Predicting COVID-19 Variant Probabilities Using Deep Learning Techniques

Group 11 Project 1

**Group Members:**

Jennifer Becerra

Sierra Landacre

Don Krapohl

**Date:** February 6, 2025

**Google Colab:** [IDC6146\_Group11\_project1.ipynb](https://colab.research.google.com/drive/1KKUghigwDqxqXZACEB-Df01ntc6tI6JH?usp=sharing)

**Data**: [project1\_data](https://drive.google.com/drive/folders/1f0-OJ3_k1EoNrnKj-M4Z0FN4XC3dTz-O?usp=sharing)

# Objective:

### The purpose of this project is to predict the probability of the five most widespread COVID-19 variants in each country based on temporal, societal, and infrastructure information.

# Data Collection:

**Sources:** Our project focused on the data collected from World Factbook by CIA and the SARS-CoV-2-Variants repository.

# Abstract

This research explores the development and training of Feedforward Neural Networks (FNN) and Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) layers to predict the highest detected COVID-19 variants in different countries. The study focuses on data preprocessing, model architecture, training, and evaluation to understand the models’ performances and their implications for public health.

# Introduction

The COVID-19 pandemic has highlighted the need for effective prediction models to understand and manage the spread of different variants. This study leverages FNN and RNN models with LSTM layers to predict the prevalence of COVID-19 variants based on socio-environmental factors. The objective is to develop models that can provide accurate predictions and inform public health strategies.

**Models Employed in this Study**

Time Series

The data were collected irregularly but does allow for time series analysis. Time series analysis is valuable to correct certain types of issues or recurring noise patterns in the samples that can come from sample collection frequency and quality differences, number of variants captured in testing, bias in the collectors, or any number of confounding errors. Moving averages and lag are particularly important in time series to smooth out the jagged nature of the data over time and to determine if there is some latency in time between something in the environment and changes in the response variables (here, COVID variant).

ANN

An Artificial Neural Network (ANN) is a generalized term for a single-layer analytical model wherein model inputs are weighted, the result rectified or threshholded in some way, measured against known values, and the difference (error) applied backwards through the model to update weights for a more accurate and nuanced prediction. Modeled on the architecture of animal/human brains they're also able to characterize, analyze, and capture complex relationships. They're fairly simple but they're usually not able to capture non-linear or complex relationships in data but do provide a simple starting point in modeling.

MLP

A multi-layered perceptron is another artificial neural network able to capture complexity as can the others here. It is an improvement on the basic ANN in that it is multi-layered and is able to better capture non-linear relationships. Unlike ANNs, MLPs are able to adapt their weights in a manner that can uncover hierarchical relationships in data.

Feed-forward Network

A feed forward network (FFN) in general is one in which the weights are only applied in a forward direction without backpropagation from the final error function through the activation functions and weights. It's less computationally intensive than most of the other neural network models we implemented but since it also tends to capture non-linear patterns almost as well as other model types it could be a good choice if we wanted to train this model over larger datasets, on limited compute, or at higher frequencies. In general FFNs are considered to be more straightforward to interpret than the more complex neural network architectures, likely because of the lack of arithmetic complexity that would come with backpropagation.

RNN+LSTM

Recurrent Neural Networks (RNNs) are a modeling approach wherein relationships between data and the order in which samples occur are potentially meaningful, which adds the power of context that more context-naive models like FFN might miss. In our project the data were samples over time and the order in which the values appear would be core to our ability to forecast or predict variant migration or population changes in the variants. Supplementing the contextual power of RNN is Long Short-Term Memory (LSTM), which provides the ability of previous values and how they change to affect the weights and activations of the current time step. This allows the models to retain that context (often referred to as "remember") over longer periods of time. Backpropagation of weights in RNNs can also experience exploding or vanishing gradients, requiring regularization or other means to avoid these problems. LSTMs retain more "state" and context information over time mitigating the vanishing gradient problem in RNNs.

GNN

Graph Neural Networks (GNNs) have the benefit of the features of graph structures internally allowing complex node relationships and non-obvious structures to be captured. As in typical graph algorithms they are able to perform operations over heterogeneous graph data where nodes and edges have differing attributes. Preprocessing can require embedding the graph relationships in the data, which can itself be a highly complex and time-consuming task. GNNs can be computationally complex and due to the inherent complexity in real-world graph structures GNN models can tend to overfit.

TCN

A Temporal Convolutional Network (TCN) is optimized to deal with time series data and capture long-term dependencies and relationships in data. Training parallelizes well even over complex network architectures while avoiding vanishing gradient problems. TCNs employ "causal convolutions" which serve to ensure that information flow in the model is solely from past to present preventing invalid models where the present can affect the past. Note that we had difficulties implementing this model and left the code in the notebook commented out.

Logistic Regression

For comparison to our neural networks we produced a simple Logistic Regression model. Logistic regression is typically used for binary yes/no predictions but we used it here for categorical multi-class probability prediction. Logistic regression is useful to analyze relationships between the variables.

# Recent Advancements in Disease Modeling and Pandemic Forecasting

Recent advancements in disease modeling and pandemic forecasting have significantly enhanced our ability to predict and manage the spread of infectious diseases. Two notable approaches are Transformer-based forecasting models and Graph Neural Networks (GNNs).

A broad range of approaches have been taken to understand the classification and patterns of evolution in the SARS-CoV (COVID-19) epidemic of 2019-present. Karthikeyan used principal component analysis and XGBoost to achieve 90% accuracy in predicting mortality (Karthikeyan, 2022). Likewise, Mollalo studied 57 candidate explanatory variables and determined 5 of these input into a single hidden-layer Multilayered Perceptron accurately characterized 65% of mortality risk (Mollalo & Rivera, 2020). Discretization and ensemble methods have also been used effectively to achieve more consistent results in mortality and transmission by grouping countries together that display roughly analagous patterns of viral behavior (Bird, 2020). This paper also noted that the modeling methods they used were more able to capture effects earlier in the pandemic with reduced quality after September 2020. A study of New York cases showed the intent to use model-based predictions to stage medical resources like hospital beds, however the predictions did not accurately represent the situation as it unfolded. The models broke down due in large part to issues in sampling, especially early in the pandemic (Chin & Samia, 2020) in contrast to changes in modeling in India that had the opposite trend (Chin & Ioannou, 2020).

# Transformer-based Forecasting Models

Transformer-based models, such as COVID-BERT and Temporal CNNs, have shown great promise in forecasting the spread of COVID-19. COVID-BERT leverages the Transformer architecture to capture temporal dependencies in the data, providing accurate predictions of case numbers and the impact of interventions. Temporal CNNs combine Convolutional Neural Networks with temporal data, effectively capturing spatial and temporal patterns to enhance forecasting accuracy.

**Graph Neural Networks (GNNs)**

Graph Neural Networks (GNNs) have emerged as powerful tools for modeling the spread of infectious diseases. GNNs can capture complex relationships between different regions and populations, making them ideal for understanding and predicting disease dynamics. Recent studies have demonstrated the effectiveness of GNNs in epidemic modeling, providing valuable insights into disease spread mechanisms.

# Authors and Roles

The project team as noted at the beginning of the paper is composed of:

Jennifer Becerra

Sierra Landacre

Don Krapohl

Each team member submitted ideas for the project, voted on which project we’d like to do, performed exploratory data analysis, provided 1-2 models to the notebook, measurements of model efficacy, and interpreted the models for quality. Jennifer authored a large part of the paper while Sierra and Don each contributed at least 1 section. Don performed early phases of data shaping and all contributed to selecting features and preprocessing. Jennifer and Sierra did the majority of the visualizations. The conclusions, applications of the model, and future work were an equal collaboration among all team members.

**Data Preprocessing**

The data preprocessing phase of this project involved several meticulous steps to ensure the dataset was clean, structured, and ready for analysis. This process started with combining country data with COVID-19 variant data, followed by extensive data cleaning and transformation procedures. First, we loaded the country data file, which contained various attributes such as urban population, food insecurity, hospital bed density, and obesity rate. Simultaneously, the COVID-19 data files named by country, which contained information on COVID-19 variants, were loaded. We then matched each row of the country data with the corresponding COVID-19 data files, appending the country attributes to the COVID-19 data for each matched country, and saved these augmented data files to an intermediate directory. Finally, all the augmented COVID-19 data files were combined into a single DataFrame and saved as a pickled file to Google Drive for subsequent use.

To handle empty data files and ensure the process was robust, try blocks were used to catch exceptions. This approach ensured continuity by skipping over any missing data files without causing interruptions. Addressing data type mismatches was crucial; we ensured that numeric columns were correctly converted to resolve comparison issues between strings and integers. The merging process integrated the socio-environmental factors with COVID-19 variant data based on common columns, resulting in a cohesive dataset ready for further analysis.

Categorical features such as 'Country' and 'location' were transformed into numerical values using label encoding with LabelEncoder. Numerical features, including 'UrbanPop', 'HospBedDensity', and 'Obesity', were cleaned by extracting numeric values and converting them to float. The 'date' column was also scaled using MinMaxScaler. To maintain data integrity, any NaNs (missing values) were filled with 0.0 using the fillna(0.0) method. This ensured that the DataFrame was well-structured and free of missing values, preventing issues during analysis or modeling.

Figure 1. COVID Variants Through Time in the United States, 2020-2023

A graph showing the number of covid-19

AI-generated content may be incorrect.We also ensured that the column names were user-friendly and descriptive by updating them using the set\_axis method. Each column was converted to its specified data type using the astype method to ensure accurate data representation. The numerical data contained in the COVID-19 dataset, which used commas as decimal separators, was standardized by replacing commas with periods using the str.replace method. These string values were then converted to floating-point numbers using the astype(float) method, ensuring the numerical data was correctly formatted. Columns with mixed data types were identified and standardized as strings to maintain consistency. By iterating over designated string columns, mixed-type columns were converted to the string data type using the astype(str) method. These steps ensure data consistency and integrity, making the DataFrame well-prepared for accurate and reliable further processing. The comprehensive preprocessing and cleanup steps laid a solid foundation for subsequent analysis and model training.

The data consisted of ~11,000 samples after cleanup and consolidation. The class distribution of the COVID variants were as in Figure 2:

A screenshot of a phone

AI-generated content may be incorrect.

Figure 2. Class Representation by Variant

Within each of these variants were many sub-variants that were parsed and mapped to their base Greek-lettered variant type or extracted through regular expression where possible. For example, “A1.2.3”, “Alpha”, “A (WHO)” would all map to the Greek-lettered variant “Alpha”. Some variant types, such as variant “K” were so rare as to be irrelevant to the study. Class inverse weighting was calculated to allow weight inputs into models that accept them, however none of the models under study here would accept class weights except Logistic Regression. There were significant problems with the quality of the data with some countries providing either no data at all or data sufficiently poor as to be unusable, which was handled by synthesizing data for missing dates as detailed below.

Categorical information from the CIA Factbook was selected manually from the roughly 1100 attributes provided. This approach was taken as some factors made no sense within the problem we were solving, some features were empty or mostly bad data, or some features were sufficiently sparse as they could not conceivably be useful in a meaningful prediction in a real-world application. If further time and resources were available in a follow-on study it would be valuable to expand the feature set to a hundred or so columns with valid data and do feature reduction despite the appearance that the column could have no predictive value.

The data represented a time series dataset with measures of COVID variants in the population collected through time. The data were not consistently collected so to provide a cleaner dataset a data row was synthesized for each missing date for each country. The features in each row generated was copied from the previous row actually collected (not synthesized) allowing monotonically increasing dates to be handled in order with no gaps. Time series methods of data smoothing were employed, specifically Exponential Moving Average smoothing but this led to lower scores in the models. Likewise interpolation between dates for missing samples (instead of simply “filling down” from the last known sample) did not yield improvements in the models.

# Methodology:

**Building the FNN Model**

The methodology for building the Feedforward Neural Network (FNN) to predict COVID-19 variants encompassed several key stages: data preprocessing, model architecture design, training, and evaluation. Each step was meticulously crafted to ensure the model's robustness and accuracy. An FNN model is ideal in that it is relatively computationally efficient, handles non-linear relationships well, and scales well if we decided to take the model to larger datasets or real-time applications.

**Model Architecture of FNN**

The design of the FNN was implemented using TensorFlow and Keras. The model was structured as a Sequential model, starting with an input layer that matched the number of features after preprocessing. Multiple dense layers with ReLU and LeakyReLU activations were used, each followed by BatchNormalization to stabilize training and improve convergence. Dropout layers were added to prevent overfitting by randomly setting a fraction of input units to zero during training. The output layer utilized a softmax activation to predict the probability distribution over the target classes (COVID-19 variants). This architecture provided a robust foundation for the neural network, enabling it to learn complex patterns in the data.

**Training and Early Stopping of FNN**

The model was compiled using the Adam optimizer and categorical cross-entropy loss function. Training involved implementing an early stopping mechanism to monitor the validation loss. This mechanism halted training if there was no improvement in the validation loss after 8 epochs, restoring the best model weights to avoid overfitting. The model was trained for up to 200 epochs, with an 80-20 train-validation split and a batch size of 128. This approach ensured that the model was trained efficiently while preventing overfitting.

**Evaluation of the FNN Model**

**Model Accuracy**

The training and validation accuracy values are visualized to understand how well the model is learning. The training accuracy plot shows the model’s performance on the training data over the epochs, ideally steadily increasing as the model learns. The validation accuracy plot represents the model’s performance on the validation data. An increasing trend in line with the training accuracy indicates good generalization.

The confusion matrix for the FFN model indicated a profound tendency to predict most input as class 3 with no predictions of anything belonging to classes 0 or 2 and very few for classes 1 and 4. This, of course, undermines the efficacy of the model as it’s not possible to discern true positive vs false positive when class 3 is predicted.

The precision for this model was 13.6% and recall 21%. This is a multi-class problem with less-than-ideal data so it was anticipated that the power of the model would be somewhat low, but these values indicate that the model was largely wrong in its predictions and shouldn’t be used for anything that could affect peoples’ health.

**Model Loss**

The training and validation loss values are visualized to understand how the error on the training and validation data changes over epochs. The training loss plot shows how the error on the training data decreases over the epochs, indicating effective learning. The validation loss plot shows how the error on the validation data changes over the epochs. A decrease similar to the training loss indicates good model performance without overfitting.

**Evaluation of the Visualizations for the FNN Model**

The evaluation of the visualizations for the FNN model reveals several insights. The training and validation accuracy curves should ideally increase and stabilize at high values, indicating good model performance and generalization. The training and validation loss curves should ideally decrease and stabilize at low values, indicating effective learning. A significant gap between the training and validation accuracy (or loss) suggests overfitting, which means the model may not perform well on unseen data.

**Predictions of FNN Model**

The FNN model was built to predict which COVID-19 variants would be most common in different countries, using information about each country. By using a neural network with several layers, we could get probabilities for each variant, showing how likely it was to appear in each country. We used techniques like label encoding and data normalization to make sure the model understood the data correctly. Regularization methods like L1 or L2 were used to prevent the model from overfitting, which means it wouldn't perform well on new, unseen data. As we fine-tuned the model and added new techniques, its predictions became more accurate and reliable.

**Implications of FNN Model**

The predictions from this model could be very useful for public health and policymaking. By knowing which COVID-19 variants were likely to spread in different regions, public health officials and policymakers could prepare better. They could allocate resources more effectively, like increasing testing or vaccination efforts in areas where a particular variant is expected to become dominant. The insights from the model could also help in ongoing research and surveillance, enabling a more proactive approach to managing the pandemic. By making informed decisions based on these predictions, policymakers could enhance public health outcomes and save lives.

**Building the RNN Model with LSTM**

The Recurrent Neural Network (RNN) model with Long Short-Term Memory (LSTM) layers was built using TensorFlow and Keras. The architecture began with an LSTM layer containing 128 units, designed to return sequences for the subsequent LSTM layer. Following this, a dropout layer with a 50% rate was added to prevent overfitting by randomly setting half of the input units to zero during training. The next component was another LSTM layer, this time with 64 units, which produced the final representation. The architecture concluded with two dense layers; the first dense layer had 32 units and utilized the ReLU activation function to introduce non-linearity into the model, while the second dense layer, serving as the output layer, used the softmax activation function for multi-class classification, with the number of units equal to the unique classes in the target variable.

**Training and Early Stopping of RNN Model with LSTM**

The model was compiled using the Adam optimizer, known for its efficiency in handling sparse gradients, and the categorical cross-entropy loss function, suitable for multi-class classification tasks. During the training process, early stopping was implemented to monitor the validation loss. If no improvement in validation loss was observed for ten consecutive epochs, training would halt, and the best model weights would be restored to prevent overfitting. Additionally, learning rate reduction was applied, reducing the learning rate by a factor of 0.2 if the validation loss plateaued for five epochs, ensuring better convergence and preventing the model from getting stuck in local minima. The model was trained for up to 50 epochs with a batch size of 32, and an 80-20 train-validation split was used to evaluate the model's performance on unseen data.

**Evaluation of the RNN Model with LSTM**

**Model Accuracy**

**A screenshot of a graph

AI-generated content may be incorrect.**The training and validation accuracy values are visualized to understand how well the model is learning and generalizing. The training accuracy plot shows the model's performance on the training data over the epochs, ideally steadily increasing as the model learns. The validation accuracy plot represents the model’s performance on the validation data. Ideally, the validation accuracy should increase along with the training accuracy and should not deviate significantly, indicating good generalization without overfitting.

Figure 3. RNN+LSTM Model Metrics

**Model Loss**

The training and validation loss values are visualized to understand how the error on the training and validation data changes over epochs. The training loss plot shows how the error on the training data decreases over the epochs, indicating effective learning. The validation loss plot shows how the error on the validation data changes over the epochs. Ideally, the validation loss should decrease similarly to the training loss. If the validation loss increases while the training loss decreases, it could indicate overfitting.

**Evaluation of the Visualizations for the RNN Model with LSTM**

The evaluation of the visualizations for the RNN model with LSTM layers reveals several insights. Both the training and validation accuracy curves should ideally increase and stabilize at high values, indicating good model performance and generalization. The training and validation loss curves should ideally decrease and stabilize at low values, indicating effective learning. A significant gap between the training and validation accuracy (or loss) suggests overfitting, where the model performs well on training data but poorly on unseen data.

**Predictions of RNN Model with LSTM**

The RNN model with LSTM layers provided a probability distribution for each COVID-19 variant, indicating the likelihood of its occurrence in different countries. These predictions can be instrumental in informing public health strategies. By understanding which variants are likely to spread in specific regions, public health officials can implement targeted interventions and allocate resources more effectively. For example, regions predicted to experience a higher prevalence of a particular variant could ramp up testing, vaccination, and other preventive measures to mitigate the spread.

**Implications of RNN Model with LSTM**

The predictions from this model could be very useful for shaping public policy, strategic health resource allocation, and countermeasure planning. By determining which COVID-19 variants were likely to spread in different regions, public health officials and policymakers could prepare better. By analyzing variant trends and predicting which COVID-19 variants were likely to spread in different regions, governments and health organizations can anticipate outbreaks and adjust public health guidelines accordingly. They could allocate resources more effectively and proactively including allocation of vaccines, antiviral medications, and medical supplies to regions at higher risk, reducing strain on healthcare systems. The insights from the model could also help in ongoing research and surveillance, enabling a more proactive approach to managing the pandemic. By making informed decisions based on these predictions, policymakers could enhance public health outcomes and save lives.

At the local level our models may also play a critical role in acute and emergency care planning by helping hospitals and emergency response teams prepare for surges in severe cases. With early warning of an emerging variant's transmissibility and severity, healthcare facilities can optimize ICU capacity, stockpile necessary treatments, and ensure adequate staffing levels.

Specifically, our most accurate model was a fairly basic model with only two small hidden layers, and was able to predict the most prominent SARS-CoV variant in a country based on attributes of the country and dates. The implications of this are that specific nations could be able to use their distribution networks effectively to distribute the appropriate antivirals to combat the most prominent viral strain ahead of its peak in the population. Further, the predictive nature of the time series built into the model will allow vaccine distribution and inoculation, which is required before the viral strain becomes widespread.

Confounding this model is that the use of the model would modify the spread of disease and a continuous data collection and retraining would be required to determine efficacy of predictions, or more importantly, when the reliability of the predictions falls below a reasonable threshold and the cost of marshalling infrastructure outweighs the health and economic value of the model.

# Building the ANN Model

Building the Artificial Neural Network (ANN) was built using tensorflow utilizing tensorflow keras models and layers. Our model is sequential with the input layer with 64 neurons, one hidden layer with 32 neurons and our final layer with 16 neurons. The output layer we use is softmax due to our data not being binary.

**Compiling and Training ANN**

After the model was built it was then compiled with categorical cross entropy since it is not dealing with all binary classifications. The compiled model also uses the adam optimizer. The model was trained for 50 epochs, batch size set to 32 and a validation split of 0.2. The validation split was used in order to evaluate the model’s performance of unseen data.

**Evaluation and Output**

**A screenshot of a computer

AI-generated content may be incorrect.**The ANN model was evaluated for its accuracy and loss. It was also visualized with a confusion matrix. The training accuracy shows the model’s performance of the training data over the epochs which increases as the model learns. The validation accuracy should also increase along with the training accuracy to represent the performance of the validation data. The loss for both training and validation help us to understand the error of the data and how it changes over epochs.

Figure 4. ANN Model Metrics and Confusion Matrix

**Comparison and Summary of ANN**

### A basic logistic regression was done on the ANN for comparison. In order to do this sklearn was implemented to use logistic regression and one vs rest classifier. A model was created for the logistic regression based on the one vs rest classifier. That model was then trained and evaluated. One vs the rest does not expose the model.evaluate so accuracy is used on the predictions. The summary provides information about the different types of losses and accuracies from the different NN tested.

# Impact & Citation Analysis

# Real-World and Scientific Implications

The models we developed in this study can have a big impact on the real world and science. By predicting which COVID-19 variants will be common, public health officials, policymakers, and healthcare providers can get useful insights.

# Comparing with Existing COVID-19 Forecasting Tools

To see how our models stack up, we should compare them with other COVID-19 forecasting tools. For example, the COVID-19 Forecast Hub and models from places like the IHME (Institute for Health Metrics and Evaluation) can serve as benchmarks. This comparison will highlight the strengths and weaknesses of our approach.

# Integration into Public Health Dashboards

Our models can be added to public health dashboards, like those used by the CDC (Centers for Disease Control and Prevention) and WHO (World Health Organization). This will help provide real-time predictions and visualizations of variant prevalence, supporting better decision-making and proactive measures. By integrating our model outputs into these systems, we can enable timely and accurate responses to emerging threats.

# Potential Uses for Policymakers and Healthcare Providers

The insights from our models can be useful for policymakers and healthcare providers. Policymakers can use them to allocate resources, decide on vaccination strategies, and set public health guidelines. Healthcare providers can anticipate surges in variant-specific cases, optimize hospital resources, and plan for shortages of medical supplies and staff. This way, everyone can make data-driven decisions to deal with COVID-19 effectively.

The use of the highest quality models in our project could be used to create forecasts for evolution of viral strains if they follow the same patterns of spread and mortality as COVID-19. The models would be monitored for model drift as well as retrained regularly against data to ensure high-quality predictions. The models could effectively be placed behind maps and dashboards to simulate spread or to project anticipated viral trends as well as mortality if those attributes were also added (Chin & Ioannou, 2020). These dashboards could be supplemented with modeling of viral spread based on social isolation recommendations and practices and how they can affect the predictions from spread and mortality models (Tiwari, 2020). Travel restrictions and viral case monitoring policies of countries would likewise affect the forecasting and of viral variant changes and should be added to a more robust dashboard that could show the potential error envelope of predictions based on these policy decisions (Yang, 2020).

# Broader Scientific Impact

This study adds to our understanding of infectious disease modeling and deep learning. Our methods and findings can be applied to other infectious diseases and public health challenges, encouraging more research and innovation in this area.

Post-pandemic researchers have identified a portion of the population that experiences long-term affects of the COVID-19 infestion (“long COVID”). The techniques we employ here were used at national scales to great affect during the pandemic and are now being harnessed to understand long COVID and other long-term impacts of disease (Sarmiento, 2023). Computing power is allowing more advanced algorithms and analysis of higher-dimension relationships across all diseases.

# Conclusion:

This study highlights the development and training of multiple neural networks such as FNN and RNN models with LSTM layers to predict the prevalence of COVID-19 variants through time. Through meticulous data preprocessing, model architecture design, and evaluation, these models provide valuable predictions that can inform public health strategies and policy making. The insights gained from these models allow for better preparation and resource allocation, ultimately enhancing public health outcomes and saving lives. Future research should continue to refine these models, addressing challenges such as class imbalance and incorporating additional features to improve accuracy and reliability.

# Citations

Bird, J. J., Barnes, C. M., Premebida, C., Ekárt, A., & Faria, D. R. (2020). Country-level pandemic risk and preparedness classification based on COVID-19 data: A machine learning approach. PLoS ONE, 15(10), e0241332.

Chin, V., Ioannou, B., Marchant, R., Samia, N. I., & Colijn, C. (2020). CoronaTracker: worldwide COVID-19 outbreak data analysis and prediction. Bulletin of the World Health Organization, 1-32.

Chin, V., Samia, N. I., Marchant, R., Rosen, O., Ioannou, B., Britton, T., ... & Colijn, C. (2020). A case study in model failure? COVID-19 daily deaths and ICU bed utilisation predictions in New York State. medRxiv.

Franceschini, L. (2024, April). The World Factbook by CIA (Version 4). Kaggle. Retrieved January 29, 2025, from https://www.kaggle.com/datasets/lucafrance/the-world-factbook-by-cia

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. International Conference on Machine Learning (ICML). Retrieved from https://arxiv.org/abs/1506.02142

Giordano, D. (2022). SARS-CoV-2-Variants [GitHub repository]. GitHub. Retrieved from https://github.com/3dgiordano/SARS-CoV-2-Variants

Karthikeyan, A., Garg, A., Vinod, P. K., & Priyakumar, U. D. (2022). Generalizable prediction of COVID-19 mortality on worldwide patient data. JAMIA Open, 5(2), ooac036.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR). Retrieved from https://arxiv.org/abs/1609.02907

Mollalo, A., Rivera, K. M., & Vahedi, B. (2020). Artificial Neural Network Modeling of Novel Coronavirus (COVID-19) Incidence Rates across the Continental United States. International Journal of Environmental Research and Public Health, 17(12), 4204.

Neal, R. M. (2012). Bayesian learning for neural networks. Springer Science & Business Media.

Sarmiento Varón L, González-Puelma J, Medina-Ortiz D, Aldridge J, Alvarez-Saravia D, Uribe-Paredes R, Navarrete MA. The role of machine learning in health policies during the COVID-19 pandemic and in long COVID management. Front Public Health. 2023 Apr 11;11:1140353. doi: 10.3389/fpubh.2023.1140353. PMID: 37113165; PMCID: PMC10126380.

Tiwari, S., Kumar, S., & Guleria, K. (2020). A mathematical modelling approach in the spread of the novel 2019 coronavirus SARS-CoV-2 (COVID-19) pandemic. Electronic Journal of General Medicine, 17(4), em205.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. Retrieved from https://papers.nips.cc/paper\_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wu, C., Xiao, L., Chen, B., & others. (2020). COVID-BERT: A pre-trained language model for COVID-19. arXiv preprint arXiv:2004.03188. Retrieved from https://arxiv.org/abs/2004.03188

Yang, Z., Zeng, Z., Wang, K., Wong, S. S., Liang, W., Zanin, M., ... & He, J. (2020). Global prediction system for COVID-19 pandemic. Science Bulletin, 65(22), 1884-1887.

Yang, Z., Zeng, Z., Wang, K., Wong, S. S., Liang, W., Zanin, M., ... & He, J. (2020). Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. Journal of Thoracic Disease, 12(3), 165.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., ... & Sun, M. (2020). Graph neural networks: A review of methods and applications. arXiv preprint arXiv:1812.08434. Retrieved from https://arxiv.org/abs/1812.08434

# Collected Sources for the Project (zipped)

* **Collected sources for the project (zipped):**

[**https://drive.google.com/file/d/1lurxZhdy2pjSKvolRqpOlw4Gysz3SQyP/view?usp=sharing**](https://drive.google.com/file/d/1lurxZhdy2pjSKvolRqpOlw4Gysz3SQyP/view?usp=sharing)