Multiplicative Gate State Space Models with Skip-Net for High Accuracy COVID-19 Time-Series Prediction

Krung Sinapiromsaran, Supakit Sroynam
Faculty of Science-Department of Mathematics and Computer Science,
Chulalongkorn University, Bangkok, Thailand

***Abstract*—The rapid propagation of the COVID-19 pandemic has placed unprecedented strain on global healthcare systems, creating an urgent need for accurate forecasting to optimize resource allocation and policy implementation. However, the highly non-linear and chaotic behavior of infection rates poses significant challenges for traditional statistical and standard deep learning models. This study proposes the Multiplicative Gating State Space Model with skip connection (MG-SSM-s), a novel architecture designed to capture complex temporal dependencies in epidemiological time series. Drawing inspiration from partial autocorrelation, the model extends modern State Space Models (SSMs) by incorporating a learnable multiplicative side channel to dynamically modulating input processing. We evaluated the efficacy of MG-SSM-s using the Google COVID-19 Open Data repository, analyzing daily confirmed cases across 40 countries, including major epicenters such as the USA, India, and Brazil. Using a 30-day look-back window, the proposed model was benchmarked against four baseline architectures: LSTM, Bi-LSTM, GRU, and standard SSM. Performance verification based on Mean Squared Error (MSE) demonstrates that MG-SSM-s outperforms all deep learning baselines and achieves competitive accuracy with a tuned ARIMA model, demonstrating comparable statistical performance to the latter. These results highlight the framework's robustness and potential as a versatile tool for time-series forecasting.**

Keywords*—State Space Model; COVID-19; time-series forecast; univariate time-series

I. Introduction

The COVID-19 pandemic exposed critical vulnerabilities in public health forecasting infrastructure. As infection rates accelerated, healthcare systems faced severe resource constraints, particularly in managing hospital bed capacity, ventilator supplies, and staffing shortages. A primary barrier to effective management was the volatility of daily case counts, which severely hindered proactive capacity planning. Consequently, developing robust methods to forecast epidemiological time-series has become essential for long-term pandemic preparedness.

COVID-19 transmission dynamics exhibit highly non-linear, non-stationary behaviors that challenge traditional statistical modeling. Infection rates rely on multiplicative interactions between transmission rates, susceptible population densities, and contact frequencies [1]. Moreover, sudden shocks, including policy interventions, viral variants, and behavioral shifts, trigger abrupt regime changes that standard time-series models often misclassify as noise [2]. While machine learning architectures have advanced time-series forecasting, simultaneously capturing non-linear multiplicative dynamics, high-frequency volatility, and long-range temporal dependencies remains an open challenge. To address these limitations, we propose the Multiplicative Gating State Space Model with skip connection (MG-SSM-s), an architecture designed specifically for volatile epidemiological data.

MG-SSM-s extends modern State Space Models with a learnable multiplicative gating mechanism and skip connections, achieving a separation between global trends and volatility. The design is inspired by autocorrelation concepts, where multiplicative product interactions enable the model to capture dependencies across multiple temporal lags. This process is similar to how partial autocorrelation identifies significant historical correlations, while skip connections preserve sudden shocks that linear state transitions would otherwise attenuate. By maintaining parallel pathways for trend and volatility, MG-SSM-s retains both long-range awareness and sensitivity to sudden shifts critical for pandemic forecasting.

A. Contributions

The primary contributions of this research are fourfold:

- Novel Hybrid Architecture: MG-SSM-s extends standard State Space Models by incorporating a learnable multiplicative gating mechanism and skip connection, enabling non-linear expression of temporal dependencies.
- Theoretically-Inspired Design: The multiplicative gate is inspired by autocorrelation concepts, enabling the model to dynamically identify and weight relevant temporal patterns, with an explicit skip connection preserving immediate input shocks.
- Comprehensive Evaluation: We evaluate MG-SSM-s on COVID-19 case forecasting datasets from 40 countries against established baselines, including LSTM, Bi-LSTM, GRU, and vanilla SSMs, demonstrating superior performance on non-stationary epidemic data.
- Practical Applicability for Epidemic Forecasting: The architecture is designed for univariate time series forecasting with linear computational complexity, making it suitable for real-time deployment across diverse healthcare systems during active pandemics.

The remainder of this study is organized as follows: Section II reviews related literature in epidemiological forecasting and State Space Models, highlighting the research gaps motivating MG-SSM-s. Section III defines the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). Section IV details the architectural design of the proposed model, and Section V evaluates its performance on COVID-19 datasets. Finally, Section VI concludes the study and outlines future research directions.

II. Related Work

A. Traditional Statistical Methods in Epidemic Forecasting

Early COVID-19 forecasting efforts relied heavily on statistical linear models, most notably the Autoregressive Integrated Moving Average (ARIMA) and its variants [3]. ARIMA models operate by transforming non-stationary data into a stationary series through differencing, then predicting future values based on a linear combination of past errors and past values. While ARIMA provides mathematical interpretability and works well for simple linear trends, it fundamentally struggles with the chaotic volatility and non-linear dynamics of pandemic data. Its reliance on linear assumptions and stationarity makes it ill-suited for capturing the abrupt infection waves and complex environmental interactions characteristic of COVID-19 propagation.

In a comprehensive analysis, it was demonstrated that meteorological factors, such as temperature, relative humidity, and wind speed, exhibit nonlinear correlations with daily new COVID-19 cases across different regions [4]. This nonlinearity persists even when controlling for confounding factors, indicating that pandemic transmission follows non-linear mechanistic principles that linear statistical models cannot express.

B. Deep Learning Approaches for Epidemic Time-Series Forecasting

To address ARIMA's limitations, Deep Learning (DL) approaches were widely adopted for COVID-19 prediction [5], with the Long Short-Term Memory (LSTM) network developed by [6] emerging as a standard baseline. LSTMs overcome limitations of standard RNNs through a memory cell state modulated by three distinct gates (input, forget, output), allowing selective retention or discarding of information over long sequences [7]. In COVID-19 forecasting, LSTMs have demonstrated the ability to learn non-linear infection trends that ARIMA misses [8]; however, they often struggle to capture very sharp, high-frequency changes (sudden spikes from variants or policy shifts) due to the slow adaptation of recurrent cell states.

1) Bidirectional LSTMs (Bi-LSTM): Bidirectional LSTM architectures, such as those introduced by [9], enhance context awareness by processing data bidirectionally. By concatenating hidden states from both directions, the model gains temporal features surrounding each time step. While bidirectional flow improves interpolation and noise smoothing, it substantially increases computational costs and does not fundamentally solve the modeling of immediate, shock-like volatility in real-time forecasting scenarios where future data is unavailable.

2) Gated Recurrent Units (GRU): The Gated Recurrent Unit, proposed by [10], offers a more computationally efficient alternative by merging the cell state and hidden state and combining input and forget gates into a single update gate. This simplification enables faster training and convergence, particularly advantageous when modeling multiple countries simultaneously. However, the GRU shares a fundamental limitation with LSTM: reliance on dense recurrence that becomes opaque and difficult to optimize for extremely long historical contexts, often leading to performance degradation when the look-back window is extended significantly.

C. State Space Models for Time-Series Forecasting

Recently, State Space Models (SSMs) have gained prominence as a powerful alternative to Transformers and RNNs for time-series tasks. The Structured State Space Sequence (S4) model introduced by [11], [12] pioneered this resurgence by combining continuous-time representations from control systems with deep learning. S4 utilizes a low-rank HiPPo matrix initialization to compress infinite-context history into a finite state, enabling efficient handling of extremely long-range dependencies via the convolution theorem during training. However, while S4 excels at capturing global trends over long horizons, its linear time-invariant (LTI) nature can be too rigid for non-stationary pandemic systems undergoing rapid structural changes, such as virus mutations or abrupt policy shifts.

More recently, an advancement to the SSM paradigm was introduced via a selection mechanism that allows system matrices to be functions of the input [13]. This formulation makes the model time-varying and input-selective, which enables it to distinguish between relevant and irrelevant information at each time step, thereby solving the content-aware limitations of S4.

However, directly deploying an unconstrained vanilla Mamba block onto raw, univariate time-series data introduces architectural vulnerabilities. Because standard selective SSMs are optimized to compress dense, multi-channel representations, applying them to low-dimensional, short-horizon vectors yields severe over-parameterization and rapid latent state saturation.

III. Background Knowledge

A. Autocorrelation and Partial Autocorrelation

Time-series forecasting relies fundamentally on analyzing the dependency of a variable on its past values. Two statistical measures are paramount in characterizing these temporal dependencies: the Autocorrelation Function (ACF) and the Partial Autocorrelation Function (PACF).

1) Autocorrelation Function (ACF): The ACF quantifies the linear relationship between a time-series X_t and a lagged version of itself X_{t-k}, accounting for both direct influence and indirect influence propagated through intermediate time steps $(X_{t-1}, \ldots, X_{t-k+1})$. Formally, for a stationary time-series, the autocorrelation coefficient at lag k, denoted as ρ_k, is defined as:

$$\rho_k = \frac{\text{Cov}(X_t, X_{t-k})}{\sqrt{\text{Var}(X_t)\text{Var}(X_{t-k})}} = \frac{\gamma_k}{\gamma_0} \tag{1}$$

where, γ_k is the autocovariance at lag k and γ_0 is the variance. In the context of deep learning, Recurrent Neural Networks (RNNs) naturally model this accumulated dependency through their hidden state updates.

2) Partial Autocorrelation Function (PACF):: While the ACF formulation in Eq. (1) captures total correlation, it fails to isolate the specific contribution of X_{t-k} independent of intervening lags. The PACF addresses this limitation by measuring the correlation between X_t and X_{t-k} after removing the linear dependence explained by the intermediate lags $\{X_{t-1}, \ldots, X_{t-k+1}\}$. Mathematically, the partial autocorrelation ϕ_{kk} at lag k is expressed as the correlation of the residuals in Eq. (2):

$$\phi_{kk} = \text{Corr}(X_t - \hat{X}_t, X_{t-k} - \hat{X}_{t-k}) \tag{2}$$

where, $\hat{X}_t$ and $\hat{X}_{t-k}$ represent the linear projections of X_t and X_{t-k}, respectively, onto the space spanned by the intermediate set $\{X_{t-1}, \ldots, X_{t-k+1}\}$.

The distinction between the total correlation in Eq. (1) and the isolated dependency in Eq. (2) inspires the proposed MG-SSM-s architecture. Standard State Space Models capture past context through recursive state equations. However, to capture dependencies, where X_{t-k} directly influences X_t without intermediate smoothing, a mechanism analogous to the PACF is required. The proposed skip-connection and multiplicative gating mechanism serve this purpose by providing a direct channel to model specific lagged dependencies that standard recurrence might otherwise oversmooth.

B. State Space Model (SSM)

The State Space Model (SSM) serves as a foundational framework for modeling dynamic systems in control theory and signal processing. In the context of deep learning, it allows for the parameterization of dependencies in sequence data through a latent state representation. A Continuous-Time Latent State Model defines a system, where an input signal $u(t) \in \mathbb{R}^D$ drives a latent state $x(t) \in \mathbb{R}^N$, which is subsequently projected to an observable output $y(t) \in \mathbb{R}^D$ [14].

Mathematically, the dynamics of this system are governed by the linear ordinary differential equations (ODEs) defined in Eq. (3) and Eq. (4):

$$x'(t) = \mathbf{A}x(t) + \mathbf{B}u(t) \tag{3}$$
$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \tag{4}$$

The system is parameterized by the state evolution matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ in Eq. (3), which governs the internal dynamics of the hidden state, and the input projection matrix $\mathbf{B} \in \mathbb{R}^{N\times D}$. The latent state is mapped to the observed data via the output projection matrix $\mathbf{C} \in \mathbb{R}^{D\times N}$ in Eq. (4). The final term, $\mathbf{D} \in \mathbb{R}^{D\times D}$, represents a direct feedthrough connection from input to output. In modern structured state space architectures, such as S4 and Mamba, this feedthrough term is frequently omitted from the core ODE dynamics and instead implemented as a separate, learnable residual connection to facilitate optimization.

1) Discretization and Recurrent Form: To facilitate the processing of discrete data sequences, the continuous-time dynamics must be discretized. Modern SSM architectures typically adopt the Zero-Order Hold (ZOH) principle, which posits that the input signal $u(t)$ remains constant within the sampling interval Δ. The discretization process transforms the continuous parameters $(\mathbf{A}, \mathbf{B})$ into their discrete counterparts $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$. The discrete state transition matrix is defined via Eq. (5):

$$\bar{\mathbf{A}} = \exp(\Delta\mathbf{A}) \tag{5}$$

While the input projection matrix $\bar{\mathbf{B}}$ is derived via integration over the sampling period, as shown in Eq. (6):

$$\bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B} \approx \mathbf{A}^{-1}(\bar{\mathbf{A}} - \mathbf{I})\mathbf{B} \tag{6}$$

Consequently, the system can be expressed as a linear recurrence via Eq. (7) and Eq. (8), highlighting the structural parallel between discretized SSMs and Recurrent Neural Networks (RNNs):

$$x_t = \bar{\mathbf{A}}x_{t-1} + \bar{\mathbf{B}}u_t \tag{7}$$
$$y_t = \mathbf{C}x_t \tag{8}$$

In standard Linear Time-Invariant (LTI) SSMs (e.g., S4), the matrices $\bar{\mathbf{A}}$ and $\bar{\mathbf{B}}$ in Eq. (7) are constant across all time steps. This stationarity allows for efficient parallel processing but limits the model's ability to adapt its dynamics to specific input tokens. Recent advancements in structured state-space models, most notably Mamba [13], address the limitations of static content compression by introducing a selection mechanism. By defining the system matrices $(\mathbf{A}, \mathbf{B}, \Delta)$ as functions of the input, these models effectively transform the architecture from an LTI system into a time-varying linear recurrence. While this input-selectivity significantly enhances the model's ability to distinguish relevant context within long sequences, it fundamentally alters the computational dynamics, as the loss of time-invariance.

IV. Methodology

A. Model Structure

The Multiplicative Gating State Space Model with skip connection (MG-SSM-s) is defined by the system of equations parameterized by the learnable set $(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{J})$ in Eq. (9) to Eq. (11):

$$h_t = \mathbf{A}h_{t-1} + \mathbf{B}x_t \tag{9}$$
$$g_t = (\mathbf{E}g_{t-1}) \odot \tilde{x}_t + \mathbf{F}x_t \tag{10}$$
$$y_t = \mathbf{C}h_t + \mathbf{D}x_t + \mathbf{J}g_t \tag{11}$$

where, $\odot$ denotes the element-wise (Hadamard) product. The term $\tilde{x}_t \in \mathbb{R}^m$ represents the scalar input x_t projected into the latent gate dimension via the all-ones vector $\mathbf{1}_m \in \mathbb{R}^m$, such that $\tilde{x}_t = x_t \cdot \mathbf{1}_m$. This operation ensures that the

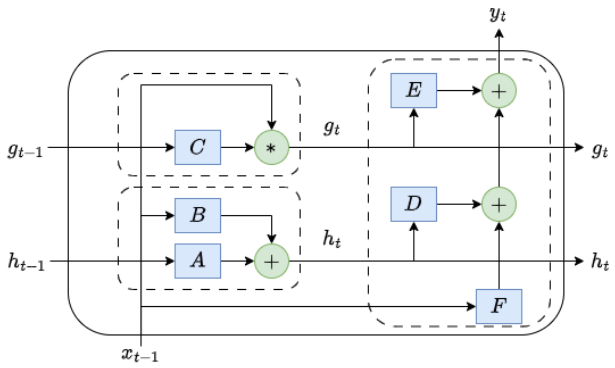

Fig. 1. Architecture of the Multiplicative Gating State Space Model with skip connection (MG-SSM-s).

scalar input modulates each channel of the latent gate state g_t uniformly. Fig. 1 presents the architecture of the Multiplicative Gating State Space Model with skip connection (MG-SSM-s).

The model processes a univariate input sequence $X = (x_1, \ldots, x_T) \in \mathbb{R}^T$ to produce an output sequence $Y = (y_1, \ldots, y_T) \in \mathbb{R}^T$. The internal memory is decoupled into two distinct pathways: a main linear state $h_t \in \mathbb{R}^l$ (where, l is the latent dimension) evolving via Eq. (9), and a multiplicative gate state $g_t \in \mathbb{R}^m$ (where, m is the gate dimension) governed by Eq. (10).

The linear state h_t in Eq. (9) evolves via standard LTI dynamics using the transition matrix $\mathbf{A} \in \mathbb{R}^{l\times l}$ and input projection $\mathbf{B} \in \mathbb{R}^{l\times 1}$. The gated state g_t in Eq. (10) introduces non-linearity through a two-step mechanism: first, the previous state g_{t-1} is transformed by $\mathbf{E} \in \mathbb{R}^{m\times m}$ and modulated element-wise by the input; second, a direct skip connection parameterized by $\mathbf{F} \in \mathbb{R}^{m\times 1}$ is added. This skip connection is crucial for expanding the recursive multiplicative terms into higher-order interaction terms.

The final output y_t, modeled in Eq. (11), aggregates information from the projected main state (via $\mathbf{C} \in \mathbb{R}^{1\times l}$), the gated state (via $\mathbf{J} \in \mathbb{R}^{1\times m}$), and a direct residual connection $\mathbf{D} \in \mathbb{R}$. Unlike architectures that initialize feedthrough terms to zero, MG-SSM-s explicitly parameterizes $\mathbf{D}$ to preserve immediate signal propagation.

B. Dynamics of the Multiplicative Gate

The recurrence in Eq. (10) expresses non-linearity through sequential multiplication, enabling the model to capture complex dependencies that linear SSMs often miss. By recursively expanding this term, we observe that g_t behaves as a cumulative product interaction through the unfolding steps in Eq. (12) to Eq. (14):

$$g_t = \mathbf{E}\left((\mathbf{E}g_{t-2}) \odot \tilde{x}_{t-1} + \mathbf{F}x_{t-1}\right) \odot \tilde{x}_t + \mathbf{F}x_t \tag{12}$$

$$= \mathbf{E}^2 g_{t-2} \odot (\tilde{x}_{t-1} \odot \tilde{x}_t) + \mathbf{EF}(x_{t-1} \odot \tilde{x}_t) + \mathbf{F}x_t \tag{13}$$

$$= \left(\mathbf{E}^t g_0 \odot \prod_{k=1}^{t} \tilde{x}_k\right) + \sum_{k=1}^{t-1}\left(\mathbf{E}^{t-k}\mathbf{F} \odot \prod_{j=k}^{t} \tilde{x}_j\right) + \mathbf{F}x_t \tag{14}$$

Eq. (14) demonstrates that the skip connection $\mathbf{F}$ transforms the pure product chain into a sum of product interactions of varying lengths. This structure effectively mimics a higher-order interaction mechanism, allowing the network to retain memory of specific input patterns (e.g., $x_{t-1}x_t$) independently of the long-term linear trend, thereby mitigating the risk of vanishing gradients [15].

V. Experiments

A. Dataset

The experimental data for this study was sourced from the publicly available Google COVID-19 Open Data project, specifically utilizing the `table-epidemiology` dataset [16]. This repository is hosted on Google's GitHub platform and provides comprehensive time-series data related to the pandemic. The raw data is distributed as a CSV file, initially covering approximately 231 countries and regions globally.

The complete dataset aggregates multidimensional time-series data, comprising metadata such as timestamps and geospatial locations, alongside a suite of epidemiological indicators. These metrics encompass both daily increments and cumulative totals for confirmed cases, mortalities, recoveries, and testing statistics. From this extensive set of indicators, our research exclusively utilizes the new confirmed cases metric. This variable was prioritized due to its superior data integrity, characterized by robust continuity and a minimal prevalence of missing values across the global cohort compared to other available signals.

To ensure high-quality data sequences for time-series modeling, the initial cohort was filtered down to 40 countries based on demographic and economic scale, as well as absolute data continuity. For the analysis interval spanning from 6th January 2020 to 6th June 2022, countries were excluded if their time series contained any missing values, NaN flags, or anomalous zero-entries resulting from reporting lags or logging omissions.

B. Experiment Setup

1) Environment and dependencies: All experiments were executed within a Google Colaboratory environment utilizing a T4 GPU runtime for computational efficiency. For reproducibility, the primary runtime environment and key library versions used were: Python 3.10; the deep learning framework was PyTorch 2.8.0+cu126; and essential libraries included torchmetrics 1.8.2, statsmodels 0.14.6, Pandas 2.2.2, scikit-learn (sklearn) 1.6.1, and NumPy 2.0.2.

TABLE I. OPTIMAL HYPERPARAMETER SETTINGS FOR COMPARATIVE MODELS.

Model	Latent Dimension (l)	Layers	Specific Configuration
ARIMA	-	-	Tuned (p,d,q) per country
LSTM	256	1	-
Bi-LSTM	256	1	-
GRU	128	1	-
SSM	32	1	-
MG-SSM-s	64	1	Hidden Gate $(m) = 32$

2) Model architectures and configuration: We evaluated six baseline architectures against our proposed MG-SSM-s: ARIMA, LSTM, Bi-LSTM, GRU, and a discrete State Space Model (SSM). The machine learning models were implemented using PyTorch, while the ARIMA model was implemented via the statsmodels library. The SSM baseline adopts the discrete state space formulation of the S4 model, utilizing random initialization rather than the HiPPo matrix to isolate the architectural performance.

To ensure fair comparison, we conducted a grid search for the optimal latent dimension $l \in \{32, 64, 128, 256\}$ for all machine learning models. For the ARIMA models, we conducted a grid search over the parameters $p, q \in \{0, 1, 2, 3, 4, 5\}$ and $d \in \{0, 1, 2\}$. Table I summarizes the optimal hyperparameters identified for each architecture. All models were configured with a univariate input and output size (1×1).

3) Training protocol: A consistent training protocol was applied across all experiments to ensure reproducibility. We utilized a look-back window (input sequence length) of $L = 30$ time steps. Training was performed using the Adam optimizer with a initial learning rate $\alpha = 10^{-3}$ and a batch size of 64. The maximum training duration was set to 1000 epochs, implemented with an early stopping mechanism that halts training if validation loss fails to improve for 50 consecutive epochs.

To facilitate model convergence, all input features were normalized to the range $[-1, 1]$ using Min-Max scaling. This strict bounding is required because the model architecture omits internal normalization layers; constraining inputs to a maximum magnitude of 1 ensures stable activations and prevents exploding gradients. Unbounded alternatives, such as Z-score standardization, were rejected as they cannot guarantee the boundaries necessary to maintain training stability. Finally, to prevent look-ahead bias and data leakage, the data was partitioned sequentially into training, validation, and testing sets using an $8 : 1 : 1$ ratio.

To account for stochastic variations, the pipeline was executed for 21 independent runs ($N_{\text{runs}} = 21$). These runs differed only by their pseudo-random weight initializations and epoch-level batch shuffling order. All model architectures, hyperparameters, and optimization settings were kept strictly identical across all runs.

4) Performance evaluation: The primary metric for comparative evaluation was the Mean Squared Error. The quadratic penalty term in MSE effectively highlights substantial deviations from ground truth, which is critical for ensuring stability in the target application.

TABLE II. GLOBAL PERFORMANCE SUMMARY ACROSS 40 COUNTRIES. RANK DISTRIBUTION DETAILS THE FREQUENCY OF ACHIEVING EACH RANK (1ST TO 6TH) ACROSS THE DATASETS

Model	**Mean Rank**	**Rank Counts**					
		1	2	3	4	5	6
ARIMA	1.975	7	27	6	0	0	0
LSTM	5.725	0	0	0	1	9	30
Bi-LSTM	5.075	0	0	0	6	25	9
GRU	4.10	0	0	2	32	6	0
SSM	2.50	6	8	26	0	0	0
MG-SSM-s	1.625	27	5	6	1	0	1

C. Experimental Results

Table II presents the aggregated performance metrics across the full 40-country cohort. The proposed MG-SSM-s demonstrated remarkable consistency, achieving the first rank in 27 out of 40 datasets and never ranking lower than second. In contrast, ARIMA and the standard Discrete SSM achieved the top rank in seven and six cases, respectively, while the recurrent baselines (LSTM, Bi-LSTM, GRU) rarely achieved competitive rankings.

In the global evaluation across all 40 countries, the average ranking confirms this advantage. MG-SSM-s achieved the best (lowest) mean rank of **1.625**, followed by ARIMA (1.975), Discrete SSM (2.50), GRU (4.10), Bi-LSTM (5.075), and LSTM (5.725).

To validate statistical significance, we performed the Friedman test, which indicated a significant difference among the $k = 6$ models across $N = 40$ datasets ($\chi^2 = 167.228$, $p = 2.846 \times 10^{-34}$).

Subsequent post-hoc analysis using the Nemenyi test ($\alpha = 0.05$) revealed a Critical Difference (CD) threshold of 1.0793. The rank difference between the top three models, MG-SSM-s, ARIMA, and Discrete SSM, was 0.875. Since $0.875 < 1.0793$, the conservative Nemenyi test is insufficient to reject the null hypothesis of equivalence between these three top-performing architectures.

Given this result, we conducted a more sensitive pairwise analysis using the Wilcoxon Signed-Rank Test to directly compare the median MSE distributions of MG-SSM-s against the ARIMA and the Discrete SSM. This test yielded a statistic of $W = 252.0$ with a p-value of $\mathbf{0.016}$. Since $p < 0.05$, we reject the null hypothesis, confirming that MG-SSM-s provides a statistically significant improvement over the Discrete SSM. In contrast, the comparison against ARIMA resulted in $W = 315.0$ and $p = 0.103$. Since $p > 0.05$, we fail to reject the null hypothesis for this pair. These results indicate that while MG-SSM-s significantly outperforms the standard machine learning baselines, its performance is statistically comparable to the tuned ARIMA model.

D. Discussion

The results conclusively establish that the proposed MG-SSM-s is the most robust architecture among the evaluated models. While the initial Nemenyi post-hoc analysis suggested broad statistical overlap between the top-tier models, the pairwise Wilcoxon Signed-Rank Test clarified that the performance gain of MG-SSM-s is statistically significant ($p \approx 0.05$). This

TABLE III. INCREMENTAL ABLATION RESULTS. RANK DISTRIBUTION ACROSS 40 COUNTRIES. THE p-VALUE INDICATES SIGNIFICANCE RELATIVE TO THE PECEDING ROW (PAIRWISE WILCOXON SIGNED-RANK TEST)

Model Variant	Mean Rank	Rank Counts			p-value
		1st	2nd	3rd	
1. Discrete SSM (Baseline)	2.20	14	4	22	-
2. MG-SSM (Gate Only)	2.10	10	16	14	$0.177^{\dagger}$
3. **MG-SSM-s (Gate + Skip)**	**1.70**	**16**	**20**	**4**	$\mathbf{0.010}^{\ddagger}$

† vs. Discrete SSM. ‡ vs. MG-SSM. Metric: Median MSE.

finding aligns with the consistent improvement in mean rank (1.625 vs. 2.50).

We attribute this performance to the architectural design: while the standard Discrete SSM relies on purely linear time-invariant dynamics, the MG-SSM-s introduces a multiplicative gate with an explicit skip connection. This structure enables the capture of the non-linear volatility observed in daily epidemiological data, which the purely linear baseline may over-smooth.

Interestingly, while MG-SSM-s significantly outperformed all neural and state-space baselines, its performance was statistically comparable to the tuned ARIMA model ($p = 0.1032$). This suggests that the linear stochastic components of ARIMA are remarkably effective for the underlying trend of this dataset. However, MG-SSM-s achieved a superior mean rank (1.625 vs. 1.975 for ARIMA).

We attribute this stability to the architectural design: while ARIMA relies on fixed lag observations, the MG-SSM-s uses a multiplicative gate with a skip connection. This allows the model to switch between long-term trends and reacting to the non-linear volatility often found in daily epidemiological data

E. Ablation Study

To validate the necessity of each architectural component, we conducted a step-wise ablation study comparing the full model (**MG-SSM-s**) against two variants: the gate-only model (**MG-SSM**) and the standard discrete baseline (**SSM**).

We omitted the skip-only variant, as removing the multiplicative term reduces the gate equation to a simple linear projection ($g_t = \mathbf{F}x_t$). This would functionally duplicate the main linear state h_t, resulting in a redundant standard SSM with increased parameters rather than a distinct architectural mechanism.

1) Impact of the skip connection: The primary theoretical contribution of this work is the stabilization of the multiplicative gate via the skip connection ($\mathbf{F}$). As derived in Eq. (14), the gate-only variant (MG-SSM) is governed by Eq. (15):

$$\begin{aligned} g_t &= \mathbf{E}g_{t-1} \odot \tilde{x}_t \\ &= \mathbf{E}^t g_0 \odot \left(\prod_{k=1}^{t} \tilde{x}_k \right) \end{aligned} \tag{15}$$

This cumulative product poses a risk of gradient instability. Empirically, this instability was evident in Table III. While the gate-only MG-SSM improved slightly over the baseline, the improvement was not statistically significant ($p = 0.177$), likely due to optimization difficulties in the pure product chain.

Fig. 2. Pairwise comparison of Median MSE for MG-SSM (Gate Only) vs. MG-SSM-s (Full). Points below the diagonal indicate countries where the addition of the skip connection reduced the error.

However, the introduction of the skip connection (Row 3) drastically stabilized performance. MG-SSM-s achieved a statistically significant improvement over the gate-only variant ($p = 0.010$), shifting the rank distribution such that it rarely fell to last place (only 4/40 cases). Visual evidence of this improvement is presented in Fig. 2.

2) Impact of the multiplicative gate: The comparison between the baseline SSM and the MG-SSM (Rows 1 vs. 2) isolated the effect of the multiplicative mechanism. Although the MG-SSM reduced the count of worst-case performances (Rank 3 dropped from 22 to 14), it failed to consistently outperform the linear baseline without the skip stabilizer. This confirmed that while the multiplicative gate provides high-capacity dynamics, the skip connection is a requisite component to harness this capacity effectively.

VI. CONCLUSION

The rapid propagation of the COVID-19 pandemic has placed unprecedented strain on global healthcare systems, particularly in densely populated nations. In this context, accurate forecasting of confirmed case counts is critical, providing policymakers with the necessary intelligence to anticipate surges, allocate resources, and implement effective public health measures. Furthermore, reliable predictive data serves to motivate public adherence to mitigation strategies intended to curb the virus's spread.

In this study, we evaluated the efficacy of five deep learning architectures—LSTM, Bi-LSTM, GRU, SSM, the proposed MG-SSM-s, and ARIMA using the Google COVID-19 Open Data repository. The analysis encompassed daily confirmed cases across 40 distinct countries, including major epicenters such as the USA, India, Brazil, France, and Germany. This methodological approach utilized a 30-day look-back window

to forecast next-day confirmed cases, capitalizing on the inherent capacity of deep learning models to capture non-linear temporal dependencies within complex epidemiological data.

Performance verification based on the Mean Squared Error metric demonstrates that the MG-SSM-s model achieves superior forecasting accuracy compared to all deep learning baselines, while delivering comparable statistical performance to a tuned ARIMA model. These results highlight the potential of the MG-SSM-s framework as a robust tool for epidemiological time-series forecasting.

The results demonstrate that the proposed MG-SSM-s is a highly robust architecture among the evaluated models. While the initial Nemenyi post-hoc analysis suggested statistical overlap among the top-tier models, the pairwise Wilcoxon Signed-Rank Test clarified that the performance gain of MG-SSM-s over the standard SSM is statistically significant ($p \approx 0.05$), aligning with its superior mean rank (1.625 vs. 2.50). Interestingly, while MG-SSM-s significantly outperformed all neural and state-space baselines, its performance was statistically comparable to the tuned ARIMA model (p = 0.1032), though MG-SSM-s maintained a superior mean rank (1.625 vs. 1.975 for ARIMA). This performance and stability stem from the architectural design: while standard SSMs rely on purely linear dynamics and ARIMA relies on fixed lag observations, MG-SSM-s introduces a multiplicative gate with an explicit skip connection. This structure enables the model to dynamically switch between tracking long-term trends and capturing the non-linear volatility observed in daily epidemiological data without over-smoothing.

A. Limitations and Future Work

While the framework demonstrates strong forecasting capabilities, its current constraints highlight several avenues for future research:

- Univariate Input Constraint: The model currently relies solely on historical case numbers. Future iterations will extend the architecture to multivariate inputs to incorporate exogenous variables like government policy changes and mobility indexes.
- Data Evaluation Scope: The empirical evaluation was restricted to 40 countries chosen primarily as major pandemic epicenters. Broader validation across diverse regional data qualities is required to verify global generalizability.
- Look-Back Sensitivity: The analysis relied on a fixed 30-day look-back window. Investigating how varying or adaptive window horizons affect forecasting accuracy remains an open direction for exploration.

Declaration of Generative AI and AI-Assisted Technologies in the Manuscript Preparation Process

During the preparation of this work, the author(s) used Google's Gemini 2.5 Flash, in order to refine and improve the clarity, grammar, and flow of the language in the manuscript. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the published study.

References

[1] H. Hu, K. Nigmatulina, and P. Eckhoff, "The scaling of contact rates with population density for the infectious disease models," *Mathematical biosciences*, vol. 244, no. 2, pp. 125–134, 2013.

[2] O. Postavaru, S. R. Anton, and A. Toma, "Covid-19 pandemic and chaos theory," *Mathematics and Computers in Simulation*, vol. 181, pp. 138–149, 2021.

[3] S. I. Alzahrani, I. A. Aljamaan, and E. A. Al-Fakih, "Forecasting the spread of the covid-19 pandemic in saudi arabia using arima prediction model under current public health interventions," *Journal of infection and public health*, vol. 13, no. 7, pp. 914–919, 2020.

[4] J. Yuan, Y. Wu, W. Jing, J. Liu, M. Du, Y. Wang, and M. Liu, "Non-linear correlation between daily new cases of covid-19 and meteorological factors in 127 countries," *Environmental research*, vol. 193, p. 110521, 2021.

[5] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for covid-19 with deep learning models of lstm, gru and bi-lstm," *Chaos, Solitons & Fractals*, vol. 140, p. 110212, 2020.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] Y. Kong, Z. Wang, Y. Nie, T. Zhou, S. Zohren, Y. Liang, P. Sun, and Q. Wen, "Unlocking the power of lstm for long term time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 11, 2025, pp. 11 968–11 976.

[8] W. Tangseefa, T. Pumpaibool, P. Khanarsa, and K. Sinapiromsaran, "Forecast covid-19 epidemics by strengthening deep learning models with time series analysis." *International Journal of Advanced Computer Science & Applications*, vol. 16, no. 7, 2025.

[9] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[10] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[11] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[12] J. T. Smith, A. Warrington, and S. W. Linderman, "Simplified state space layers for sequence modeling," *arXiv preprint arXiv:2208.04933*, 2022.

[13] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First conference on language modeling*, 2024.

[14] W. L. Brogan, *Modern control theory*. Pearson education india, 1985.

[15] S.-H. Noh, "Analysis of gradient vanishing of rnns and performance comparison," *Information*, vol. 12, no. 11, p. 442, 2021.

[16] G. C. Platform, "Covid-19 open data: Epidemiology table," https://github.com/GoogleCloudPlatform/covid-19-open-data/blob/main/docs/table-epidemiology.md, 2020–2022, accessed: 12/11/2025.