Hybrid Model for Pandemic Forecasting

By Dennis Reyes

Problem Statement

Pandemics profoundly impact public health, economies, and social behavior worldwide. Accurately forecasting future case trends provides valuable insights for policymakers and healthcare professionals, helping them anticipate surges, allocate resources, and implement timely interventions.

XGBoost + GRU Hybrid Model

What is XGBoost?

- Extreme Gradient Boosting (XGBoost) is a powerful tree-based model that excels at:
  - Identifying key features from structured tabular data
  - Generating interpretable feature importance rankings
  - Rapidly processing large datasets with optimized performance

How I Used XGBoost in this Hybrid Model

- I trained XGBoost using location-based features and lag variables to detect important contributors to outbreak spread. By leveraging its structured learning, XGBoost identifies which geographic factors drive case trends, helping refine predictions.

What is GRU?

- Gated Recurrent Units (GRU) are a type of recurrent neural network (RNN) designed to:
  - Capture sequential dependencies in time-series data
  - Improve long-term trend tracking while filtering irrelevant signals
  - Smooth short-term fluctuations in predictions

How I Used GRU in this Hybrid Model

- I trained GRU on time-series sequences to learn outbreak progression patterns based on historical cases. Unlike XGBoost, GRU adapts dynamically to changes over time, helping refine final forecasts.

Data Sources

Google Covid-19 Open Data (Data from 2020 - 2022) ~12 Million Records

- Epidemiology Dataset
  - New confirmed cases
  - Cumulative cases
  - New Deceased
  - Cumulative Deceased
- Geography Dataset
  - Latitude/Longitude
  - Location Key

Data Preprocessing

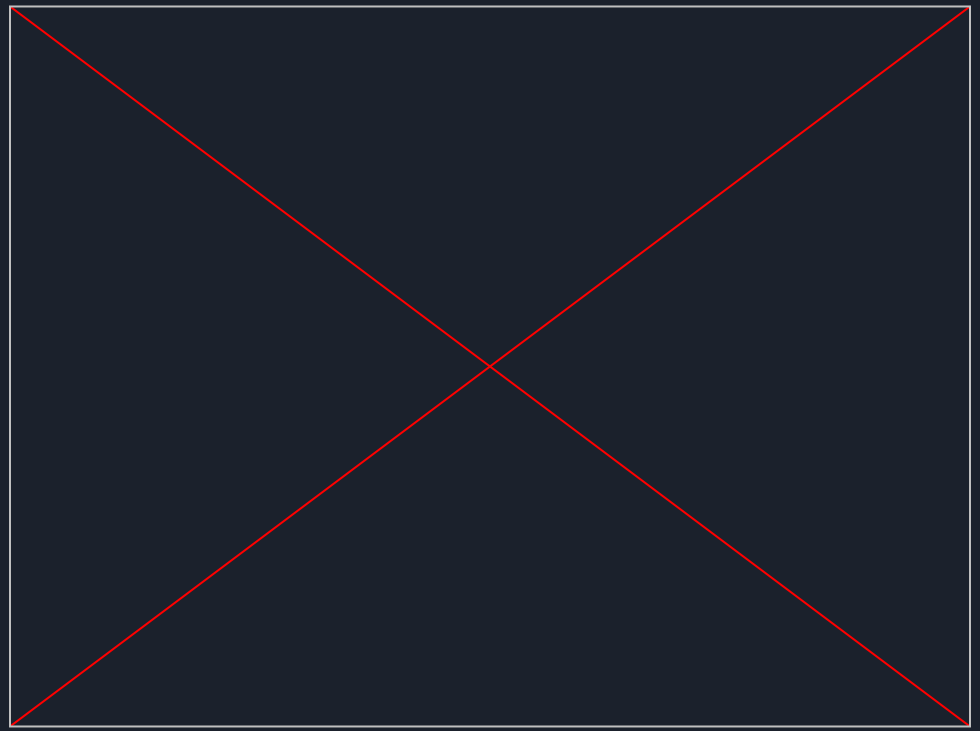
Data Preprocessing

- Lag-based features - Previous Day and Previous Week new confirmed cases.
  - Adding lags allows powerful transformations like moving averages, rolling statistics, or trend shifts
  - Time-series data has inherent patterns lag features help models recognize recurring trends
- Scaling features
  - Avoids dominance of Large-Scale Features
- Handle missing data
  - Removed missing data (new confirmed cases, latitude, longitude)
- Removed erroneous data
  - Negative new confirmed cases

Challenges

- Handling Outliers - Most likely not all countries accurately reporting new cases
- Data Quality - Countries started report cases at different time intervals, so choosing a time window where the data accurately represents the actual trend.

2022 US Weekly New Confirmed Cases



Analytics Methodology

Why combining XGBoost and GRU improves forecasting?

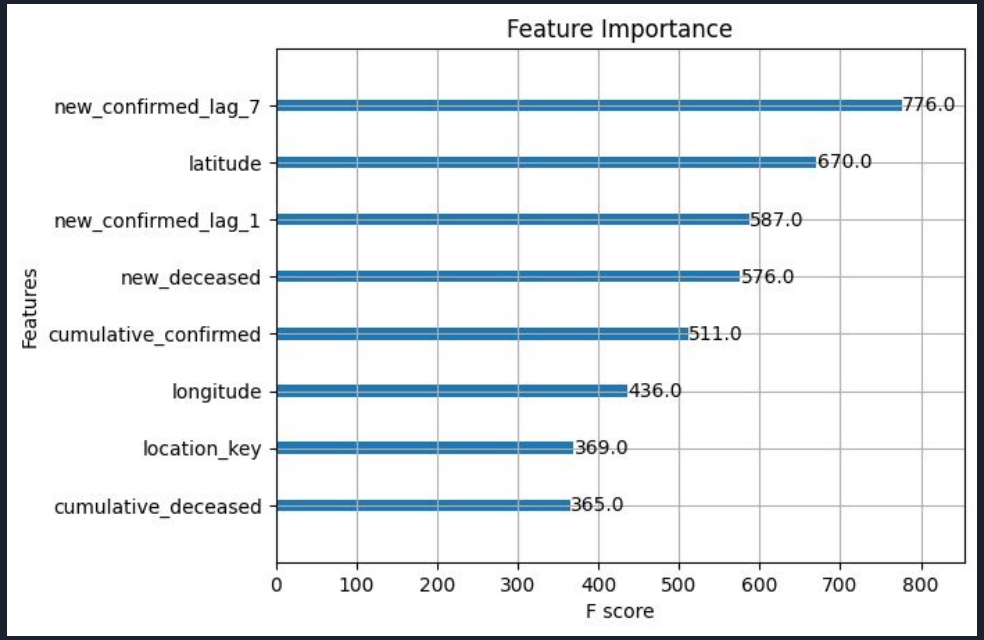
- Captures both Structured & Temporal Dependencies
  - XGBoost excels at handling tabular data, identifying complex nonlinear relationships
  - GRU specializes in sequential dependencies, understanding long-term trends in time-series data
  - Hybrid approach learns both feature-based patterns AND temporal correlations

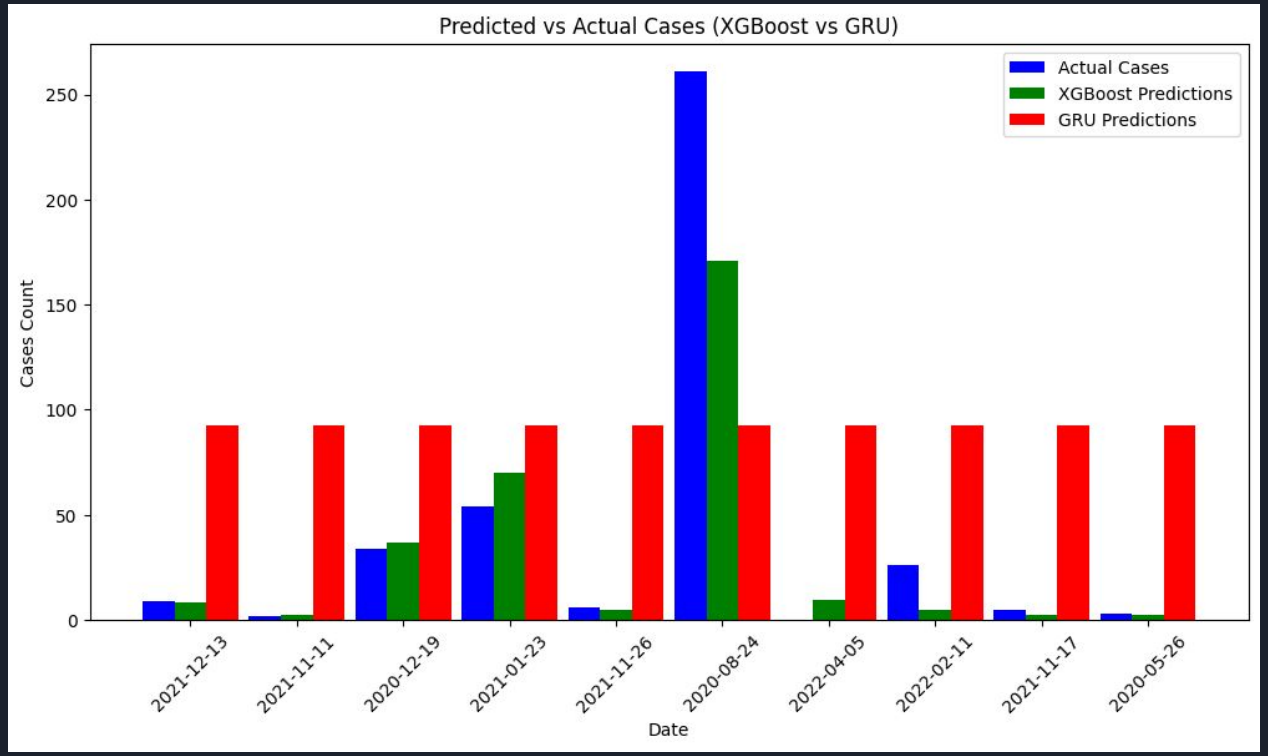
XGBoost: Handles structured learning from static features

- Excels at handling feature importance, including static indicators (e.g., region, lag, prior cases)
- Outputs interpretable importance rankings, helping identify key drivers of outbreak trends and pinpoints which geographic factors influence disease spread, helping guide feature selection for GRU.

GRU: Captures time-series dependencies (sequential trends)

XGBoost Important Features




Predicted vs Actual Cases (XGBoost vs GRU)
Actual Cases
XGBoost Predictions
GRU Predictions
Cases Count
0
50
100
150
200
250
2021-12-13
2021-11-11
2020-12-19
2021-01-23
2021-11-26
2020-08-24
2022-04-05
2022-02-11
2021-11-17
2020-05-26
Date

Hybrid Model Flow

1. Data Loading & Preprocessing
    a. Clean missing values, normalize data
    b. Handle time-series formatting for sequential models
    c. Convert categorical variables (e.g., locations) into usable features
2. Feature Engineering
    a. Create lag-based features to capture historical trends
    b. Apply scaling to ensure feature consistency across models
3. Train XGBoost Model with Location & Lag-Based Features
    a. Train XGBoost using structured inputs (geographic factors, lag variables)
    b. Identify feature importance to understand leading predictors
    c. Optimize hyperparameters (`learning_rate`, `max_depth`, `n_estimators`)
4. Train GRU Using Sequential Trends
    a. Construct GRU layers to model time dependencies
    b. Define sequence length to balance memory retention vs computational efficiency
    c. Adjust dropout & batch size to improve stability
5. Ensemble Predictions (XGBoost + GRU Fusion)
    a. Blend both outputs to improve forecast accuracy
    b. Assign optimal ensemble weights based on validation performance
6. Evaluate Model Performance
    a. Compare accuracy metrics (RMSE, MAE, R^2)
    b. Validate robustness using residual plots & loss curves
7. Visualize Predictions on Geographic Maps
    a. Display forecasts with Folium
    b. Animate spatial case spread over time

Model Impact

Why is this model significant?

- Captures Temporal Dependencies → GRU detects outbreak trends over time
- Leverages Structured Features → XGBoost identifies key geographical factors
- Balances Ensemble Learning → Improves accuracy and reduces error volatility

How can this be used for better disease prevention?

- Early Detection & Response Planning
  - Predicts outbreaks before they escalate, enabling proactive interventions
  - Helps authorities allocate medical resources efficiently
- Public Health Awareness Campaigns
  - Forecasted trends help design targeted programs
  - Allows better communication of risk factors to communities

Techniques Used to Handle Large Datasets

- GPU Acceleration with **tf.config.optimizer.set_jit(True)**
  - Enables XLA (Accelerated Linear Algebra) optimization
    - Speeds up TensorFlow computations
    - Reduces memory footprint by compiling operations more efficiently
  - Helps optimize GRU computations when handling long sequences
- Mini-Batch Processing for GRU
  - Breaks dataset into smaller chunks → Prevents memory overload
  - Prevents crashes when training deep models like GRU on large time-series data
- Parallel Data Loading for Faster Training
  - Speeds up input pipeline processing
  - Improves efficiency when preparing batches for training
  - Ensures real-time data loading and prevents I/O delays
- Dask
  - Integrates with Pandas and numPy
  - Uses lazy loading and computes when it needs to

Thank You / Q&A