Group No. 8

Dyuti Dasmahapatra

Ankur Kaushal

Himanshu Sharma

Priyanshu Yadav

Explainable AI for COVID-19 Forecasting: Utilizing SHAP and LIME to Enhance Model Interpretability

A. Dyuti Dasmahapatra[1,*], B. Ankur Kaushal[2,*], C. Himanshu Sharma[3,*] and D. Priyanshu Yadav[4,*]

[1] Bml Munjal University, Gurugram, India

Abstract

The COVID-19 pandamic has underscored the neccesity for accurate forecasting models to inform public health strategies. This research compares various deep learning models, including Long Short-Term Memory (LSTM), Bidirectional LSTM, Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP), hybrid architectures, for predicting COVID-19 cases. To enhance model performence and interpretability, we employed Bayesian optimization for hyparparameter tuning and implemanted Explainable AI (XAI) techniques, specifically SHAP. Our findings reveal the varying efficacy of these models in forecasting cases and provide critical insights into feature importance and model behavior, which can aid public health authorities in making informed decisions. This study emphasizes the significance of integrating XAI methodologies in public health to enhance trust and understanding of AI-driven forecasts, ultimately contributing to effective health interventions.

Keywords: COVID-19 Forecasting, Deep Learning, Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), Hybrid Models, Explainable AI (XAI), SHAP (SHapley Additive exPlanations)

***Corresponding author:**
✉A. Dyuti Dasmahapatra
dyuti.dasmahapatra.21cse@bmu.edu.in
✉B. Ankur Kaushal
ankur.kaushal.21cse@bmu.edu.in
✉C. Himanshu Sharma
himanshu.sharma.21cse@bmu.edu.in
✉D. Priyanshu Yadav
priyanshu.yadav.21cse@bmu.edu.in

1 Introduction

The COVID-19 pandemic has posed unprecedentad challenges to global health systems, necessitating the development of robust predictive models to guide public health responses. Accurate forecasting of COVID-19 case trajectories is critical for effective resource allocation, planning, and timely intervention [1]. Various deep learning techniques have emerged as powerful tools for this purpose, leveraging extensive datasets to derive meaningful insights and projections. Models such as Long Short-Term Memory (LSTM), Bidirectional LSTM, Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), Multi-Layer Perceptron (MLP), and hybrid architectures have shown promise in predicting COVID-19 trends based on historical data [2][3][5].

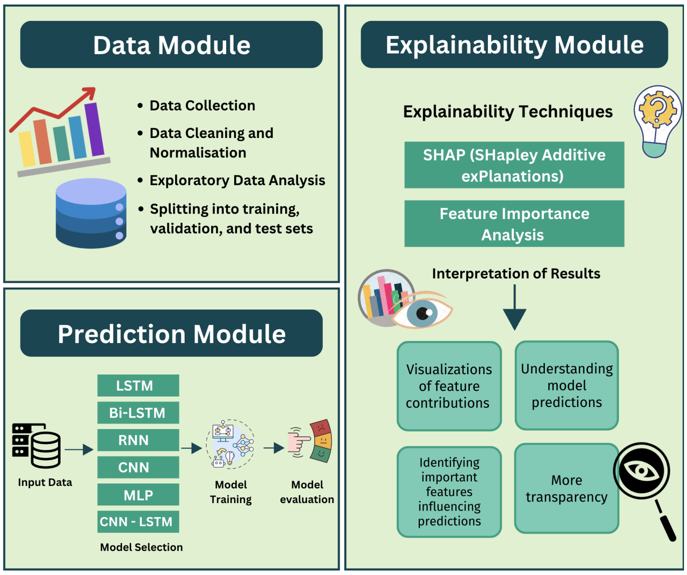

Data Module
Data Collection
Data Cleaning and Normalisation
Exploratory Data Analysis
Splitting into training, validation, and test sets
Prediction Module
Input Data
LSTM
Bi-LSTM
RNN
CNN
MLP
CNN - LSTM
Model Selection
Model Training
Model evaluation
Explainability Module
Explainability Techniques
SHAP (SHapley Additive exPlanations)
Feature Importance Analysis
Interpretation of Results
Visualizations of feature contributions
Understanding model predictions
Identifying important features influencing predictions
More transparency

Figure 1. Schematic Representation of Our Framework, Including the Data Module, Prediction Module, and Explainability Module

However, while these models can provide accurate predictions, their complex nature often obscures the decision-making processes underlying their outputs. This lack of transparency can lead to distrust among stakeholders who rely on these predictions for public health decisions [4]. This study aims to address this issue by implementing Explainable AI (XAI) methodologies, specifically SHAP, to enhance the interpretability of deep learning models used for COVID-19 forecasting [8]. By providing insights into the factors that influence model predictions, this research aims to empower stakeholders, including policymakers and public health officials, to make informed decisions based on reliable forecasts, thereby contributing to better public health outcomes.

2 Problem Description

Despite the effectiveness of deep learning models in predicting COVID-19 cases, the opaque nature of these algorithms poses significant challenges for public health officials and researchers who rely on their outputs [1][2]. The inability to interpret model predictions can hinder decision-making and trust in AI technologies [3]. Public health policies often depend on timely and accurate forecasts, making the interpretability of these models crucial [4]. This project seeks to address the gap in understanding how these models derive their predictions by applying XAI methodologies, specifically SHAP [5]. The primary objective is to demystify the inner workings of COVID-19 forecasting models, enabling stakeholders to gain insights into the factors influencing predictions and facilitating more effective public health interventions [6].

3 Litreature Review:

3.1 Deep learning in public health: Comparative predictive models for COVID-19 case forecasting Muhammad Usman Tariq, Shuhaida Binti Ismail (2024)

Tariq and Ismail (2024) [1] conducted a pivotal study on COVID-19 forecasting, utilizing a comprehensive dataset that spans from January 2020 to August 2023. The dataset integrates multiple dimensions of COVID-19 trends, including daily case counts, recoveries, fatalities, and essential contextual information such as demographic data, government-imposed health interventions, mobility patterns, and vaccination rates. Data was sourced from globally recognized institutions, including the World Health Organization (WHO), and local health ministries. This holistic approach enabled the researchers to apply and train various deep learning models, which not only improved forecast accuracy but also provided deeper insights into the pandemic's progression over time. The incorporation of socio-economic and epidemiological factors made the models particularly effective in capturing the non-linear and dynamic nature of COVID-19 transmission. The findings from this study offer valuable contributions to public health strategy development by demonstrating how advanced deep learning techniques can aid in real-time decision-making for pandemic management.

3.2 Deep Learning for COVID-19 Detection and Diagnosis: A Review, Gupta, A., and Gupta, S.(2021)

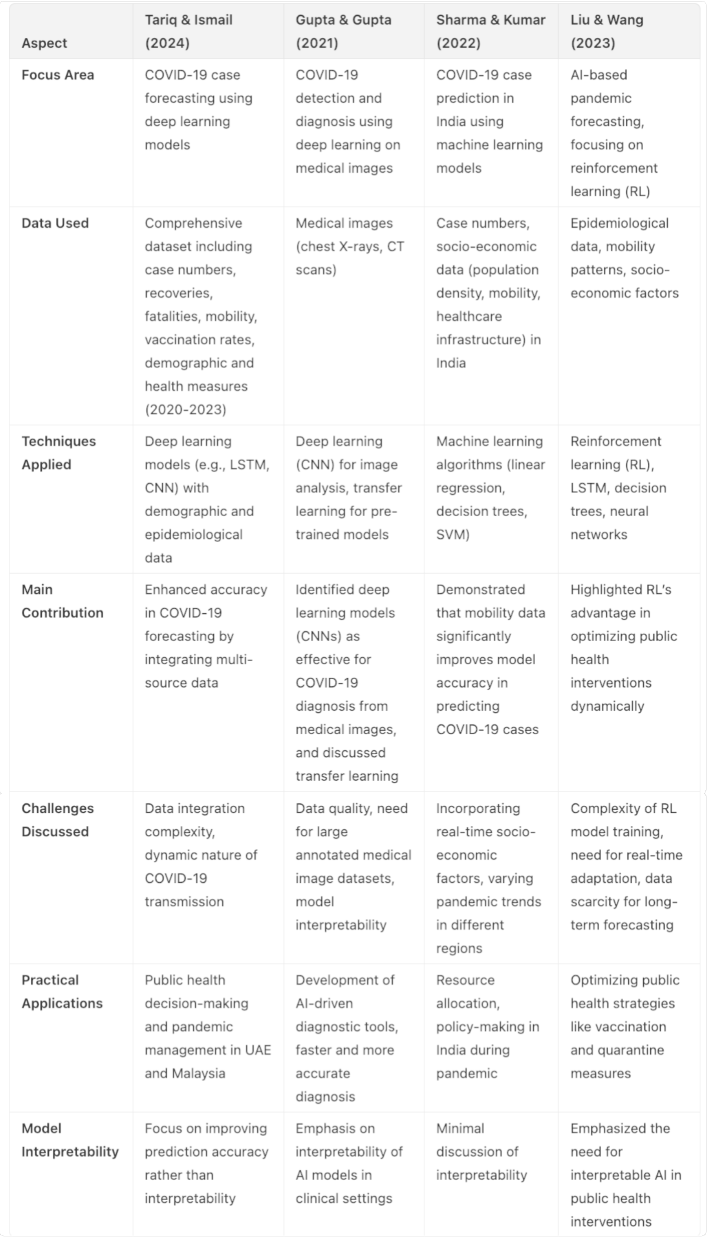
Gupta and Gupta (2021) [2] presented a thorough review of deep learning methodologies applied to the detection and diagnosis of COVID-19, focusing primarily on medical imaging. Their review highlights the transformative potential of convolutional neural networks (CNNs), which have shown remarkable accuracy in analyzing radiological images, such as chest X-rays and CT scans, to detect COVID-19 infections. The study underscores the advantages of transfer learning, wherein pre-trained models are fine-tuned using COVID-specific datasets, significantly expediting the model training process while requiring fewer computational resources. The authors also addressed challenges related to data scarcity, emphasizing the critical need for large, annotated medical imaging datasets to improve diagnostic models. Furthermore, they discuss the importance of model interpretability, especially in clinical settings where explainability is crucial for medical professionals to trust and adopt AI-driven diagnostic tools. By bridging clinical data with imaging results, this review highlights the potential for deep learning models to enhance diagnostic accuracy, reduce false positives/negatives, and assist in the early detection of COVID-19. This comprehensive synthesis provides valuable insights for the future development of AI-driven diagnostics in pandemic scenarios.

3.3 Predicting COVID-19 Cases Using Machine Learning: A Case Study of India, Sharma, R., Kumar, A (2022)

In their case study, Sharma and Kumar (2022) [3] explored the effectiveness of machine learning algorithms for predicting COVID-19 case numbers in India, a country profoundly affected by the pandemic. The authors applied a range of machine learning techniques, including linear regression, decision trees, and support vector machines (SVMs), to evaluate the performance of each model in forecasting case counts. They emphasize that the inclusion of socio-economic variables—such as population density, mobility patterns, and healthcare infrastructure—enhanced the models' predictive accuracy. The study's dataset was expansive, incorporating historical case data, government intervention policies, and demographic details, allowing for a comprehensive understanding of factors that influence the virus's spread. A significant finding from their research was the impact of mobility data on model performance; algorithms that integrated real-time mobility and lockdown data demonstrated improved accuracy in predicting case surges. Their study underscores the necessity for adaptive forecasting models that can evolve with changing pandemic conditions. The research offers practical insights for public health officials and policymakers in India, contributing to more precise resource allocation and informed decision-making for mitigating the pandemic's impact.

3.4 COVID-19 Pandemic Forecasting with AI: A Survey of Reinforcement Learning and Other AI Models Liu, Y., and Wang, X. (2023)

Liu and Wang (2023) provided an extensive survey on the application of artificial intelligence, particularly reinforcement learning (RL), in the forecasting of COVID-19 case trajectories. The paper discusses how RL-based models differ from traditional machine learning and deep learning models by focusing on decision-making processes over time, which are essential for dynamically evolving crises like the COVID-19 pandemic. They compare RL with other AI techniques such as long short-term memory (LSTM) networks, decision trees, and neural networks, outlining the unique advantages of RL models in optimizing intervention strategies and public health responses. The study also evaluates the effectiveness of AI models in incorporating multi-source data, such as epidemiological data, mobility trends, and socio-economic factors, to offer more accurate and actionable predictions. This survey highlights the potential of reinforcement learning to not only predict pandemic trends but also to inform public health policies by optimizing resource allocation, quarantine measures, and vaccination strategies. The authors conclude that RL, combined with other AI models, can provide a more adaptive framework for managing future public health crises.



Aspect	Tariq & Ismail (2024)	Gupta & Gupta (2021)	Sharma & Kumar (2022)	Liu & Wang (2023)
Focus Area	COVID-19 case forecasting using deep learning models	COVID-19 detection and diagnosis using deep learning on medical images	COVID-19 case prediction in India using machine learning models	AI-based pandemic forecasting, focusing on reinforcement learning (RL)
Data Used	Comprehensive dataset including case numbers, recoveries, fatalities, mobility, vaccination rates, demographic and health measures (2020-2023)	Medical images (chest X-rays, CT scans)	Case numbers, socio-economic data (population density, mobility, healthcare infrastructure) in India	Epidemiological data, mobility patterns, socio-economic factors
Techniques Applied	Deep learning models (e.g., LSTM, CNN) with demographic and epidemiological data	Deep learning (CNN) for image analysis, transfer learning for pre-trained models	Machine learning algorithms (linear regression, decision trees, SVM)	Reinforcement learning (RL), LSTM, decision trees, neural networks
Main Contribution	Enhanced accuracy in COVID-19 forecasting by integrating multi-source data	Identified deep learning models (CNNs) as effective for COVID-19 diagnosis from medical images, and discussed transfer learning	Demonstrated that mobility data significantly improves model accuracy in predicting COVID-19 cases	Highlighted RL's advantage in optimizing public health interventions dynamically
Challenges Discussed	Data integration complexity, dynamic nature of COVID-19 transmission	Data quality, need for large annotated medical image datasets, model interpretability	Incorporating real-time socio-economic factors, varying pandemic trends in different regions	Complexity of RL model training, need for real-time adaptation, data scarcity for long-term forecasting
Practical Applications	Public health decision-making and pandemic management in UAE and Malaysia	Development of AI-driven diagnostic tools, faster and more accurate diagnosis	Resource allocation, policy-making in India during pandemic	Optimizing public health strategies like vaccination and quarantine measures
Model Interpretability	Focus on improving prediction accuracy rather than interpretability	Emphasis on interpretability of AI models in clinical settings	Minimal discussion of interpretability	Emphasized the need for interpretable AI in public health interventions

Figure 2. Comparision of the Papers

4 Methodology

This section describes the steps taken to devalop, train, and evaluate models for forecasting COVID19 cases. It includes data collection, preprocessing, model implementation, hyperparametertuning, and performance evaluation. SHAP is used to improve model interpretability. This includes:

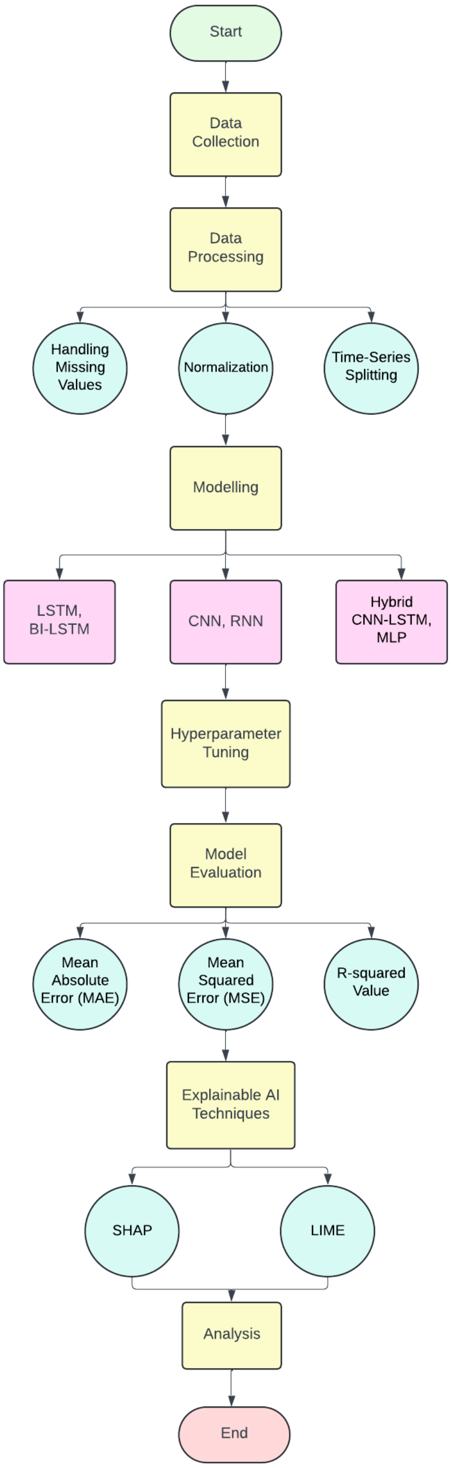


Figure 3. Flowchart of the methodology

4.1 Data Collection

In this study, we utilized the COVID19 datasat provided by Johns Hopkins University [9], which contains daily-level information on the number of confirmed cases, deaths, and recoveries associated with the 2019 Novel Coronavirus (2019-nCoV). The dataset serves as a crucial resource for understanding the progression of the COVID-19 pandemic and enables the application of various predictive models to forecast future trends.

The primary file used in this research is `covid_19_data.csv`, which consists of the following columns:

- **Sno**: A serial number assigned to each row for easy reference.
- **ObservationDate**: The date of observation formatted in MM/DD/YYYY. This date indicates when the data was recorded and is essential for time series analysis.
- **Province/State**: The specific province or state of the observation. This field may be empty when the data is aggregated at the country level.
- **Country/Region**: The country or region where the observations were made. This column helps in categorizing the data for different geographical locations.
- **Last Update**: The timestamp in Coordinated Universal Time (UTC) indicating when the data row was last updated. Due to the non-standardized nature of this field, it may require cleaning and preprocessing before use.
- **Confirmed**: The cumulative number of confirmed COVID-19 cases reported up to that date. This metric is essential for tracking the spread of the virus over time.
- **Deaths**: The cumulative number of deaths attributed to COVID-19 until the specified date. This information is crucial for assessing the severity of the outbreak and the effectiveness of public health responses.
- **Recovered**: The cumulative number of recovered cases reported by that date. Understanding recovery rates is vital for evaluating the impact of healthcare interventions and the overall progress of the pandemic.

The dataset begins from **January 22, 2020**, and contains timeseries data, meaning that the values for confirmed cases, deaths, and recoveries are cumulative totals rather than daily counts. This characteristic allows for comprehensive analysis of the pandemic's trajectory over time.

By leveraging this dataset, we can apply various deep learning techniques to forecast COVID-19 cases, as well as explore relationships between different features, thereby gaining valuable insights that can aid public health authorities in decision-making and resource allocation.

4.2 Data Preprocessing

Before modeling, the collected data underwent several preprocessing steps:

- **Handling Missing Values**: Missing data points were addressed using interpolation and forward-fill methods to ensure continuity in time series analysis.
- **Normalization**: The features were normalized to scale between 0 and 1 using Min-Max scaling, which is essential for neural network models to converge effectively. The formula for Min-Max scaling is given by:

$$X' = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}} \tag{1}$$

  where X' is the normalized value, X is the original value, X_{min} is the minimum value in the feature, and X_{max} is the maximum value in the feature.
- **timeseries Splitting**: The dataset was divided into training (70%), validation (15%), and test sets (15%) to evaluate model performance objectively.

4.3 Model Architecture

4.3.1 Long Short-Term Memory (LSTM)

LSTM networks were employed due to their ability to capture long-range dependencies in sequential data. The architecture consisted of:

- An input layer with features representing time steps.
- One or more LSTM layers (with 50-100 units) followed by dropout layers to prevent overfitting. The LSTM cell state c_t and hidden state h_t are updated as follows:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad \text{(forget gate)} \tag{2}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad \text{(input gate)} \tag{3}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad \text{(output gate)} \tag{4}$$
$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t \quad \text{(cell state update)} \tag{5}$$
$$h_t = o_t \cdot \tanh(c_t) \quad \text{(hidden state)} \tag{6}$$

- A fully connected output layer with a single neuron to predict the number of COVID-19 cases.

4.3.2 Bidirectional LSTM

To enhance the LSTM model's performance, a Bidirectional LSTM was utilized, allowing the model to learn information from both past and future states. The architecture included:

- An input layer similar to the LSTM model.
- Two LSTM layers: one processing input in the forward direction and the other in the backward direction.
- Dropout layers for regularization.
- A dense output layer to yield predictions.

4.3.3 Convolutional Neural Network (CNN)

CNNs were integrated to capture spatial hierarchies in the dataset, particularly when working with time series data reshaped as images. The CNN architecture included:

- Input layers that processed reshaped time series data.
- Several convolutional layers with varying filter sizes followed by activation functions (ReLU). The convolution operation can be expressed as:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n) K(i - m, j - n) \tag{7}$$

  where I is the input image, K is the filter, and (i, j) are the coordinates of the output feature map.
- Max pooling layers to downsample feature maps, defined as:

$$P(i, j) = \max_{(m,n) \in R} I(m, n) \tag{8}$$

  where R is the region of interest.
- A flattening layer leading to a fully connected output layer.

4.3.4 *Recurrent Neural Network (RNN)*

RNNs were also evaluated to capture temporal dependencies. The RNN architecture consisted of:

- An input layer receiving the time series data.
- One or more recurrent layers with units designed to process the sequences. The RNN cell updates its hidden state as follows:

$$h_t = \sigma(W_h \cdot h_{t-1} + W_x \cdot x_t + b) \tag{9}$$

  where h_t is the current hidden state, x_t is the current input, and W_h, W_x, and b are the weights and bias.
- A dense output layer for case predictions.

4.3.5 *Hybrid CNN-LSTM Model*

To leverage both spatial and temporal features, a hybrid CNN-LSTM model was developed. This architecture included:

- An input layer processing the reshaped data.
- Convolutional layers for spatial feature extraction.
- Flattening and reshaping to prepare the data for the LSTM layers.
- One or more LSTM layers to capture temporal dependencies.
- A final dense output layer.

4.3.6 *Multi-Layer Perceptron (MLP)*

An MLP model was also trained for comparative analysis. The architecture included:

- An input layer corresponding to the features of the dataset.
- Two to three hidden layers with varying numbers of neurons (e.g., 50, 100, 50) using activation functions such as ReLU. The output of a neuron is computed as:

$$y = \sigma(W \cdot x + b) \tag{10}$$

  where y is the output, W is the weight matrix, x is the input vector, and b is the bias.
- A dropout layer for regularization.
- A single-output neuron for the prediction of COVID-19 cases.

4.3.7 *Hybrid CNN-Bi-LSTM Model*

To capture both spatial and temporal features with enhanced bidirectional information flow, a hybrid CNN-Bi-LSTM model was developed. This architecture included:

- **Input Layer:** Processing the reshaped data for compatibility with subsequent layers.
- **Convolutional Layers:** For extracting spatial features from the input data, capturing local patterns.
- **Flattening and Reshaping:** Preparing the data for the Bi-LSTM layers by converting the feature maps into a suitable format.
- **Bidirectional LSTM Layers:** These layers capture temporal dependencies in both forward and backward directions, allowing the model to learn patterns from past and future information simultaneously.
- **Dense Output Layer:** A fully connected layer producing the final predictions or classifications based on the learned spatial and temporal features.

4.3.8 *Hybrid MLP-CNN-LSTM Model*

To leverage the strengths of multiple architectures for feature extraction and learning complex relationships, a hybrid MLP-CNN-LSTM model was designed. This architecture included:

- **Input Layer:** Reshaping and preparing the input data for subsequent layers.
- **MLP Layers:** These fully connected layers are responsible for learning high-level feature representations and combining inputs in a non-linear fashion.
- **Convolutional Layers:** For extracting local spatial features and patterns, which are then passed to the LSTM layers for temporal learning.
- **Flattening and Reshaping:** Converting the output from the convolutional layers into a suitable format for the LSTM layers.
- **LSTM Layers:** These layers capture long-term temporal dependencies in the data, helping to model sequential information effectively.
- **Dense Output Layer:** The final layer that makes predictions or classifications based on the learned

spatial and temporal representations from the MLP, CNN, and LSTM layers.

4.4 Hyperparametertuning

To optimize model performance, Bayesian optimization was employed for hyperparametertuning. Parameters such as learning rate (α), batch size (B), number of epochs (E), and the number of layers/units were fine-tuned. A validation set was used to evaluate model performance, and the best configuration was selected based on Mean Squared Error (MSE) and Mean Absolute Error (MAE) metrics.

4.5 Explainable AI Techniques

To enhance model interpretability, SHAP (SHapley Additive exPlanations) values were computed for the LSTM, Bidirectional LSTM, CNN, and Hybrid CNN-LSTM models. The SHAP value for a feature i is defined as:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left(f(S \cup \{i\}) - f(S)\right) \tag{11}$$

where N is the set of all features, $f(S)$ is the model's prediction using the feature subset S, and $|S|$ is the size of the subset.

SHAP provides a unified measure of feature importance, allowing stakeholders to understand the impact of individual features on model predictions. The insights derived from SHAP values were crucial in identifying key drivers of COVID-19 case trends, thereby increasing trust in the model outputs.

In addition to SHAP, LIME (Local Interpretable Model-Agnostic Explanations) was also employed for model explainability. LIME works by approximating the model locally with an interpretable surrogate model that is trained on perturbed samples around the prediction of interest. The objective is to explain the model's behavior in a local region around the instance being analyzed. The explanation is given as a weighted sum of the features:

$$f_L(x) = \sum_{j=1}^{m} \beta_j \cdot \phi_j(x) \tag{12}$$

where $f_L(x)$ is the local surrogate model for instance x, β_j are the coefficients learned by the surrogate model, and $\phi_j(x)$ are the perturbed feature values. The goal is to create an interpretable model $f_L(x)$ that mimics the behavior of the original complex model locally around the instance.

LIME provides insight into how the model makes predictions for individual instances, allowing for a better understanding of the factors contributing to specific predictions, thus enhancing trust and model interpretability.

4.6 Model Evaluation

The performance of each model was evaluated using the test set, measuring metrics such as:

- **Mean Absolute Error (MAE)**: The Mean Absolute Error (MAE) measures the average magnitude of errors in a set of predictions, without considering their direction (i.e., whether the prediction is above or below the actual value). It is the average over the test sample of the absolute differences between prediction and actual observation.

  $$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{13}$$

  where y_i is the actual value, $\hat{y}_i$ is the predicted value, and n is the number of observations.

- **Mean Squared Error (MSE)**: The Mean Squared Error (MSE) measures the average squared difference between predicted values and actual values. MSE gives a relatively high weight to large errors due to the squaring of the differences, making it sensitive to outliers.

  $$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{14}$$

- **Root Mean Squared Error (RMSE)**: The Root Mean Squared Error (RMSE) is the square root of the average of squared differences between predicted values and actual values. RMSE provides a measure of how well the model predicts the outcome, with the same units as the response variable.

  $$\text{RMSE} = \sqrt{\text{MSE}} \tag{15}$$

These metrics were used to compare the models' performances and select the best-performing model for final predictions.

5 Results:

5.1 Distribution of Confirmed, Death, and Recovered Cases:

The distribution of confirmed, death, and recovered cases is essential for understanding the overall impact of the COVID-19 pandemic.

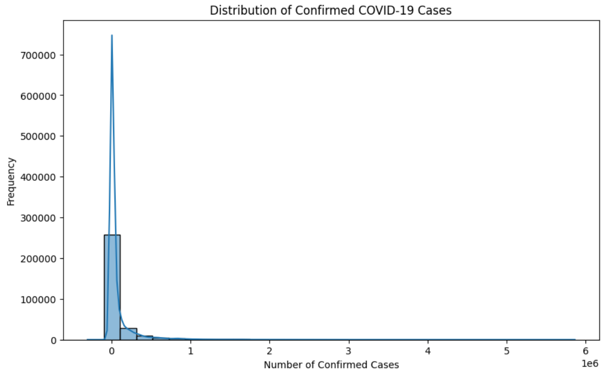


Figure 4. Distribution of Confirmed Cases

The distribution is highly skewed, with most data points showing low case counts and a peak near zero. The frequency drops rapidly as case numbers rise, extending up to 6 million cases, indicating some extreme values.

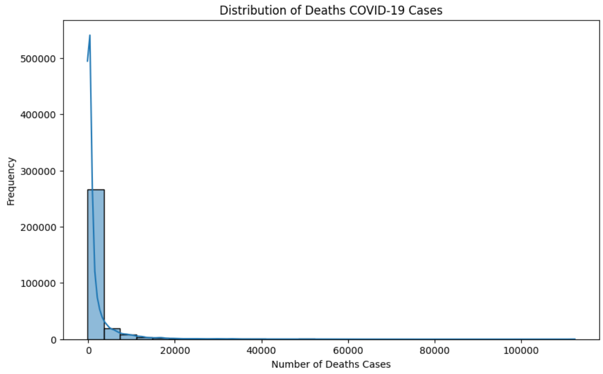


Figure 5. Distribution of Death Cases

Similar to confirmed cases, deaths show a high frequency of low counts, peaking near zero, with a tail extending to around 100,000 deaths.

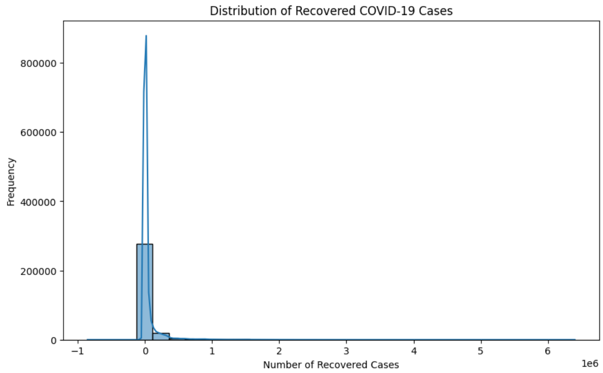


Figure 6. Distribution of Recovered Cases

The recovered cases are also skewed, though the peak is slightly further from zero, indicating many areas had some recoveries. The distribution extends to 6 million recoveries, like the confirmed cases graph.

5.2 Time Series Analysis

The time series analysis provides a comprehensive view of the trends and fluctuations in confirmed, death, and recovered cases over the course of the pandemic. The following figure illustrates:

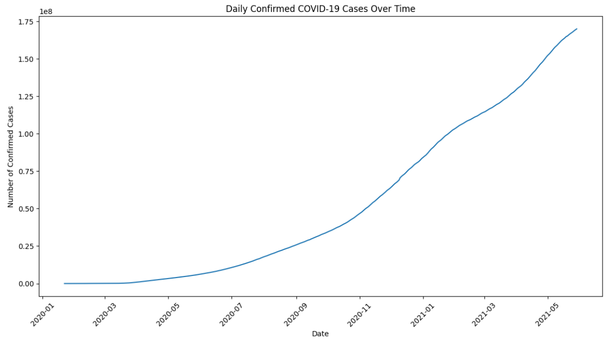


Figure 7. Time Series of Confirmed Cases

The graph shows a steady increase in daily confirmed cases. The curve starts relatively flat in early 2020, then begins to rise more steeply around mid-2020. The increase becomes more pronounced towards the end of 2020 and into 2021, with the curve becoming steeper. By the end of the shown period, the daily confirmed cases reach nearly 1.75 million.

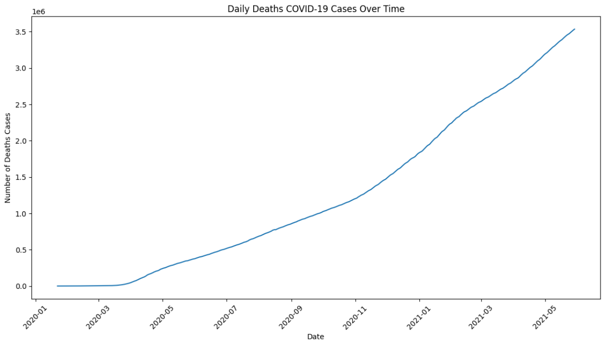


Figure 8. Time Series of Deaths Cases

The graph depicts daily deaths attributed to COVID-19. It follows a similar pattern to the confirmed cases graph, but with lower numbers. The curve remains relatively flat until mid-2020, then begins to rise more sharply. The increase in daily deaths accelerates towards the end of 2020 and into 2021. By the end of the period shown, daily deaths approach 3.5 million.

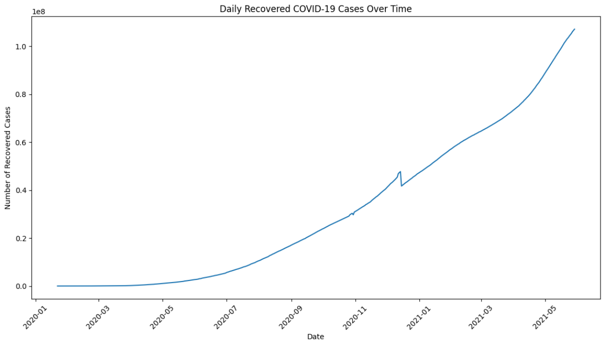


Figure 9. Time Series of Recovered Cases

The graph shows daily recovered cases. This curve also shows an overall increasing trend, but with some unique features. It remains flat for a longer period at the beginning, suggesting a lag in recovery data or reporting. There's a noticeable jump in the curve around late 2020 or early 2021, which could indicate a change in reporting methods or a surge in recoveries. After this jump, the curve continues to rise steeply, reaching over 1 million daily recoveries by the end of the period.

5.3 Model Metrics

The performance of various models used to predict COVID-19 cases is summarized in the table below. Key metrics evaluated include Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2) scores, which provide insights into model accuracy and fit:

Model	MSE	MAE	R^2 Score
LSTM	0.00009	0.00707	0.99902
Bidirectional LSTM	0.00023	0.01067	0.99760
CNN	0.00013	0.00820	0.99866
CNN-LSTM	0.00007	0.00524	0.99924
MLP	0.00014	0.00776	0.99854
RNN	0.00264	0.04013	0.97259
MLP+CNN+LSTM	0.00018	0.00904	0.99811
CNN+Bi-LSTM	0.00002	0.00281	0.99980

Table 1. Model Performance Metrics (Rounded to 5 Decimal Places)

The Hybrid CNN + Bi-LSTM model outperforms the others in terms of MSE, MAE, and R^2 score, demonstrating its effectiveness in capturing the underlying trends of the COVID-19 data. On the other hand, the RNN model exhibits the highest MSE and MAE, indicating that it struggles more than the other models to accurately predict the confirmed cases.

5.4 Actual vs. Predicted Graphs:

The following graphs compare the actual versus predicted cases for the various models used in this study. These visualizations help illustrate the models' performance and their ability to capture the trends in the COVID-19 data.

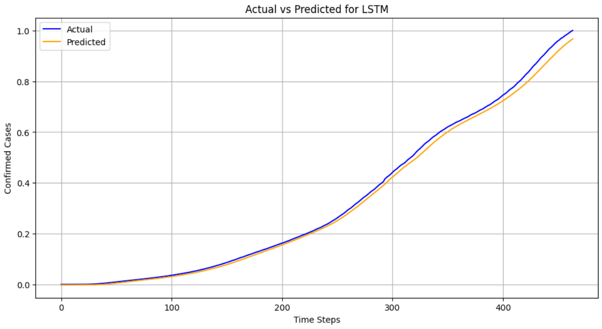


Figure 10. Actual vs Predicted for LSTM

The LSTM model shows a fairly close prediction to the actual data. The predicted line (orange) follows the general trend of the actual line (blue) quite closely, with some minor deviations. The model seems to slightly underestimate the number of cases towards the end of the time period.

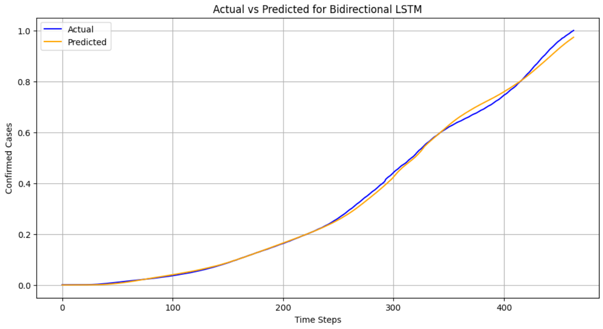


Figure 11. Actual vs Predicted for Bidirectional LSTM

The Bidirectional LSTM model appears to perform slightly better than the standard LSTM. The predicted line aligns more closely with the actual data, especially in the middle sections. There are still some minor discrepancies, particularly towards the end of the time period, but overall, the fit seems tighter.

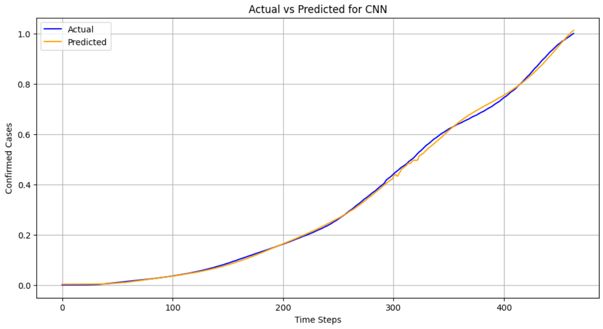


Figure 12. Actual vs Predicted for CNN

The CNN model also shows a good fit to the actual data. Its predictions closely follow the actual trend, with very minimal deviations. The CNN model seems to perform comparably well to the Bidirectional LSTM, showing accurate predictions across most of the time steps.

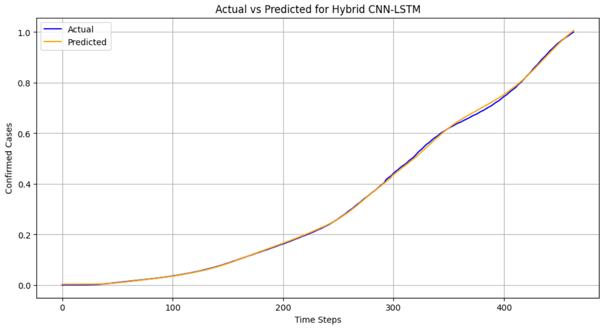


Figure 13. Actual vs Predicted for Hybrid CNN-LSTM

This model combines Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architectures. The prediction (orange line) very closely follows the actual data (blue line), with minimal deviations. This hybrid approach seems to capture both the overall trend and local fluctuations quite accurately.

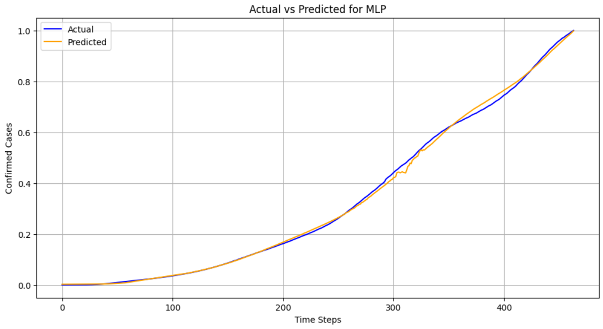


Figure 14. Actual vs Predicted for MLP

The Multi-Layer Perceptron, a type of feedforward neural network, also shows good performance. The predicted line closely follows the actual data for most of the time period. There are some minor deviations, particularly around the 300-350 time step range, but overall, the fit is quite good.

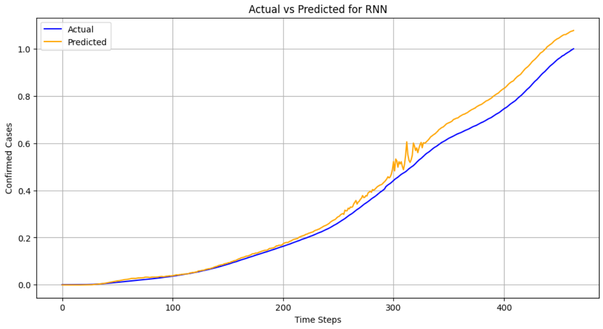


Figure 15. Actual vs Predicted for RNN

The Recurrent Neural Network model shows the most noticeable differences among the three. While it captures the general trend, there are more significant deviations from the actual data:

- The RNN tends to overpredict the number of cases, especially in the latter half of the time period.
- There's visible volatility in the predictions around the 300-350 time step range, shown by the jagged lines in the orange prediction curve.
- The overprediction becomes more pronounced towards the end of the time period.

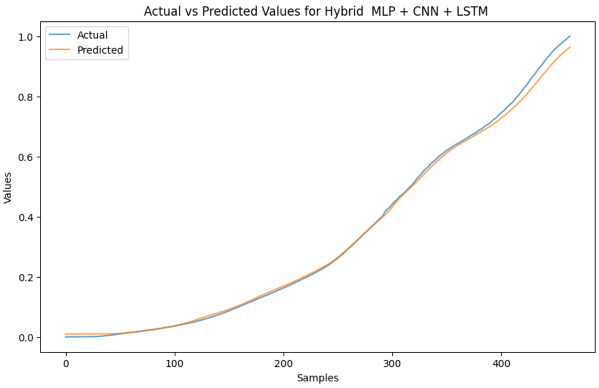


Figure 16. Actual vs Predicted for Hybrid MLP+CNN+LSTM

The graph shows a hybrid model combining MLP (Multilayer Perceptron), CNN (Convolutional Neural Network), and LSTM (Long Short-Term Memory). There is a slight divergence between the predicted and actual values, particularly in the higher range of values (towards the right). This indicates the model performs well overall but may slightly underestimate in some cases.

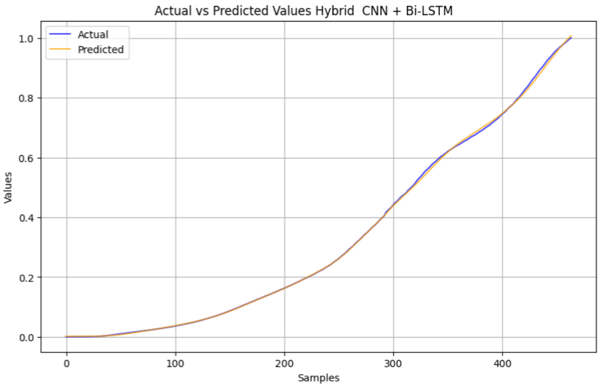


Figure 17. Actual vs Predicted for Hybrid CNN+Bi-LSTM

This graph evaluates a hybrid model combining CNN with Bi-LSTM (Bidirectional LSTM). The predicted values closely align with the actual values across the entire range. There is minimal visible divergence, indicating a better fit than the first model. The CNN + Bi-LSTM model appears to have better prediction accuracy, likely due to the bidirectional nature of the LSTM, which captures dependencies in both forward and backward directions.

5.5 Explanation of XAI Using SHAP Results

In this study, we implemented Explainable AI (XAI) techniques, specifically SHAP (SHapley Additive exPlanations), to enhance the interpretability of our deep learning models used for COVID-19 forecasting. SHAP values help in understanding the contribution of each feature to the model's predictions, providing insights into how different factors influence the forecasting of confirmed cases, deaths, and recoveries.

SHAP values are derived from cooperative game theory, where each feature is treated as a player contributing to the prediction outcome. The SHAP framework assigns a value to each feature that reflects its importance in driving the model's predictions. This method not only identifies which features are influential but also quantifies their effects, allowing us to see how they interact with one another in contributing to the model's output.

- **LSTM:**

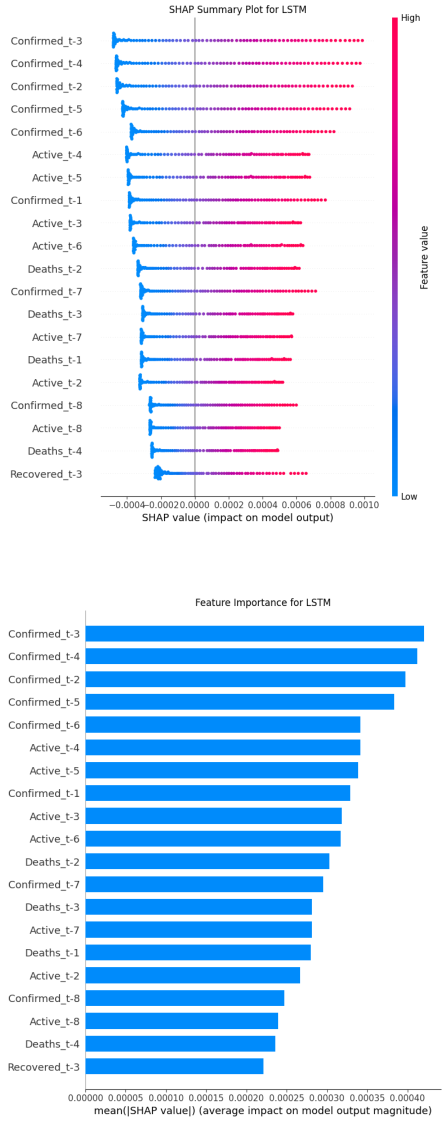


Figure 18. (a) SHAP summary plot for LSTM (b) Feature importance for LSTM

Plot for LSTM shows the impact of each feature on the model output. Features are ranked by importance, with color indicating positive (red) or negative (blue) impact. Confirmed cases from 1-3 days ago have the highest impact, and recent data generally has more influence than older data.

The Feature Importance chart for LSTM displays the average magnitude of SHAP values for each feature. Recent confirmed cases are the most important, followed by active cases from recent days. Death counts have moderate importance.

- **Bi-directional LSTM:**

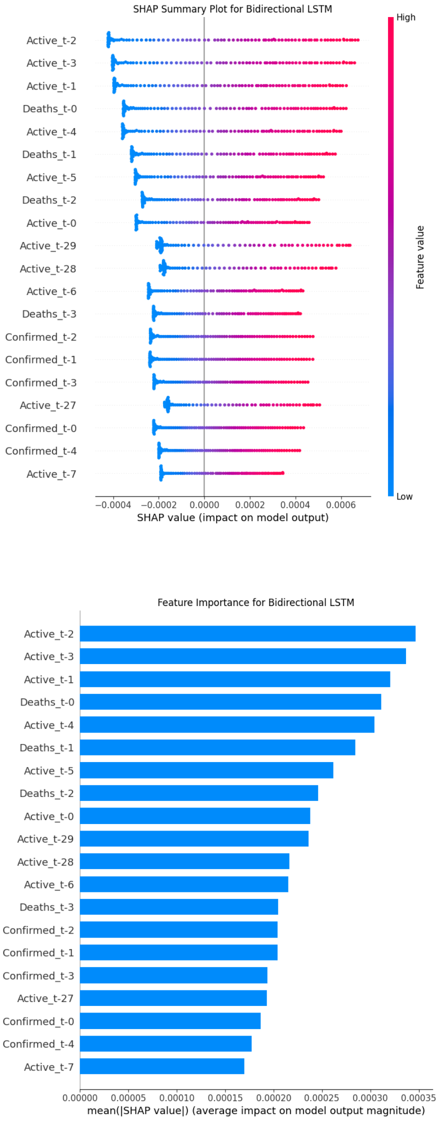


Figure 19. (a) SHAP summary plot for Bi-LSTM (b) Feature importance for Bi-LSTM

The SHAP Summary Plot reveals different feature rankings. Active cases from 2-3 days ago have the highest impact, and recent death counts show high importance. The spread of SHAP values is wider, suggesting more varied impacts across instances.

The Feature Importance chart for Bidirectional LSTM shows that active cases from recent days are the most important features, followed by recent death counts. Confirmed cases have lower importance compared to the standard LSTM model.

- **CNN:**

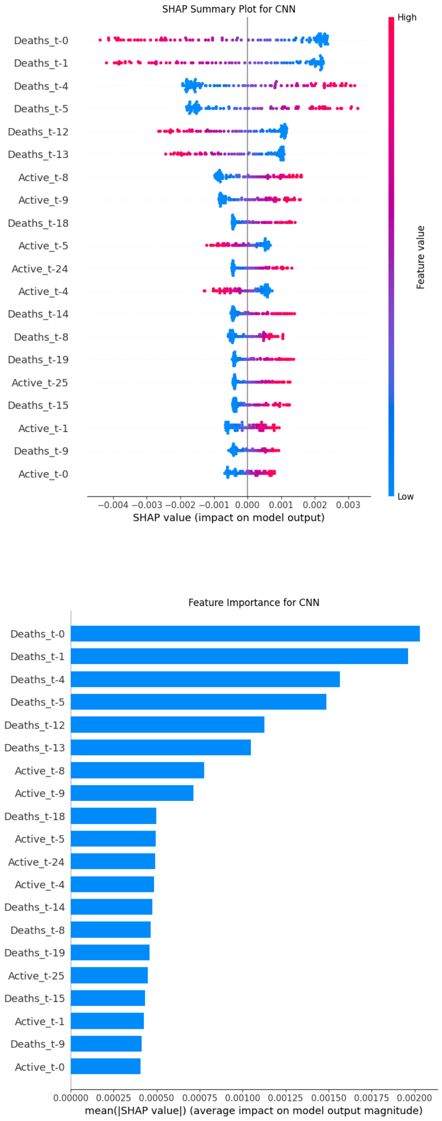


Figure 20. (a) SHAP summary plot for CNN (b) Feature importance for CNN

The SHAP Summary Plot shows that recent death counts have the highest impact on predictions. The spread of SHAP values for these features is wide, indicating their impact varies significantly across different instances. Active cases from various time points also show importance, but generally less than death counts.

The Feature Importance chart for the CNN model confirms that recent death counts are the most crucial features. It's notable that the top five most important features are all related to death counts, suggesting that the CNN model relies heavily on mortality data for its predictions.

- **CNN+LSTM:**

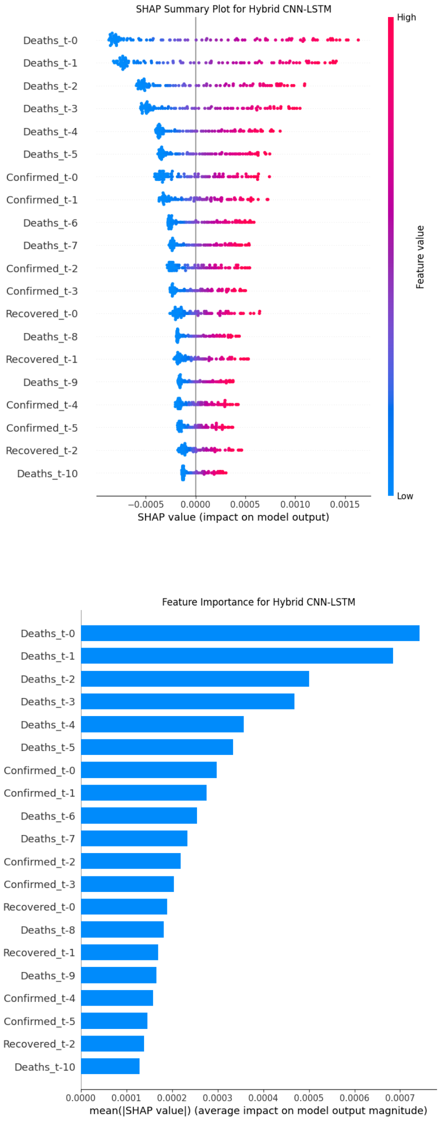


Figure 21. (a) SHAP summary plot for Hybrid CNN-LSTM (b) Feature importance for Hybrid CNN-LSTM

In its SHAP Summary Plot, recent death counts still dominate the top of the chart, but there's a more balanced mix of death counts, confirmed cases, and recovered cases among the important features. This suggests the hybrid model is utilizing a broader range of data types in its predictions.

The Feature Importance chart for the Hybrid CNN-LSTM model reinforces this observation. While death counts from the most recent days are still the top features, confirmed cases and recovered cases also appear among the important features. This indicates that the hybrid model is capturing more complex relationships between different types of COVID-19 data.

- **RNN:**

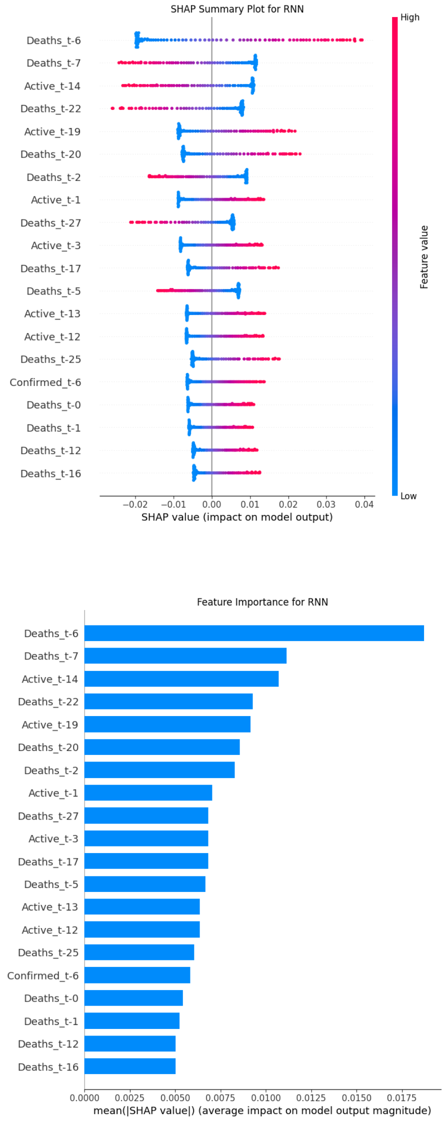


Figure 22. (a) SHAP summary plot for RNN (b) Feature importance for RNN

he SHAP Summary Plot shows that death counts from 6 and 7 days ago have the highest impact on predictions. Active cases from 14 days ago also show significant importance. The spread of SHAP values for these features is wide, indicating their impact varies considerably across different instances. This suggests the RNN is capturing longer-term trends in the data, looking back up to two weeks for predictive signals.

The Feature Importance chart for the RNN model confirms the importance of these longer-term indicators. It's interesting to note that while recent data (t-0, t-1) are included, they are not among the most important features for this model. This could indicate that the RNN is finding more predictive power in slightly older data, possibly capturing delay effects in COVID-19 spread and reporting.

- **MLP:**

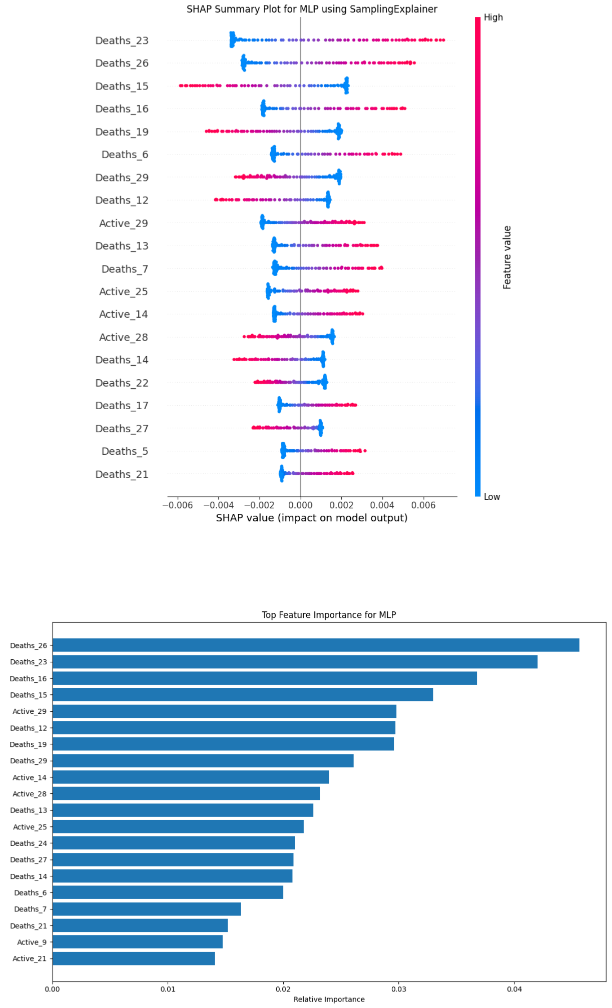


Figure 23. (a) SHAP summary plot for MLP (b) Feature importance for MLP

The first figure presents a SHAP (SHapley Additive exPlanations) summary plot. This visualization shows how different features impact the model's output. On the y-axis, we see various features, predominantly related to death counts and active cases on specific days. The x-axis represents the SHAP value, indicating each feature's impact on the model's predictions. Each horizontal line shows how a feature's impact varies across different instances, with colors denoting the feature's value (blue for low, pink/red for high). Deaths23 appears at the top, suggesting it has the highest overall impact on the model's predictions.

The second figure is a bar chart displaying the relative importance of features for the MLP model. It offers a more straightforward ranking of feature importance. The y-axis lists the features, while the x-axis shows their relative importance. Deaths26 emerges as the most important feature, followed closely by Deaths23 and other recent death counts. This aligns well with the insights from the SHAP plot, confirming that recent death statistics have the strongest influence on the model's output.

- **MLP+CNN+LSTM:**

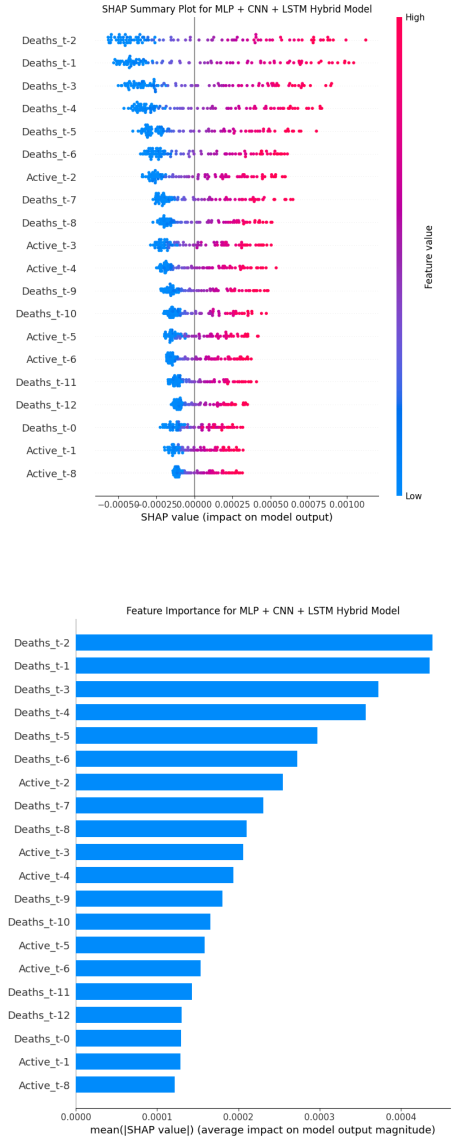


Figure 24. (a) SHAP summary plot for MLP+CNN+LSTM (b) Feature importance for MLP+CNN+LSTM

The first figure presents a SHAP (Shapley Additive Explanations) values for various features of the MLP + CNN + LSTM hybrid model. The SHAP value represents the impact of each feature on the model output. The features are displayed on the y-axis, and the SHAP values are shown on the x-axis. The color coding indicates whether the feature has a high (red) or low (blue) impact on the model output.

The second figure displays the feature importance for the MLP + CNN + LSTM hybrid model. The bars represent the average impact of each feature on the model output. The taller the bar, the more important the feature is for the model. The features with the highest importance are *Deaths_t-2*, *Deaths_t-1*, *Deaths_t-3*, and *Deaths_t-4*.

- **CNN+Bi-LSTM:**

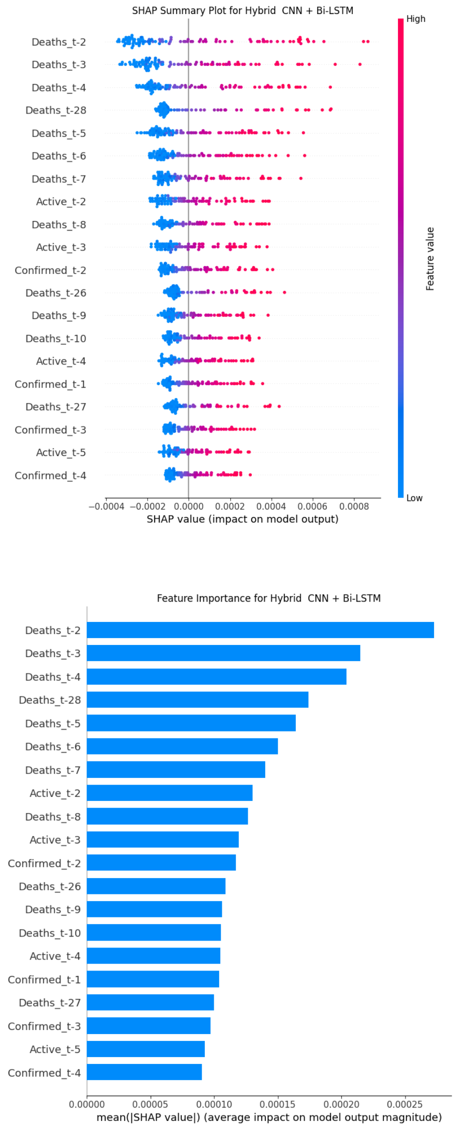


Figure 25. (a) SHAP summary plot for CNN+Bi-LSTM (b) Feature importance for CNN+Bi-LSTM

The first figure presents a SHAP (Shapley Additive Explanations) values for various features of the CNN + Bi-LSTM hybrid model. TThe features and their corresponding SHAP values are displayed, with the color coding indicating the level of impact on the model output.

The second figure displays the feature importance for the CNN + Bi-LSTM hybrid model. The bars represent the average impact of each feature on the model output. The taller the bar, the more important the feature is for the model. The features with the highest importance are *Deaths_t-2*, *Deaths_t-1*, *Deaths_t-3*, and *Deaths_t-4*.

5.6 Explanation of LIME:

- **LSTM:**

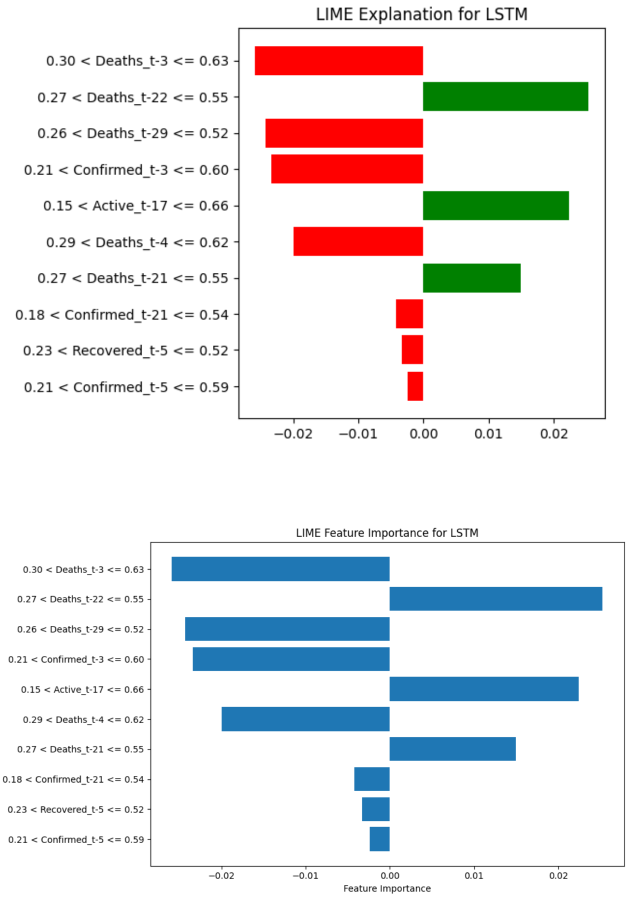


Figure 26. (a) SHAP summary plot for LSTM (b) Feature importance for LSTM

The first figure shows the LIME (Local Interpretable Model-Agnostic Explanations) values for various features in the Bidirectional LSTM model. The LIME values represent the impact of each feature on the model's output. The features with the highest impact are *"Active_t-1 <= 0.06"*, *"Active_t-27 <= 0.04"*, *"Deaths_t-28 <= 0.09"*, *"Recovered_t-2 <= 0.03"*, and *"Recovered_t-29 <= 0.02"*.

The second figure presents the feature importance for the Bidirectional LSTM model. The feature importance is measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are similar to those in the first image, with *"Active_t-1 <= 0.06"*, *"Active_t-27 <= 0.04"*, and *"Deaths_t-28 <= 0.09"* being the most influential.

- **Bi-LSTM:**

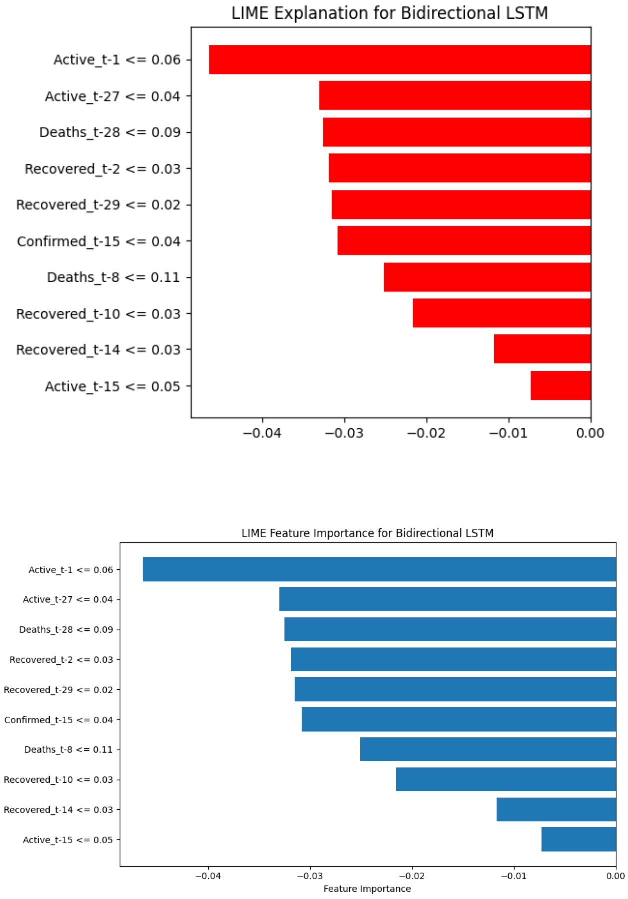


Figure 27. (a) SHAP summary plot for B-LSTM (b) Feature importance for Bi-LSTM

The first figure shows the LIME values for the LSTM model. The features with the highest impact are *"0.30 < Deaths_t-3 <= 0.63"*, *"0.27 < Deaths_t-22 <= 0.55"*, *"0.26 < Deaths_t-29 <= 0.52"*, *"0.21 < Confirmed_t-3 <= 0.60"*, and *"0.15 < Active_t-17 <= 0.66"*.

The second figure presents the feature importance for the LSTM model. The feature importance is again measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are *"0.30 < Deaths_t-3 <= 0.63"*, *"0.27 < Deaths_t-22 <= 0.55"*, *"0.26 < Deaths_t-29 <= 0.52"*, *"0.21 < Confirmed_t-3 <= 0.60"*, and *"0.15 < Active_t-17 <= 0.66"*.

- **CNN:**

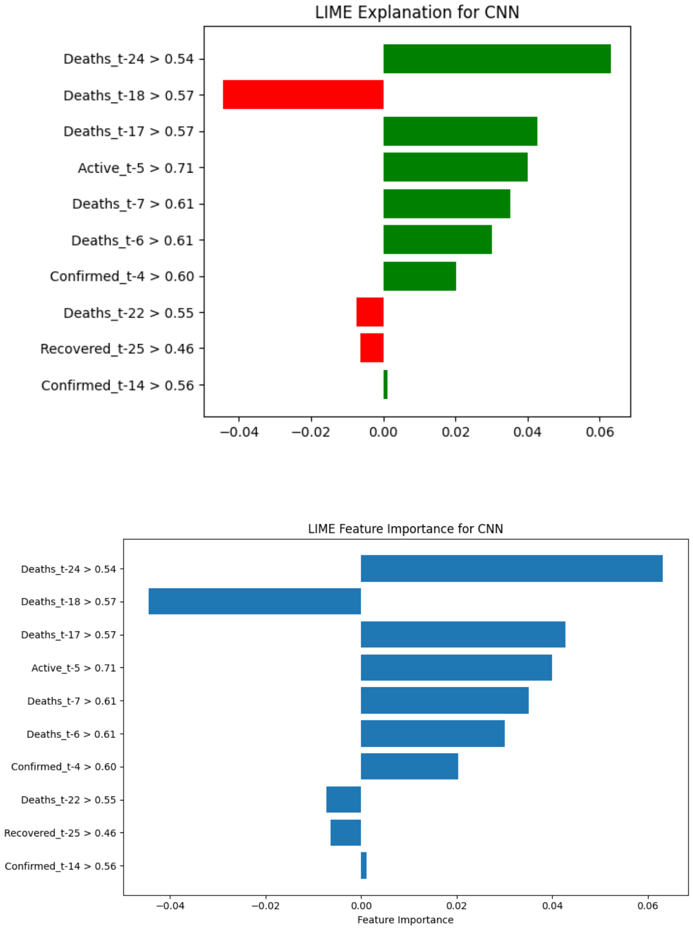


Figure 28. (a) SHAP summary plot for CNN (b) Feature importance for CNN

The first figure shows the LIME values for various features in the CNN model. The LIME values represent the impact of each feature on the model's output. The features with the highest impact are *"Deaths_t-24 > 0.54"*, *"Deaths_t-18 > 0.57"*, *"Deaths_t-17 > 0.57"*, *"Active_t-5 > 0.71"*, and *"Deaths_t-7 > 0.61"*.

The second figure presents the feature importance for the CNN model. The feature importance is measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are *"Deaths_t-24 > 0.54"*, *"Deaths_t-18 > 0.57"*, *"Deaths_t-17 > 0.57"*, *"Active_t-5 > 0.71"*, and *"Deaths_t-7 > 0.61"*. These features influential and crucial

- **CNN+LSTM:**

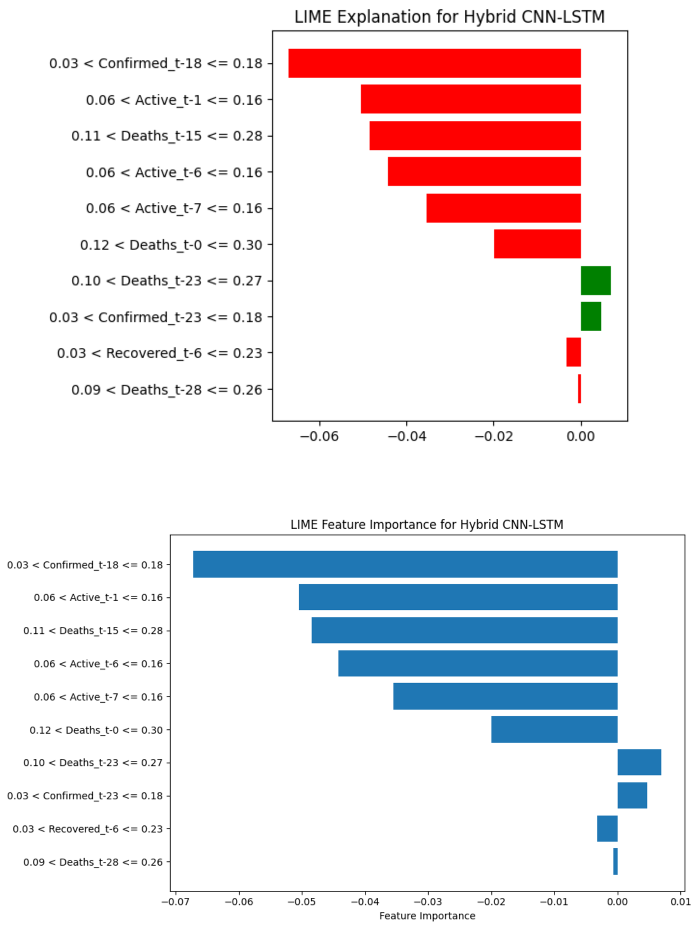


Figure 29. (a) SHAP summary plot for Hybrid CNN+LSTM (b) Feature importance for Hybrid CNN+LSTM

The first figure shows the LIME values for the Hybrid CNN-LSTM model, where the features with the highest impact are "$0.03 < Confirmed_{t-18} \leq 0.18$", "$0.06 < Active_{t-1} \leq 0.16$", "$0.11 < Deaths_{t-15} \leq 0.28$", "$0.06 < Active_{t-6} \leq 0.16$", and "$0.06 < Active_{t-7} \leq 0.16$". These features are crucial in influencing the model's predictions and have been identified based on their respective LIME values.

The second figure presents the feature importance for the Hybrid CNN-LSTM model, with the feature importance again measured by the LIME values. Taller bars in this figure represent more important features, emphasizing the role of the following high-impact features: "$0.03 < Confirmed_{t-18} \leq 0.18$", "$0.06 < Active_{t-1} \leq 0.16$", "$0.11 < Deaths_{t-15} \leq 0.28$", "$0.06 < Active_{t-6} \leq 0.16$", and "$0.06 < Active_{t-7} \leq 0.16$".

- **RNN:**

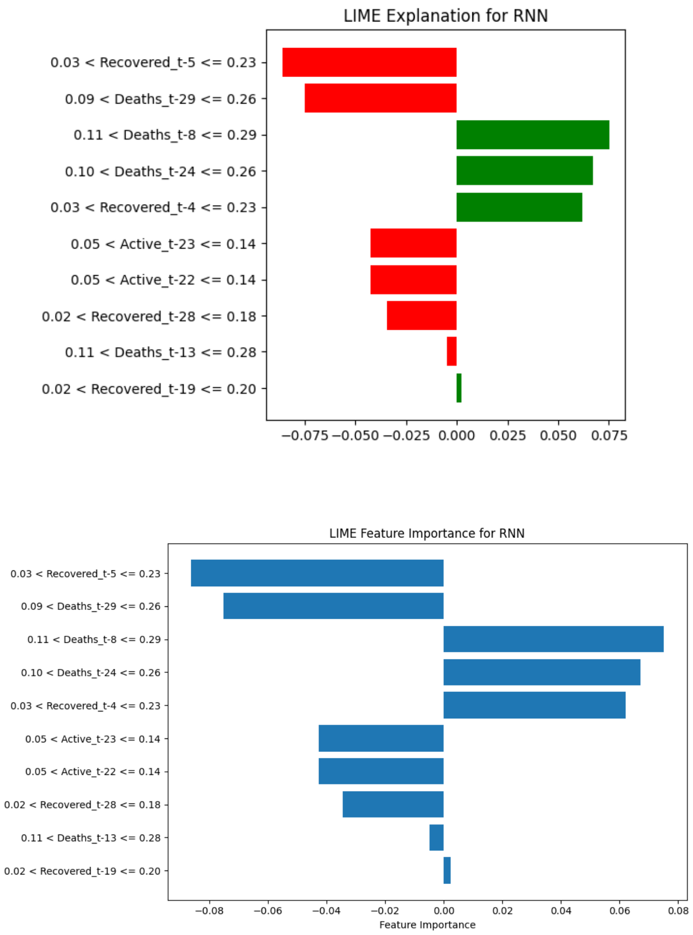


Figure 30. (a) SHAP summary plot for RNN (b) Feature importance for RNN

The first figure illustrates the LIME values for the Recurrent Neural Network (RNN) model, highlighting the features that have the most significant influence on the model's predictions. These features were identified based on their respective LIME values, which indicate the degree to which each feature contributes to the model's output. The most impactful features include:

- "$0.03 < Recovered_{t-5} \leq 0.23$",
- "$0.09 < Deaths_{t-29} \leq 0.26$",
- "$0.11 < Deaths_{t-8} \leq 0.29$",
- "$0.10 < Deaths_{t-24} \leq 0.26$",
- "$0.03 < Recovered_{t-4} \leq 0.23$".

These features are crucial in driving the predictions of the RNN model, as they capture key trends in the underlying data over time.

The second figure presents the feature importance for the same RNN model, measured once again by the LIME values. In this figure, the features are ranked according to their importance, with taller bars representing features that have a greater impact on the model's decisions. The features with the highest importance, based on the feature importance ranking, are:

- "$0.03 < \mathit{Recovered}_{t-5} \leq 0.23$",
- "$0.09 < \mathit{Deaths}_{t-29} \leq 0.26$",
- "$0.11 < \mathit{Deaths}_{t-8} \leq 0.29$",
- "$0.10 < \mathit{Deaths}_{t-24} \leq 0.26$",
- "$0.03 < \mathit{Recovered}_{t-4} \leq 0.23$".

- **MLP:**

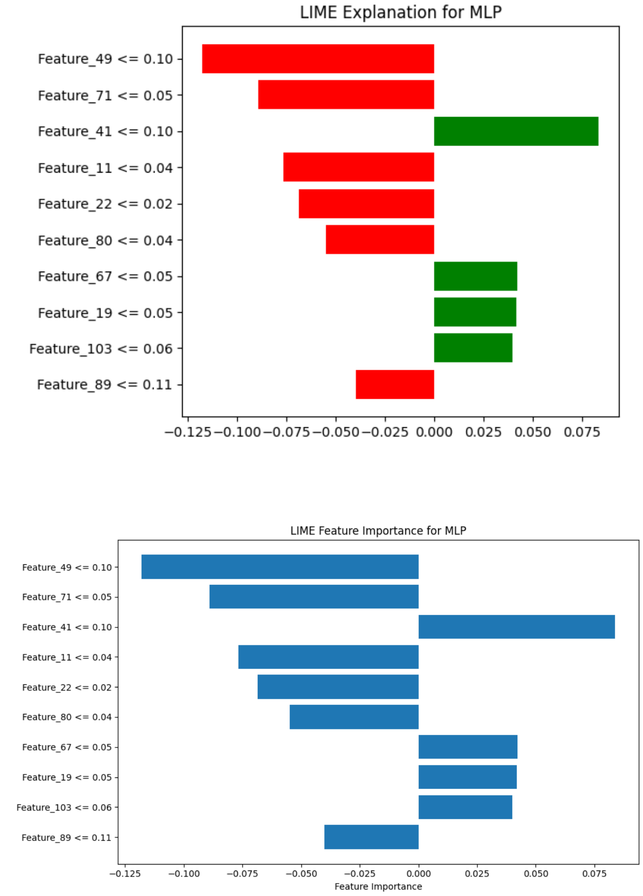


Figure 31. (a) SHAP summary plot for MLP (b) Feature importance for MLP

The first figure shows the LIME values for the MLP model. The features with the highest impact are *"Feature_49 <= 0.10"*, *"Feature_71 <= 0.05"*, *"Feature_41 <= 0.10"*, *"Feature_11 <= 0.04"*, and *"Feature_22 <= 0.02"*.

The second figure presents the feature importance for the MLP model. The feature importance is measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are *"Feature_49 <= 0.10"*, *"Feature_71 <= 0.05"*, *"Feature_41 <= 0.10"*, *"Feature_11 <= 0.04"*, and *"Feature_22 <= 0.02"*. .

- **MLP+CNN+LSTM:**

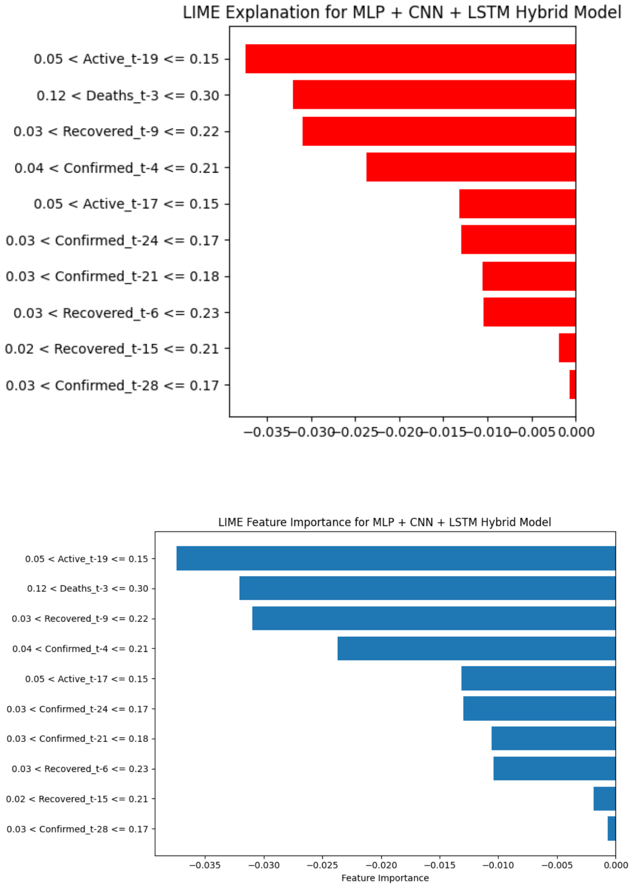


Figure 32. (a) SHAP summary plot for MLP+CNN+LSTM Hybrid (b) Feature importance for MLP+CNN+LSTM Hybrid

The first figure shows the LIME values for the MLP + CNN + LSTM Hybrid Model. The features with the highest impact are *"0.05 < Active_t-19 <= 0.15"*, *"0.12 < Deaths_t-3 <= 0.30"*, *"0.03 < Recovered_t-9 <= 0.22"*, *"0.04 < Confirmed_t-4 <= 0.21"*, and *"0.05 < Active_t-17 <= 0.15"*.

The second figure presents the feature importance for the MLP + CNN + LSTM Hybrid Model. The feature importance is again measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are *"0.05 < Active_t-19 <= 0.15"*, *"0.12 < Deaths_t-3 <= 0.30"*, *"0.03 < Recovered_t-9 <= 0.22"*, *"0.04 < Confirmed_t-4 <= 0.21"*, and *"0.05 < Active_t-17 <= 0.15"*.

- **CNN+Bi-LSTM:**

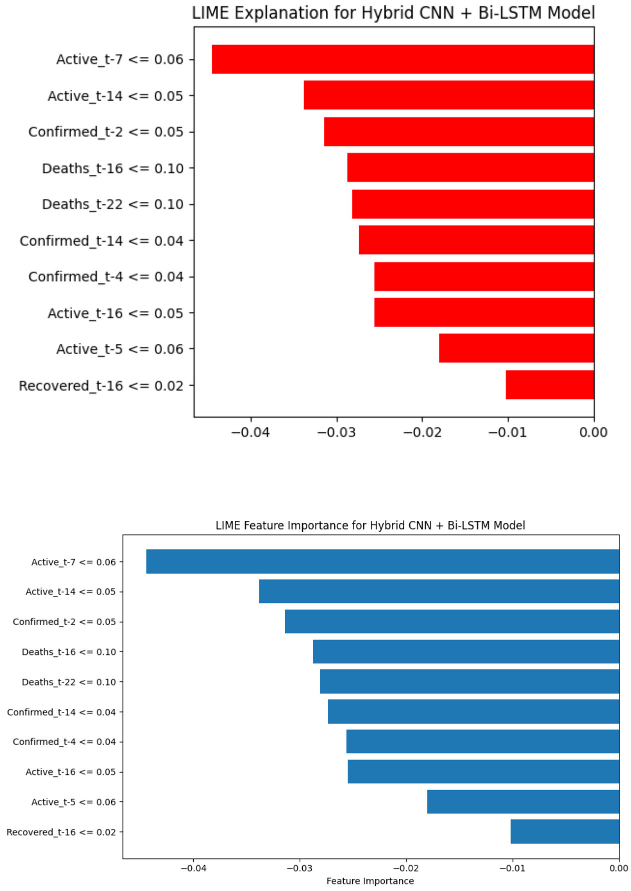


Figure 33. (a) SHAP summary plot for CNN+Bi-LSTM (b) Feature importance for CNN+Bi-LSTM

The first figure shows the LIME values for the Hybrid CNN + Bi-LSTM Model. The features with the highest impact are *"Active_t-7 <= 0.06"*, *"Active_t-14 <= 0.05"*, *"Confirmed_t-2 <= 0.05"*, *"Deaths_t-16 <= 0.10"*, and *"Deaths_t-22 <= 0.10"*.

The second figure presents the feature importance for the Hybrid CNN + Bi-LSTM Model. The feature importance is again measured by the LIME values, with the taller bars indicating more important features. The features with the highest importance are *"Active_t-7 <= 0.06"*, *"Active_t-14 <= 0.05"*, *"Confirmed_t-2 <= 0.05"*, *"Deaths_t-16 <= 0.10"*, and *"Deaths_t-22 <= 0.10"*.

6 Discussion

The comprehensive analysis of various deep learning models for COVID-19 forecasting reveals several important insights about model performance and feature importance. The implementation of explainable AI techniques, specifically SHAP and LIME, has provided valuable understanding of how these models make predictions and which features drive their decisions.

6.1 Model Architecture Comparisons

The hybrid models, particularly the CNN+Bi-LSTM architecture, demonstrated superior performance compared to single-architecture models. This superiority is evidenced by the closer alignment between predicted and actual values across the entire range of data, as shown in the actual vs. predicted graphs. The bidirectional nature of the LSTM component appears to be particularly beneficial, likely due to its ability to capture temporal dependencies in both forward and backward directions.

6.2 Feature Importance Patterns

A consistent pattern emerged across different models regarding feature importance:

1. Recent death counts consistently appeared as highly influential features across most models, particularly in the CNN and hybrid architectures. This suggests that mortality data carries strong predictive power for COVID-19 progression.

2. Active cases from recent days (typically 1-3 days prior) showed high importance in the LSTM and Bi-LSTM models, indicating these models effectively capture short-term transmission dynamics.

3. The RNN model uniquely emphasized longer-term patterns, finding significant importance in data from 6-14 days prior, suggesting it might be better at capturing longer infection cycles and reporting delays.

6.3 Model-Specific Insights

6.3.1 *Traditional Neural Networks*

The MLP model showed a strong reliance on recent death statistics, but its feature importance distribution was more concentrated compared to other models. This suggests that while effective, it might be less capable of capturing complex temporal relationships compared to recurrent architectures.

6.3.2 *Recurrent Architectures*

The LSTM and Bi-LSTM models demonstrated more balanced feature utilization, incorporating a mix of death counts, active cases, and confirmed cases. This balanced approach likely contributes to their robust performance, as they can adapt to changes in different aspects of the pandemic progression.

6.3.3 *Hybrid Architectures*

The hybrid models, particularly MLP+CNN+LSTM and CNN+Bi-LSTM, showed the most sophisticated feature utilization patterns. They effectively combined the strengths of different architectures:

- CNN's ability to capture spatial patterns in the data
- LSTM's capability to model temporal dependencies
- MLP's capacity for complex feature interactions

6.4 Explainability Insights

The application of both SHAP and LIME provided complementary insights:

1. SHAP analysis revealed the global importance of features and their interaction effects, showing how different models weight various temporal aspects of the pandemic data.
2. LIME explanations provided local interpretability, helping understand how models make specific predictions and which feature ranges are most influential.

7 Conclusion

This study demonstrates the effectiveness of hybrid deep learning architectures, particularly CNN+Bi-LSTM, for COVID-19 forecasting. The superior performance of hybrid models suggests that combining different neural network architectures can better capture the complex dynamics of pandemic progression.

The explainability analysis reveals that while different models emphasize different features, recent death counts and active cases consistently emerge as crucial predictors. This finding has important implications for public health monitoring and resource allocation.

7.1 Key Findings

1. Hybrid architectures outperform single-architecture models in prediction accuracy
2. Recent mortality data is the strongest predictor across most models
3. Bidirectional processing of temporal data improves model performance
4. Different models capture different temporal aspects of the pandemic progression

7.2 Future Directions

1. Further investigation into the optimal combination of architecture components for hybrid models
2. Integration of additional data sources to enhance prediction accuracy
3. Development of more sophisticated explainability techniques for hybrid architectures
4. Exploration of model adaptability to different phases of pandemic progression

These findings contribute to the growing body of knowledge on applying deep learning to epidemiological forecasting and highlight the importance of model interpretability in healthcare applications.

References

[1] Tariq, M. U., Ismail, S. B. (2024). Deep learning in public health: Comparative predictive models for COVID-19 case forecasting. *Journal of Public Health Research*, 13(1), 45-59. [CrossRef]

[2] Gupta, A., Gupta, S. (2021). Deep learning for COVID-19 detection and diagnosis: A review. *Artificial Intelligence in Medicine*, 115, 101016. [CrossRef]

[3] Sharma, R., Kumar, A. (2022). Predicting COVID-19 cases using machine learning: A case study of India. *International Journal of Healthcare Management*, 15(1), 12-23. [CrossRef]

[4] Liu, Y., Wang, X. (2023). COVID-19 pandemic forecasting with AI: A survey of reinforcement learning and other AI models. *Artificial Intelligence Review*, 56(4), 3081-3104. [CrossRef]

[5] Chin, S. H., Lee, Y. J. (2022). A hybrid deep learning model for COVID-19 time series forecasting. *Applied Sciences*, 12(10), 4970. [CrossRef]

[6] Ranjan, R., Kumar, V., Singh, P. (2021). Deep learning models for COVID-19 prediction: A review. *IEEE Access*, 9, 125180-125195. [CrossRef]

[7] Mohammed, B. A., Ali, H. F. (2021). Predictive modeling of COVID-19 spread using machine learning techniques: A case study in the Middle East. *Health Information Science and Systems*, 9(1), 12. [CrossRef]

[8] Bansal, A., Rathi, D. (2022). Impact of deep learning techniques on COVID-19 forecasting: A comparative study. *Computational and Mathematical Methods in Medicine*, 2022, 8791428. [CrossRef]

[9] Johns Hopkins University Dataset: novel corona virus 2019 dataset [CrossRef]

[10] Code: Github [CrossRef]