Application of SIRD Model and Physics Informed Neural Networks to Model COVID-19 Disease Dynamics

Aleya Hadenfeldt, Paige Keller

December 18 2025

Abstract

Infectious diseases such as COVID-19 present significant global health challenges, making accurate modeling of disease dynamics essential for understanding transmission and informing public health decisions. Traditional mathematical models, such as the SIRD, predict different dynamics of disease by tracking the number of individuals moving between four distinct groups and using differential equations. In this study, we compare constant-parameter and time-varying SIRD models with neural network–based approaches using publicly available COVID-19 data. A time-varying SIRD model is implemented as a flexible baseline capable of adapting to changes in disease behavior over time. In comparison, Physics-Informed Neural Networks (PINNs) are developed to integrate data-driven learning with the governing SIRD differential equations. While a multi-output PINN exhibited instability and poor predictive performance for certain compartments, training separate PINNs for individual compartments substantially improved accuracy. Overall, the time-varying SIRD model outperformed the PINN-based approaches, highlighting both the strength of classical models with adaptive parameters and the sensitivity of PINNs to data preprocessing and scaling choices. These results emphasize the importance of incorporating time-related variability in epidemic modeling.

1 Background

The **SIR** model is a mathematical framework used in epidemiology to describe and predict how an infectious disease spreads throughout a population over time. It divides the population into three groups: susceptible, infected, and recovered. The model assumes a fixed population and is represented as $N = S + I + R$. The SIR model is governed by a system of ordinary differential equations (ODEs) that describe how individuals move between groups. The 2 parameters are β, the transmission rate, and γ, the recovery rate.

The **SIRD** model adds a fourth group called Deceased (D) and a third parameter μ which represents disease induced mortality rate. The population is then represented by $N = S + I + R + D$.

A **Neural Network** is a computational model that learns to approximate complex relationships in data by adjusting internal parameters (weights and biases) to minimize a loss function, which measures the difference between its predictions and the true values. A **Physics Informed Neural Network (PINN)** is a type of neural network that incorporates physical laws described by differential equations into its loss function to guide the learning process toward solutions that are more consistent with the underlying physics.

Imputation is a common way to handle data when it contains missing values. The application of a dataset when it is not complete can be limiting. Thus, there are various methods to approximate these missing values in a dataset so that the values can be better understood in its application. **K-Nearest-Neighbors (KNN)** is a data imputation method that fills in missing entries using information from the most similar rows in the data. **Iterative Imputation** is another method that estimates missing values by leveraging the relationships among features. Each incomplete feature is treated as a response variable, predicted from the remaining features using a regression estimator. Missing entries are updated iteratively over multiple cycles, allowing the imputer to refine its estimates and reflect the underlying joint distribution of the dataset.

$\mathbf{R^2}$ is a metric, commonly used in regression, that measures how well a model fits observed data. An R^2 of 1 indicates a perfect fit between the model predictions and the data. In addition, **Mean Absolute Percentage Error (MAPE)** measures prediction error by expressing it as a percentage.

2 Materials and Methods

Jupyter Notebook and Python were used for data processing and implementation of the PINN. Data was initially obtained from the Our World in Data COVID-19 public repository; however, due to the lack of detailed epidemiological features, the Google COVID-19 Repository dataset was used instead. The data was downloaded with each feature in a separate dataset and on a different scale. Thus, data preprocessing was required to combine and normalize the datasets. It was noticed that the datasets started and ended on difference dates. Because a Neural Network cannot handle missing values, rows were dropped after combining so that each day in the dataset contained a value for every feature. This slightly decreased the number of rows (days) from 700 to 523.

With the rows of missing data being removed, there might not be enough data to get an accurate and generalizable working model. Thus, it was necessary to try data imputation and see how it affected the models performance. First, KNN imputation was applied to the dataset. This gave the model about 1000 rows of data to use for training and testing. Iterative imputation was also applied, but it was found that KNN gave better results.

MATLAB was initially used to develop and implement the constant-parameter SIRD model to analyze COVID-19 transmission dynamics and estimate key parameters. Initial conditions for each compartment were specified, and the recovery rate, γ, was fixed based on an assumed mean infectious period of ten days due to the lack of specific recovery data in the Our World in Data dataset. This limitation motivated the transition to the Google COVID-19 Repository, which provided the epidemiological features necessary for parameter estimation. At this stage, Python was adopted for computing the constant-parameter SIRD model. Additionally, R was used to construct a time-varying SIRD model, which served as a benchmark for comparison with the PINN.

3 SIRD Model Implementations

We applied two different approaches, a constant and time-varying, to the classical SIRD model in order to obtain reliable epidemiological parameters and generate a strong baseline for later comparison with the Physics-Informed Neural Network. The modeling process evolved through multiple stages, beginning with MATLAB and the original dataset, and ultimately transitioning to Python and an improved data source that enabled more accurate parameter estimation.

SIRD Model Equations Before describing each implementation, we recall the classical SIRD equations that form the foundation of the modeling.

$$\begin{aligned}
\frac{dS}{dt} &= -\beta S I, \\
\frac{dI}{dt} &= \beta S I - \gamma I - \mu I, \\
\frac{dR}{dt} &= \gamma I, \\
\frac{dD}{dt} &= \mu I,
\end{aligned}$$

where β is the transmission rate, γ is the recovery rate, and μ is the mortality rate.

Constant versus Time-Varying SIRD Model The constant and time-varying SIRD models differ primarily in how they treat the transmission and transition parameters over time. In the constant SIRD model, the parameters β, γ, and μ are assumed to remain fixed throughout the modeling period. While this assumption simplifies analysis and interpretation, it may fail to capture real-world dynamics such as infection peaks. In contrast, the time-varying SIRD model allows these parameters to evolve over time, providing greater flexibility to reflect shifts in disease progression. As a result, time-varying models often yield improved fit to observed data and more realistic forecasts, though they require more complex estimation techniques and may be more sensitive to data quality.

MATLAB with Original Dataset The initial SIRD model was constructed and solved in MATLAB using the early COVID-19 dataset imported from Our World in Data. Parameter estimation was performed using MATLAB's `fminsearch` function, which minimized the sum of squared errors between the model predictions and the empirical infection and mortality time series.

Despite producing a workable model, this implementation suffered from several limitations. The original dataset had inconsistencies such as missing values, like recovered data, and fluctuations due to reporting delays. These issues introduced instability into the optimization process and yielded parameter estimates for β, γ, and μ that did not accurately reflect observed epidemiological behavior. As a result, the MATLAB-based SIRD model was able to roughly capture the qualitative dynamics of the pandemic but lacked the precision required for comparison with the PINN.

Python and R with Updated Google Dataset If there is interest in the code used for both the constant and time-varying SIRD models, it is publicly available in our GitHub repository, which contains the full implementations of each approach.

3.1 Contant SIRD Model

The SIRD model was reimplemented in Python using Jupyter Notebook and a more complete and structured dataset sourced from Google's COVID-19 data repository. This dataset provided cleaner cumulative counts and enabled a significantly more stable parameter fitting. The updated fits produced more accurate and epidemiologicaly plausible estimates for β, γ, and μ. These improved parameters resulted in a SIRD trajectory that aligned better with the real data. This constant-parameter Python model now serves as the primary baseline against which the more sophisticated learning-based models can be evaluated.

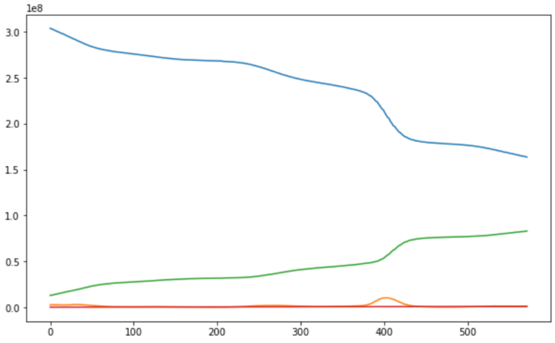


Figure 1: Observed Data

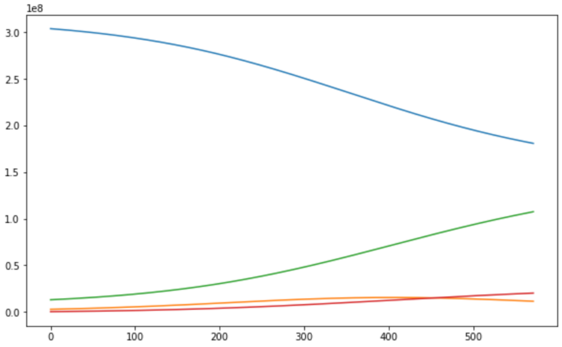
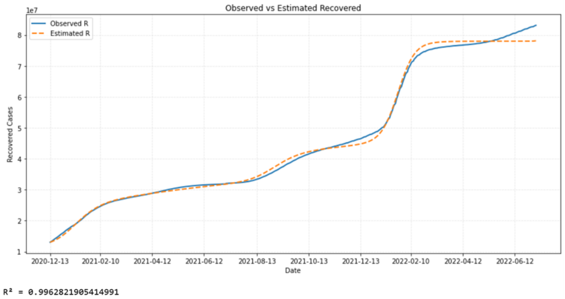


Figure 2: Constant SIRD Model Approximation

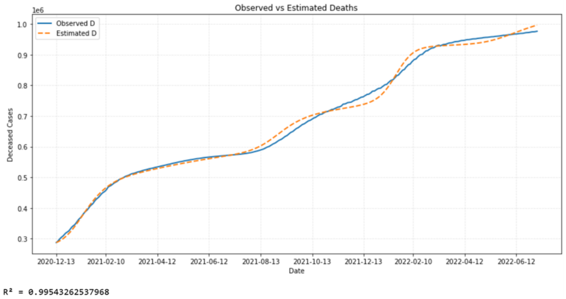
3.2 Time-Varying SIRD Model

To construct the time-varying SIRD model, we adapted the R code developed by [1] for a time-varying SIR framework. Their original implementation was modified to accommodate our dataset and extended to include a deceased compartment, resulting in a complete SIRD model. Allowing model parameters to vary over time enabled the model to capture changes like evolving outbreak conditions. In comparison to the constant-parameter SIRD model, the time-varying model demonstrated significantly improved performance, as it more accurately reflected the observed dynamics of the outbreak.



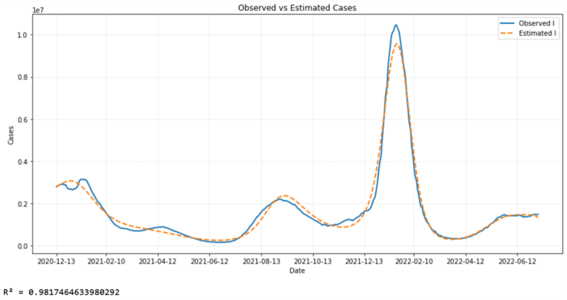
```
R² = 0.9962821905414991
```

Figure 3: Time-Varying Recovered Approximation



```
R² = 0.99543262537968
```

Figure 4: Time-Varying Deaths Approximation



```
R² = 0.9817464633980292
```

Figure 5: Time-Varying Cases Approximation

4 Physics-Informed Neural Networks (PINNs)

We implemented a neural network to predict the three compartments of the SIRD model: cumulative cases (C), cumulative deaths (D), and cumulative recoveries (R). The network takes as input a set of epidemiologically relevant features, including day number, cumulative vaccination rate, stringency index, and approximate recovered cases. The network outputs the predicted values of the three compartments simultaneously. The hidden layers of the network each consist of 32 neurons and employ the hyperbolic tangent activation function. This choice of activation introduces nonlinearity while maintaining smooth derivatives, which are essential for computing the physics-informed residuals. To reduce overfitting and improve generalization, we apply dropout with probability $p = 0.2$ after each hidden layer. Additionally, ℓ_2 regularization is applied to all network weights through a small weight decay term during training. The final layer is linear, producing the three outputs corresponding to C, D, and R. Let the neural network's predictions be denoted where N is the number of observations and the corresponding ground-truth values as

$$\mathbf{Y} = [C, D, R] \in R^{N \times 3}.$$

The total loss function combines a data-fitting term and a physics-informed term derived from the SIRD differential equations. To ensure the model fits the observed data, we compute the mean squared error (MSE) separately for each output:

$$\mathrm{MSE}_C = \frac{1}{N}\sum_{i=1}^{N}(\hat{C}_i - C_i)^2, \quad \mathrm{MSE}_D = \frac{1}{N}\sum_{i=1}^{N}(\hat{D}_i - D_i)^2, \quad \mathrm{MSE}_R = \frac{1}{N}\sum_{i=1}^{N}(\hat{R}_i - R_i)^2.$$

Because the three outputs differ significantly in magnitude and variance, we introduce output-specific weights (w_C, w_D, w_R) to balance their contributions:

$$\mathcal{L}_{\text{data}} = w_C \,\mathrm{MSE}_C + w_D \,\mathrm{MSE}_D + w_R \,\mathrm{MSE}_R.$$

To incorporate epidemiological knowledge, we enforce the SIRD system of differential equations. Let f_C, f_D, and f_R denote the residuals of the governing equations evaluated on the network predictions:

$$\begin{aligned} f_C &= \frac{d\hat{C}}{dt} - \beta\,\hat{S}\,\hat{I}, \\ f_D &= \frac{d\hat{D}}{dt} - \mu\,\hat{I}, \\ f_R &= \frac{d\hat{R}}{dt} - \gamma\,\hat{I}, \end{aligned}$$

where $\hat{I}$ is the estimated number of active infections, $\hat{S}$ is the susceptible fraction, and β, γ, μ are learnable epidemic parameters. The corresponding physics loss is defined as

$$\mathcal{L}_{\text{phys}} = \frac{1}{N}\sum_{i=1}^{N}\left(f_{C,i}^2 + f_{D,i}^2 + f_{R,i}^2\right).$$

Finally, the total loss is a weighted combination of the data and physics terms where λ is a hyperparameter controlling the contribution of the physics-informed component. Minimizing this loss encourages the network to both fit the observed epidemic data and respect the underlying SIRD dynamics.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \lambda\,\mathcal{L}_{\text{phys}},$$

4.1 Multi-Output PINN

We initially developed a multi-output PINN to simultaneously predict infected, deceased, and recovered populations. However, this approach exhibited substantial uncertainty and poor predictive performance for certain compartments. As summarized in Table 1, the model produced negative R^2 values for both the infected and recovered populations, indicating that the predictions were comparable to or worse than a baseline mean predictor. Although the death compartment achieved relatively strong performance, the overall instability of the multi-output PINN motivated the exploration of separate, single-output PINN models for each compartment.

Table 1: Performance Metrics for the Multi-Output PINN model.

Output	**MAPE (%)**	**R^2**
Infected	20.7405	-1.1161
Deaths	1.7083	0.9295
Recovered	16.3253	-0.2516

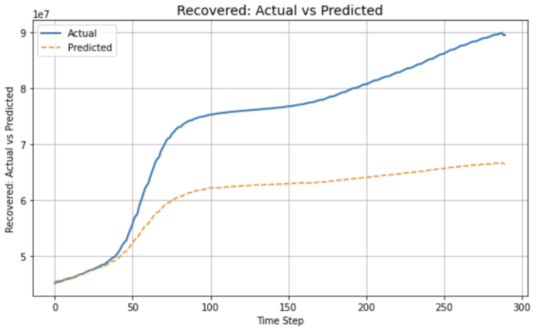


Figure 6: Multi-Output Recovered Approximation

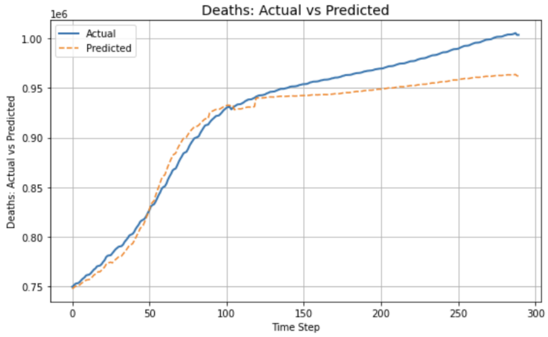


Figure 7: Multi-Output Deaths Approximation

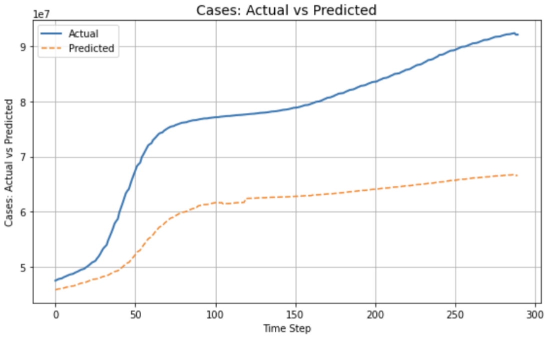


Figure 8: Multi-Output Cases Approximation

4.2 Individual PINNs

Due to the poor predictive performance of the multi-output PINN, we instead trained separate PINNs for each compartment. This approach resulted in improved accuracy across all outputs, as reflected by the performance metrics reported in Table 2. In particular, all R^2 values are now positive, representing a substantial improvement over the multi-output model. Additionally, the Mean Absolute Percentage Error (MAPE) values for the infected and recovered populations are significantly reduced, indicating better agreement between predicted and observed data.

Table 2: Performance Metrics for the Individual PINN models.

Output	**MAPE (%)**	$\mathbf{R^2}$
Infected	7.98	0.660
Deaths	3.157	0.6965
Recovered	12.475	0.2909

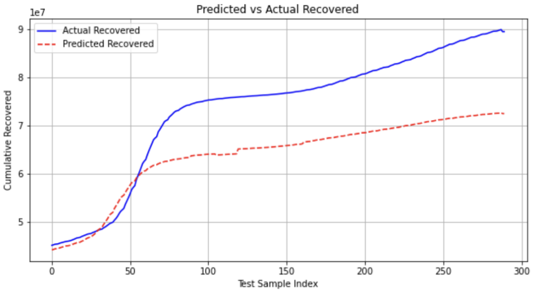


Figure 9: Individual Recovered Model

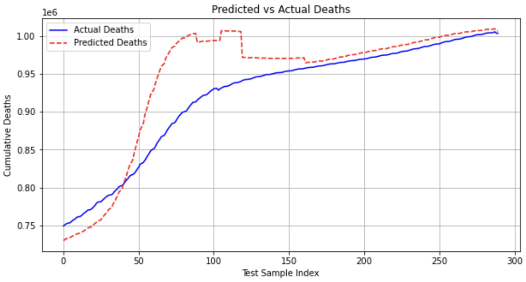


Figure 10: Individual Deaths Model

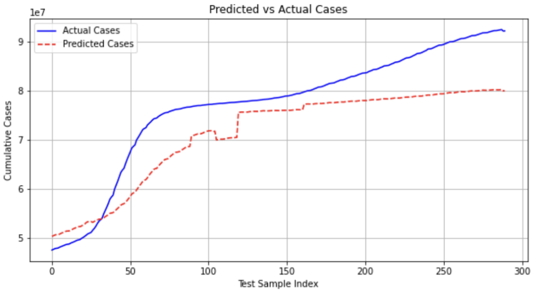


Figure 11: Individual Cases Model

5 Conclusions

Overall, the time-varying SIRD model outperformed the PINN-based approaches. This difference in performance may be due to the high sensitivity of PINN models to data scaling choices, including the imputation and normalization methods applied during preprocessing. Additionally, limited time for extensive hyperparameter tuning may have constrained the performance of the PINN models. Despite these limitations, the results from both the time-varying SIRD model and the PINN frameworks highlight the importance of accounting for changes in disease behavior over the course of the outbreak.
Data/Code Availability: The code developed and used for this research is available on GitHub. https://github.com/Aleya-Hadenfeldt

6 Future Work

Continuing this project, we plan to test additional imputation and scaling methods by exploring different ways to handle missing data and rescale it so the model can learn more effectively. We also aim to apply K-Fold cross-validation to evaluate the model's performance across multiple subsets of the data, helping to ensure reliability and reduce overfitting. Finally, we plan to apply our model to other diseases, such as measles and influenza, to assess whether it can generalize to different outbreaks while still producing accurate and interpretable results.

References

[1] Hyokyoung G Hong and Yi Li. Estimation of time-varying reproduction numbers underlying epidemiological processes: A new statistical tool for the covid-19 pandemic. *PloS one*, 15(7):e0236464, 2020.

[2] Yulia Abramova and Vasiliy Leonenko. The past helps the future: Coupling differential equations with machine learning methods to model epidemic outbreaks. In *Computational Science – ICCS 2024: 24th International Conference, Malaga, Spain, July 2–4, 2024, Proceedings, Part IV*, page 247–254, Berlin, Heidelberg, 2024. Springer-Verlag.

[3] Seungchan Ko and Sang Hyeon Park. Vs-pinn: A fast and efficient training of physics-informed neural networks using variable-scaling methods for solving pdes with stiff behavior. *Journal of Computational Physics*, 494:112516, 2023.

[4] Sagi Shaier, Maziar Raissi, and Padmanabhan Seshaiyer. Data-driven approaches for predicting spread of infectious diseases through dinns: Disease informed neural networks, 2022.