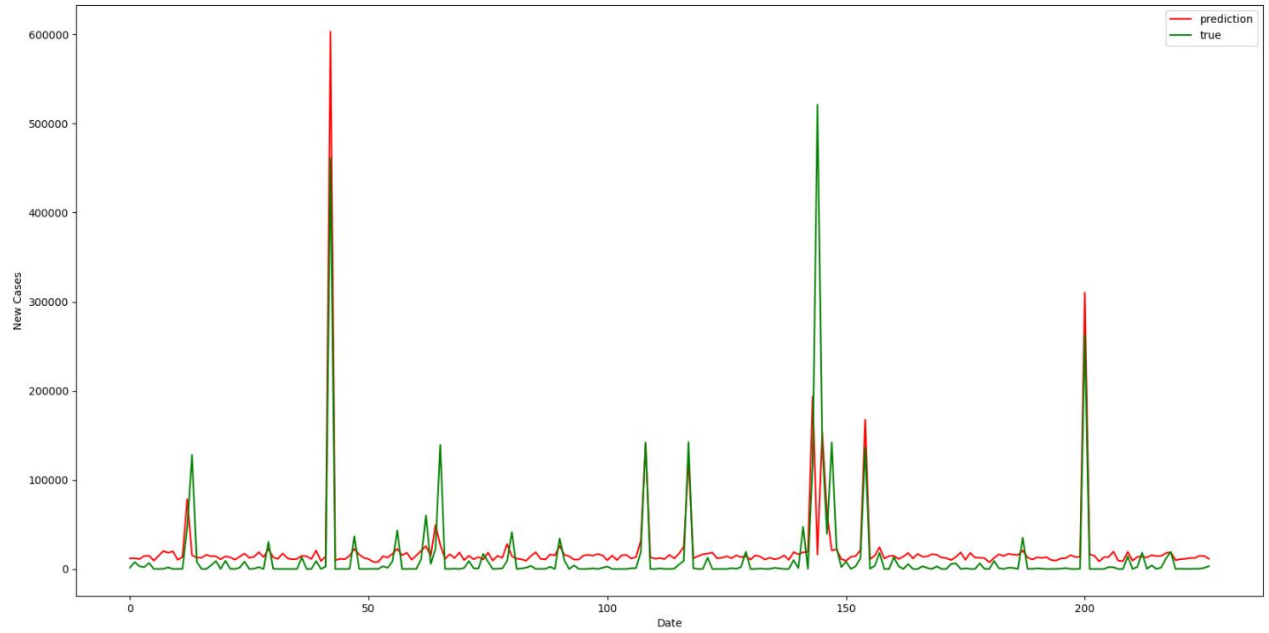


Figure 5.8: Globally Covid-19 new cases prediction, with x-axis as Dates and y-axis as New Cases count. The green line indicates actual value and the red line indicates predicted value. The calculations are done similar to the "Cumulative Deaths" feature.

Chapter 6

6.Discussion

6.1. Performance Analysis

Spatio-temporal analysis in a combined model is very uncommomn practice, although there has been lot of works in this field. The proposed model combines a time series model with a graph neural network model neural. Implementing GraphSAGE in the field of disease prediction is quite uncommon. This model extends the baseline LSTM with the graphical analysis, which turns into a quite sophisticated approach. The graph we created with 227 nodes is certainly a big size to learn and this impacts the long training time, also this graph is very complicated network to learn.

This model performs quite well in terms of forecasting. The geographical and time series analysis together adds up to the accuracy, that a baseline LSTM might not have met due lack of features. This model makes the forecast in macroscale and works on country level data. This model may be extended to work with localised data with larger data sets or appropriate adjustments. Although, in terms of performance and considering the period of time considered where so many diverse types of changes were happening in various countries, this model gives good insights of spread of covid 19 in terms of mortality and new infections.

This model lacks the availability if mobility and seasonal data. The literature reviews have shown the that incorporating mobility and seasonal data into the macroscale analysis helps a better curve fitting and reliable prediction. Although this model may not perfectly fit the analysis but the power of the combined analysis is not less. To use this model into its full potential, some other parameters like mobility data, climate changes, effect of Covid control measures or epidemiological features to generate parameters might be helpful.

6.2. Future Scope

In this paper we have studied the WHO covid 19 data in terms of Spatio-temporal analysis. We have chosen graph as primary mode of representation of our data. In our graph after all pre-processing, we got 227 countries. The countries were used as nodes in the graph with the time series data of Covid cases as node features. The edges were marking the distance. Therefore, this analysis is done based on just the geographical distances and time series

analysis. But we can't ignore the power of GraphSAGE which convoluted the features of each node with their neighbouring nodes and LSTM analysed the converted feature matrix. We introduced Mean Squared Error for the error evaluation, which helped the model to fit the dataset as the epochs proceeded.

Macro-scale time-series analysis requires to fit the seasonal changes. For different countries Covid 19 pattern is different. They show a different timeline in Covid 19 curve. The starting date, spreading of covid, peak infection rate, causes behind the spreading are different for different countries. At times of peak rate every country adopted preventive measures as much as feasible. The lockdown or curfew were implemented periodically or in stretch in almost all countries and their effect is reflected in the time series. The introduction of vaccination was also a turning point of the pandemic. Moreover, the virus have evaluated many times, some were stronger and some could be resisted. Treatments evaluated to arrest the disease. With the treatment some cases of comorbidity also araised which were mistaken by effect Covid.

Our model inherently includes these as the timeseries is the record of daily cases and intuitively, reflected these changes, sometimes such close changes are hard to analyse. Explicit inclusion of these feature might help to use the model for more localised analysis, like state or city based. Incorporating people or community-based features also will be helpful in improving the model.

The algorithms GraphSAGE and LSTM with their trivial implementations do not have a compatible interface. It is hard to incorporate feature with time series in GraphSAGE without losing the original structure. Further modification in implementation might help to stand out this model as a strong and unique model for time-series analysis

Chapter 7,8

7. Conclusion

The global impact of the COVID-19 pandemic has posed unparalleled challenges to economies and healthcare systems worldwide. This virus swiftly spread across the globe, disrupting our daily lives, healthcare infrastructure, economies, and virtually every aspect of society. Tragically, we have witnessed the loss of approximately 6.9 million human lives throughout this historic period.

Since the early days of the pandemic, researchers have endeavoured to comprehend the nature of the COVID-19 virus and its profound implications for humanity. However, due to limited data availability during the initial stages, assessing the widespread impact of this deadly virus proved challenging. Nonetheless, a significant volume of research has been dedicated to forecasting future scenarios.

Numerous machine learning models have been proposed and successfully implemented to aid in prediction efforts. In our study, we have collected comprehensive global statistics on COVID-19, encompassing daily new cases and deaths across various countries. There has been numerous research works in different aspects at all levels. Researches are still going on to estimate the severity of such pandemics at early stages.

Through this paper, we propose a deep learning model called the Long Short-Term Memory Graph Convolutional Network (GCN-LSTM). Our aim is to investigate COVID-19 case-count data from nations worldwide and gain insights into the macro-scale transmission patterns of the virus. It is important to note that the findings of this study serve as an initial exploration of the potential insights that can be derived from meso-scale time-series forecasting. With further refinements to our models, we anticipate building a foundation for evaluating and preparing for future pandemics, contributing to improved readiness and response strategies.

8. References:

1. Columbia, Mailman School of Public Health, Columbia University Irving Medical Center, 05/2023.

2. Centers for Disease Control and Prevention, Epidemiology, 05/2023.

3. UpToDate, “Covid-19: Epidemiology, Virology and Prevention”,05/2023.

4. World Health Organization. Director-General's remarks at the media briefing on 2019-nCoV on 11 February 2020. (Accessed on February 12, 2020).

5. Meyerowitz EA, Richterman A, Gandhi RT, Sax PE. Transmission of SARS-CoV-2: A Review of Viral, Host, and Environmental Factors. Ann Intern Med 2021; 174:69.

6. Morawska L, Milton DK. It Is Time to Address Airborne Transmission of Coronavirus Disease 2019 (COVID-19). Clin Infect Dis 2020; 71:2311.

7. World Health Organization. Transmission of SARS-CoV-2: Implications for infection prevention precautions. (Accessed on July 10, 2020).

8. Klompas M, Baker MA, Rhee C. Airborne Transmission of SARS-CoV-2: Theoretical Considerations and Available Evidence. JAMA 2020.

9. Chagla Z, Hota S, Khan S, et al. Re: It Is Time to Address Airborne Transmission of COVID-19. Clin Infect Dis 2021; 73:e3981.

10. Duval D, Palmer JC, Tudge I, et al. Long distance airborne transmission of SARS-CoV-2: rapid systematic review. BMJ 2022; 377:e068743.

11. Lu J, Gu J, Li K, et al. COVID-19 Outbreak Associated with Air Conditioning in Restaurant, Guangzhou, China, 2020. Emerg Infect Dis 2020; 26:1628.

12. Hamner L, Dubbel P, Capron I, et al. High SARS-CoV-2 Attack Rate Following Exposure at a Choir Practice - Skagit County, Washington, March 2020. MMWR Morb Mortal Wkly Rep 2020; 69:606.

13. Shen Y, Li C, Dong H, et al. Community Outbreak Investigation of SARS-CoV-2 Transmission Among Bus Riders in Eastern China. JAMA Intern Med 2020; 180:1665.

14. W. H. O. (WHO), “WHO Coronavirus (Covid-19) Dashboard,” 05/2023.

15. Epidemiological Model and Covid 19 – A Comparative view, national library of Medicine, Pub Med Central, 05/2023.

16. “Predicting the cumulative number of cases for the COVID-19 epidemic in China from early data” by Zhihua Liu, Pierre Magal, Ousmane Seydi, Glenn

17. C. Ian, A. Mondal and C. G. Antonopoulos, “A SIR model assumption for the spread of COVID-19 in different communities.,” Chaos, Solitons & Fractals, vol. 139, p. 110057, 2020.

18. N. Bannur, H. Maheshwari, S. Jain, S. Shetty, M. Srujana and A. Raval, "Adaptive COVID-19 forecasting via Bayesian optimization", *Proc. 8th ACM IKDD CODS 26th COMAD*, pp. 432-432, Jan. 2021.

19. H. Gupta, S. Kumar, D. Yadav, O. P. Verma, T. K. Sharma, C. W. Ahn, et al., "Data analytics and mathematical modeling for simulating the dynamics of COVID-19 epidemic—A case study of India", *Electronics*, vol. 10, no. 2, pp. 21, Jan. 2021.

20. S. He, Y. Peng and K. Sun, "SEIR modeling of the COVID-19 and its dynamics" by *Nonlinear Dyn.*, vol. 101, no. 3, pp. 1667-1680, Jun. 2020.

21. F. S. Lobato, G. B. Libotte and G. M. Platt, "Identification of an epidemiological model to simulate the COVID-19 epidemic using robust multiobjective optimization and stochastic fractal search", *Comput. Math. Methods Med.*, vol. 2020, Oct. 2020.

22. L. Russo, C. Anastassopoulou, A. Tsakris, G. N. Bifulco, E. F. Campana, G. Toraldo, et al., "Tracing day-zero and forecasting the COVID-19 outbreak in lombardy italy: A compartmental modelling and numerical optimization approach", *PLoS ONE*, vol. 15, no. 10, Oct. 2020.

23. S. Ghamizi, R. Rwemalika, L. Veiber, M. Cordy, T. F. Bissyande, M. Papadakis, et al., "Data-driven simulation and optimization for Covid-19 exit strategies" at *arXiv:2006.07087*, 2020,.

24. L. Guan, C. Prieur, L. Zhang, C. Prieur, D. Georges and P. Bellemain, "Transport effect of COVID-19 pandemic in France", *Annu. Rev. Control*, vol. 50, pp. 394-408, 2020.

25. F. Martínez-Álvarez, G. Asencio-Cortés, J. F. Torres, D. Gutiérrez-Avilés, L. Melgar-García, R. Pérez-Chacón, et al., "Coronavirus optimization algorithm: A bioinspired Metaheuristic based on the COVID-19 propagation model", *Big Data*, vol. 8, no. 4, pp. 308-322, Aug. 2020.

26. A. Behnood, E. Mohammadi Golafshani and S. M. Hosseini, "Determinants of the infection rate of the COVID-19 in the U.S. Using ANFIS and virus optimization algorithm (VOA)", *Chaos Solitons Fractals*, vol. 139, Oct. 2020.

27. A. Mokhtari, C. Mineo, J. Kriseman, P. Kremer, L. Neal and J. Larson, "A multi-method approach to modeling COVID-19 disease dynamics in the United States", *Sci. Rep.*, vol. 11, no. 1, pp. 12426, Jun. 2021.

28. T. D. Do, M. M. Gui and K. Y. Ng, "Assessing the effects of time-dependent restrictions and control actions to flatten the curve of COVID-19 in Kazakhstan", *PeerJ*, vol. 9, Feb. 2021.

29. T. Kufel, "ARIMA-based forecasting of the dynamics of confirmed Covid-19 cases for selected European countries," Equilibrium. Quarterly Journal of Economics and Economic Policy, vol. 16, no. 2, 2021.

30. Chimmula, V. K. Reddy and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks.," Chaos, Solitons & Fractals, vol. 135, no. 109864, 2020.

31. Tian, Yuan, I. Luthra and X. Zhang, "Forecasting COVID-19 cases using Machine Learning models," MedRxiv, 2020.

32. Zeroual, Abdelhafid, F. Harrou, A. Dairi and Y. Sun, "Deep learning methods forforecasting COVID-19 time-Series data: A Comparative study.," Chaos, Solitons & Fractals, vol. 140, no. 110121, 2020.

33. Mohammadhossein Toutiaee, Xiaochuan Li, Yogesh Chaudhari, Shophine Sivaraja, Aishwarya Venkataraj, Indrajeet Javeri, Yuan Ke, Ismailcem Arpinar, Nicole Lazar, John Miller ·" Improving COVID-19 Forecasting using eXogenous Variables"20 Jul 2021.

34. "Spatio-temporal prediction of the COVID-19 pandemic in US counties: modeling with a deep LSTM neural network" by Behnam Nikparvar, Md. Mokhlesur Rahman, Faizeh Hatami & Jean-Claude Thill

35. Xianlin Wang, Yuqing Liu, Haohui Xin, "Bond strength prediction of concrete-encased steel structures using hybrid machine learning method". Fig, 05/2023

36. Understanding Graph Convolutional Networks for Node Classification,Inneke Mayachita, Towards Data Science.

37. Book "Deep Learning for Robot Perception and Cognition",edited by: Alexandros Iosifidis and Anastasios Tefas, Chapter 4, contributed by Negar Heidari, Lukas Hedegaard and Alexandros Iosifidis

38. Graph Neural Networks: A Deep Neural Network for Graphs - Explore different Graph Neural Networks (GNN): GCN, GraphSAGE, and GAT by Renu Khandelwal, 05/2023

39. Inductive Representation Learning on Large Graphs by William L. Hamilton, Rex Ying, Jure Leskovec.

40. T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In ICLR, 2016

41. Time Series Analysis and Forecasting | Data-Driven Insights (Updated 2023), by Shanthababu Pandian — Published On October 23, 2021 and Last Modified On April 26th, 2023. Accessed on 05/2023.

42. Page 18-19, Practical Time Series Forecasting with R: A Hands-On Guide., by by Galit Shmueli (Author), Kenneth C. Lichtendahl Jr (Author)

43. S. Hochreiter and J. Schmidhuber, "LONG SHORT-TERM MEMORY," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.

44. Deep Learning Specialization on Coursera,Master Deep Learning, and Break into AI,Instructor: Andrew Ng, [images LSTM.png, LSTM-rnn.png, 05/2023].

45. Can ReLU Cause Exploding Gradients if Applied to Solve Vanishing Gradients?The exploding and disappearing gradient problems are the issues that arise when using gradient-based learning methods and backpropagation to train artificial neural networks.By Vijaysinh Lendave, published on November 25, 2021 . 05/2023

46. Published on January 11, 2018, Types of Activation Functions In Neural Networks And Rationale Behind It, by Kishan Maladkar. 05/2023

47. Various Optimization Algorithms For Training Neural Network -The right optimization algorithm can reduce training time exponentially, by Sanket Doshi,Towards Data Science ,05/2023

48. Latitude and Longitude for Every Country and State - GPS coordinates for every world country and every USA state, Paul Mooney, 05/2023

49. Activation Functions: Sigmoid vs Tanh, written by: Panagiotis Antoniadis, 05/2023