Contents lists available at GrowingScience

Decision Science Letters

homepage: www.GrowingScience.com/dsl

Extending the forecasting horizon of daily new COVID-19 cases using non-pharmaceutical measures and the effective reproduction number (Rt): A deep learning-based framework

Tuga Mauritsius[a*]

[a]*BINUS Graduate Program - Master of Information System Management, Information Systems Management Department, Jakarta 11480, Indonesia*

C H R O N I C L E

Article history:
Received: October 1, 2024
Received in revised format: January 28, 2025
Accepted: January 28 2025
Available online: January 30, 2025

Keywords:
Covid-19
Deep Learning
Multivariate Time Series
MASE
Rt (effective reproduction number)
Mathematical Epidemiological Model (MEM)/ Compartmental Model

A B S T R A C T

Amid the ongoing pandemic, such as the Covid-19 outbreak, there exists a critical need to comprehend and forecast the dynamic trends of daily confirmed cases to effectively prevent and mitigate the impact of its consequences. This study aims to investigate the essential factors acting as predictors for forecasting daily new confirmed cases specifically within the Indonesian setting. Utilizing advanced Deep Learning (DL) methodologies, including Deep Feedforward Neural Networks (DFNN), Long Short-Term Memory (LSTM), one-dimensional convolutional neural networks (CONV1D), and Gated Recurrent Units (GRU), this research endeavors to predict daily confirmed Covid-19 cases in Indonesia. To achieve this, a comprehensive set of 80 variables (predictors), encompassing the effective reproduction number (Rt), was utilized as input parameters. Before model construction, rigorous variable selection procedures and statistical analyses were conducted to enhance data understanding. The effectiveness of the predictive model was assessed using various metrics, such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Scaled Error (MASE), which evaluates MAE relative to a baseline model. Results indicate that DL models incorporating two key predictors—daily confirmed case count and Rt—exhibited superior predictive performance, capable of forecasting daily confirmed cases up to 13 days in advance. The inclusion of additional variables was found to diminish the predictive accuracy of DL algorithms.

1. Introduction

The Covid-19 pandemic, caused by the SARS-CoV-2 virus, has led to significant morbidity and mortality globally, transcending national boundaries to become an unprecedented crisis with multifaceted repercussions. Its impact extends beyond health, affecting socioeconomic, educational, tourism, and entertainment sectors, causing a widespread economic downturn across nations. Indonesia, having recorded its initial cases on March 02, 2020, reported a staggering 6,556,627 positive cases and 159,068 deaths as of November 12, 2022, positioning it as a significant hotspot in Southeast Asia. Understanding daily case fluctuations is crucial for resource allocation and effective response strategies, guiding healthcare provisions and governmental interventions to mitigate pandemic consequences. Since the onset of the Covid-19 pandemic, numerous studies have sought to comprehend its dynamics and quantify its propagation. A literature review conducted by (Gnanvi et al., 2021) disclosed that during the period spanning January 01, 2020, to November 30, 2020, a

* Corresponding author.
E-mail address tmauritsus@binus.edu (T. Mauritsius)

ISSN 1929-5812 (Online) - ISSN 1929-5804 (Print)

doi: 10.5267/j.dsl.2025.1.008

minimum of 4311 peer-reviewed articles investigated modeling methodologies applied to Covid-19. These studies predominantly utilized mathematical modeling or compartmental models (46.1%) and statistical approaches (growth models and time series) (31.8%), while a smaller proportion employed artificial intelligence techniques (6.7%), Bayesian methodologies (4.7%), network models (2.3%), and agent-based modeling (1.3%).

Consistent with the findings in (Gnanvi et al., 2021) , Zhang et al. (2022) corroborated that compartmental models constitute the predominant approach employed by researchers to forecast the transmission dynamics of Covid-19. These models trace their origins back to the seminal work of Kermack and McKendrick (1927), who introduced a mathematical framework in which the population is partitioned into distinct compartments, each characterized by individuals sharing similar attributes. Termed the SIR model, it encompasses three compartments: susceptible (S), infected (I), and recovered (R) individuals. The model's dynamics are encapsulated within three differential equations, governing the movement of individuals between compartments over time. The rate at which individuals move between these compartments is governed by parameters such as the transmission rate (β) and the recovery rate (γ).

The Effective Reproduction Number (Rt) is a key metric in epidemiology, closely associated with the Susceptible-Infectious-Recovered (SIR) model (Bsat et al., 2022; Cortés-Carvajal et al., 2022; Locatelli et al., 2021; Yang et al., 2022). It represents the average number of secondary infections caused by a single infectious individual at a given time. Rt is derived by multiplying the basic reproduction number (R0) by the proportion of the population still susceptible. This model elucidates the dynamics of disease transmission and its impact on population dynamics. An Rt value exceeding 1 indicates exponential spread, with each infected individual, on average, infecting more than one other person. Conversely, an Rt below 1 indicates a decline in the disease, with each infected individual, on average, infecting fewer than one other person.

Compartmental models are adaptable frameworks in epidemiological research, allowing researchers to modify their structure to improve accuracy by introducing new compartments or refining existing ones to better align with observed data. This flexibility has led to the development of diverse pandemic models rooted in compartmental frameworks, as evidenced by various publications utilizing mathematical epidemic models (MEM) or compartment-based models to analyze Covid-19 transmission dynamics (Goel et al., 2021; Inthamoussou et al., 2022; Luqmanul et al., 2021; Marinov & Marinova, 2022; Wintachai & Prathom, 2021; Zheng et al., 2020). However, despite its prevalence, the compartmental approach has notable limitations. Foremost among these is its reduced efficacy in long-term prediction, attributed to static hyperparameters that fail to adequately capture the dynamic nature of interventions such as governmental policies, pharmaceuticals, and non-pharmaceutical measures, as well as the impact of virus mutations (Lucas et al., 2023). Additionally, the compartmental methodology lacks systematic validation of long-term forecasts, hindering comparative analysis with other models and highlighting its inadequacy in capturing the evolving complexity of Covid-19 dynamics.

Numerous scholars have opted for machine learning as an alternative approach for forecasting the progression of Covid-19 within a country (Abolmaali & Shirzaei, 2021; Absar et al., 2022; Alassafi et al., 2022; Cinaglia & Cannataro, 2022; Delli Compagni et al., 2022; John-Otumu et al., 2024; Khalifa et al., 2023; Kim et al., 2022; Masum et al., 2022; Nabi et al., 2021; Qu et al., 2023; Shuai et al., 2024; Verma et al., 2022; Wathore et al., 2023; Xu et al., 2022). The utilization of machine learning methodologies typically seeks to enhance the efficacy of forecasting models. This goal is pursued through two primary avenues: algorithm refinement and the utilization of superior-quality data inputs during modeling. While the majority of studies concentrate on optimizing forecasting performance through algorithmic enhancements, scarce literature addresses the enhancement of model performance via data quality improvements. Nevertheless, only recently a few studies have been identified in this regard (Jo & Kim, 2023; Khalifa et al., 2023; Kim et al., 2022; Livieris, 2023; Lucas et al., 2023; Trajanoska et al., 2022; Wathore et al., 2023). These authors highlight that prior researches predominantly emphasize enhancing forecasting accuracy through the adoption of more complex model architectures and advanced learning methodologies, often neglecting the crucial aspect of obtaining high-quality training data. Given that machine learning fundamentally operates on data-driven principles, meticulous attention to input data quality is essential. The distinctive contribution of our study lies in this direction.

Our study integrates a multitude of pharmaceutical and non-pharmaceutical variables sourced from diverse repositories such as the Indonesian Covid-19 Task Force (Satuan Tugas Penanganan COVID-19, 2022), Our World in Data (Our World in Data, 2022), Blavatnik School of Government (Blavatnik School of Government, 2022), and Google Community Mobile Reports (Google, 2022). Hence, the initial contribution of our study lies in the necessity for variable filtration due to the extensive array of variables involved, aiming to pinpoint those variables substantially influencing the daily case count over the forthcoming days. To accomplish this, we employed the Extreme Gradient Boost Regressor (XGBRegressor) algorithm to assess the significance of each variable. Furthermore, preceding the modeling phase, we undertook data analysis and exploration utilizing statistical methodologies to provide researchers with a preliminary understanding of the dataset's characteristics.

The significance of this procedure was underscored by its ability to unveil the previously unexplored yet noteworthy role of Rt as one of the predictors within our model. Before undertaking the variables filtering procedure, the fundamental premise of our study posits that Rt holds predictive value for the trajectory of epidemics or outbreaks. With accurate knowledge or estimation of Rt, projections of forthcoming infections can be made, thereby informing healthcare resource allocation, vaccination strategies, and other mitigation initiatives. While Rt's utility as an immediate metric is well-established, its potential for forecasting future outbreaks remains incompletely explored. To the best of our knowledge, no prior study has integrated Rt into forecasting models for predicting the trajectory of outbreaks. Thus,

the primary contribution of this study resides in this novel approach. Ultimately, this study introduces a novel approach for evaluating model performance, constituting another noteworthy contribution. A model's efficacy is deemed satisfactory if its performance equals or surpasses that of a baseline model. The selected baseline model in this research is a simplistic naïve model, which assumes the current number of cases to be identical to the previous day. Overcoming the naïve model presents a challenge due to the inherent nature of time-series data, which tends to exhibit persistence. As depicted in Figures 3 and 4, the autocorrelation analysis of Covid-19 time series data in Indonesia reveals a notably high correlation at lag = 1, justifying the selection of the baseline model. Consequently, the predictive capability of a model is evaluated based on the maximum horizon within which it can accurately forecast. Unlike prior studies where the prediction horizon was treated as an input parameter for models, this study considers the horizon as one of the output metrics of our models.

2. Theory/Calculation

2.1 Time series

Sample Time-series data refers to a sequence of data points gathered at regular intervals. If each data point in the time-series dataset comprises multiple values (multidimensional), rather than solely a single value, it is categorized as a multivariate time series (Wei, 2018). Otherwise, it is classified as a univariate time series. The typical structure of time-series data can be expressed as follows:

$$x = \{x_1, x_2, \dots, x_t\} \tag{1}$$

where x_t is the current value, and $\{x_1, x_2, \dots, x_{t-1}\}$ are the past values. In the case of a multivariate time series, point $\boldsymbol{x_i}$ is a vector of fixed length *d,* that is, $\boldsymbol{x_i} = \{x_{i,1}, x_{i,2} \dots x_{i,d}\}$. Time-series forecasting models predict future values of a target y_{t+h} for a given entity at time *t+h*. The *h-step-ahead* forecasting model takes the following form.

$$\hat{y}_{t+h} = f(y_{t-k:t}, x_{t-k:t}) \tag{2}$$

where

$\hat{y}_{t+h}$ is the prediction of the model in the *h* future time horizon

$$y_{t-k:t} = \{y_{t-k,\dots} y_t\}$$

$$x_{t-k:t} = \{x_{t-k,\dots} x_t\}$$

In this context, $\hat{y}_t$ represents the predicted value of the target variable, which in this study pertains to new daily cases. The input vector, denoted as, $x_{t-k:t}$ encompasses exogenous factors presumed to influence the observed value, considering a look-back window of length *k*. The function *f(.)* represents the prediction function derived from modeling subsequent to the training phase conducted with historical data. Given that Deep Learning (DL) serves as the technique employed in this investigation, the subsequent section provides an overview of Artificial Neural Networks (ANN) and its derivatives in DL.

2.2 ANN

Artificial neural networks replicate the operational mechanism of the human brain (Witten et al., 2011). The human brain encompasses numerous neural connections where incoming signals undergo processing within each neuron. Correspondingly, an ANN is composed of multiple interconnected nodes. Each node accepts input signals presented in matrix format (x) alongside a weight matrix (W) originating from the preceding layer.

Furthermore, in each node, the following two operations are performed:

1. Multiply (inner product) between the input and weight and add up with the y-intercept b term

$$v = Wx + b \tag{3}$$

2. Activating the resulting weighted sum by using a function. The output of this function calculation continues to the next node(s) as input for that node(s).

$$y = \varphi(v) = \varphi(Wx + b) \tag{4}$$

3. The activation function $\boldsymbol{\varphi}(.)$ and bias term b are crucial components in artificial neural networks (ANNs), facilitating information retention and pattern extraction through weight and bias storage. ANNs can vary in architecture, with shallow networks containing only one hidden layer, while deep networks have two or more hidden layers, currently dominating practical applications. During supervised learning, weights are adjusted iteratively to minimize disparities between expected and model outputs, with the back-propagation algorithm playing a pivotal role. This algorithm systematically updates weights based on provided information, constituting a structured learning rule within the training regimen of deep neural networks (DNNs)**.**

2.3 Deep Neural Networks and Deep Learning

Increasing the depth of a deep neural network (DNN) architecture doesn't consistently improve model performance, often leading to diminished results due to inadequate training. Two primary challenges encountered during training are the vanishing gradient problem and overfitting. The vanishing gradient occurs when error signals fail to effectively propagate to initial layer nodes, hindering weight updates in subsequent layers, rendering additional hidden layers ineffective. Overfitting is a significant concern in DNNs due to their complex structure and larger parameter count, exacerbated by the presence of numerous hidden layers. Dropout, randomly deactivating a percentage of nodes during training, has proven effective in mitigating overfitting, along with incorporating a regularization term into the cost function. In time series analysis, Deep Learning (DL) (Witten et al., 2011) within DNNs involves encoding historical data into a latent variable followed by final estimation solely from this variable.

$$f(y_{t-k:t}, x_{t-k:t}) = g_d(z_t) \tag{5}$$

$$z_t = g_e(y_{t-k:t}, x_{t-k:t}) \tag{6}$$

The functions $g_e(.)$ and $g_d(.)$ denote the encoder and decoder functions, respectively. Consequently, these encoders and decoders constitute the fundamental components of deep learning frameworks, wherein the selection of the network influences the kinds of relationships that the models are capable of learning.

2.3.1 Convolutional Neural Networks (CNN)

A CNN, designed to emulate brain visual processing, consists of two primary layers: a feature extraction (encoder) layer and a classification (decoder) layer employing a multiclass classification neural network. During training, CNN autonomously extracts features using a feature-extraction layer comprising convolutional and pooling layers. The convolution layer applies filters to the input, generating feature maps, while the pooling layer reduces dimensionality by merging adjacent elements. CNN operates conceptually in two-dimensional space for image data but can adapt to one-dimensional data like time series.

2.3.2 1-D Convolution Neural Network (Conv1d) for Time Series

In the realm of time-series data, a 1-D Convolutional Neural Network (Conv1d) operates with a fixed-width kernel matching the dataset's width and variable-length progression across the time steps. At each step, the kernel convolves with the time series, producing filtered elements subjected to a non-linear activation function. Max-pooling extracts predominant values from each filtered vector, forming a maximum vector input for the subsequent fully connected decoding layer.

2.3.3 Long Short-Term Memory Network LSTM and GRU

LSTM and GRU represent deep learning neural network architectures designed to comprehend the inherent order dependencies present in sequence prediction tasks. These models incorporate internal mechanisms termed gates, responsible for managing the information flow. These gates possess the ability to learn and discern data, enabling them to determine the significance of information for retention or omission. Consequently, LSTM and GRU are capable of transmitting pertinent information along extensive sequence chains to facilitate predictive objectives. By employing a sequence of LSTM and GRU gates, the network can effectively retain solely relevant information while disregarding extraneous data.

2.4 XGBoost

XGBoost is a highly popular ensemble algorithm, utilizing bootstrapping sampling techniques. It belongs to the class of gradient boosting algorithms, which amalgamate multiple weak learners, usually decision trees, to enhance predictive models. Compared to traditional gradient boosting methods, XGBoost boasts advantages in speed and scalability. It employs optimization techniques such as parallel processing and column block for efficient memory utilization. Additionally, XGBoost offers flexibility through support for various objective functions and evaluation metrics, catering to diverse problem domains.

2.5 Time Series Data Analysis

2.5.1 Autocorrelation function (ACF)

A fundamental aspect of time-series analysis involves the autocorrelation function (ACF). Autocorrelation becomes evident when examining the scatter plot of data pairs y_{t+1} with y_t for a lag of one ($k = 1$). If no discernible structured pattern emerges between these observations, they are deemed uncorrelated. In essence, the value of y in the present time period does not offer informative insights into the value of y observed in the subsequent period.

2.5.2 Multicollinearity

Multicollinearity, as discussed in academic literature, denotes strong correlations among independent variables in a multiple regression model, potentially leading to increased variability in coefficients. Although multicollinearity doesn't directly diminish the overall predictive capability of the model, it does affect the reliability and consistency of predictions for individual variables. Detection methods include assessing Pearson's correlation coefficient, where multicollinearity is indicated by correlations exceeding 0.8, and examining Variance Inflation Factor (VIF) values, representing the ratio of variances observed in the full model versus models with specific variables under scrutiny.

2.6 Performance Metric Evaluation

2.6.1 Mean Absolute Error (MAE)

MAE provides the average value of the absolute difference between a model's actual and predicted values across all the datasets. This value measures the average error or the residual. The MAE was calculated using the following formula:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{7}$$

where

y_i the actual values

$\hat{y}_i$ the predicted values

2.6.2 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) method provides information on the extent to which the forecasting error is compared to the actual value of the dataset. The smaller the percentage error value in the MAPE, the more accurate the forecasting results of the model.

The formula for MAPE is as follows:

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{8}$$

2.6.3 Mean Squared Error (MSE)

The (MSE) is the average squared error between the actual and forecast values. A low Mean Squared Error value or a mean squared error value close to zero indicates that the forecasting results follow the actual data and can be used for forecasting calculations in future periods. The mean squared error (MSE) was calculated using the following equation.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{9}$$

2.6.4 Root Mean Squared Error (RMSE)

As indicated by the name, the RMSE is the square root of the MSE. RMSE was introduced to make the error scale equal to the target scale.

$$RMSE = \sqrt{MSE} \tag{10}$$

2.6.5 Mean Absolute Scaled Error (MASE)

MASE, an error metric devoid of scale dependency, evaluates the efficacy of a forecasting model vis-à-vis a reference method, typically a simplistic naive forecasting $\widehat{y_{t+1}} = y_t$. In our investigation, we utilize Mean Absolute Error (MAE) as the model's performance benchmark. MASE is computed by dividing the MAE of the forecast by that of the naive approach. A value of 1 signifies that the forecast model performs equivalently to the naive model, while values below 1 denote superior performance and those above 1 indicate inferior performance compared to the naive model. Notably, MASE equals one for the naive forecast or closely approximates it.

3. Data and methods

3.1 Dataset description

This dataset amalgamated information from multiple sources: daily active cases data were sourced from the Our World in Data (OWID) website (Our World in Data, 2022), control policy data implemented by the Indonesian government were retrieved from The Oxford Covid-19 Government Response Tracker (OxCGRT) website (Blavatnik School of Government, 2022), and community mobility patterns were acquired from Google Community Mobility Reports (Google, 2022). Spanning from March 16, 2020, to August 09, 2022, encompassing a total of 877 days (as illustrated in Fig. 1), this dataset was partitioned into 80% for training purposes and the remaining portion for testing and validation.

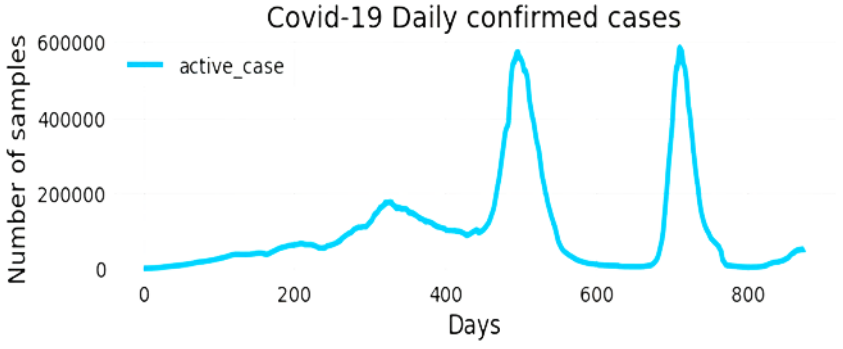

Fig. 1 Time series depicting daily confirmed Covid-19 cases in Indonesia spanning from March 16, 2020, to August 09, 2022

Comprising a total of 80 columns, the dataset was primarily derived from OxCGRT (44 columns), supplemented by seven columns from the Google Mobility index, and 19 columns from OWID. Additionally, seven columns were sourced from the Indonesian Covid-19 Task Force (Satuan Tugas Penanganan COVID-19, 2022) . Two additional variables were generated by the authors: data pertaining to holidays and the count of individuals with no prior exposure.

The authors constructed holiday data by assigning a value of 0 to denote a working day and an integer value greater than or equal to 1 (represented as x) if the day fell within a series of consecutive holidays spanning x days. All data acquired from the aforementioned sources were inherently numerical and did not necessitate additional quantification. Following an initial review, redundancies were identified where multiple columns contained identical information under different labels. Consequently, redundant columns were eliminated, resulting in the formation of a final dataset comprising 877 records and 67 columns, primed for subsequent analysis.

3.2 Data preparation and exploration

During the preparatory and exploratory phases, two procedures were conducted. Firstly, the dataset, initially consisting of 67 columns, underwent multicollinearity assessment using Variance Inflation Factor (VIF) values to identify and eliminate highly correlated variables. The elimination process ceased when all variables exhibited VIF values below 100, resulting in a subset of 25 variables for further analysis. Secondly, series lag relationships were examined to assess seasonality within the dataset. Scatter diagrams were initially used to explore correlations between datasets with lag intervals up to 14, followed by calculations of Partial Autocorrelation Function (PACF) to refine the observation. Fig. 3 exhibits a discernible systematic pattern evident in the new case data for a lag of one day. The plot illustrates a relationship that conforms to the equation $y_{t+1} = y_t$ indicative of consistent data distribution and a pronounced correlation within the dataset at a lag of one day. As lag intervals increase, the correlation diminishes

proportionately. Furthermore, Figure 4 corroborates the exceptionally high correlation nearing unity between data points with a lag of one day. Predicting tomorrow's new cases as equivalent to the current count represents a plausible assumption, supported by the robust performance of the naive model. However, while the naive model demonstrates high accuracy in estimating next-day case counts, its practical utility is limited as it fails to forecast cases beyond a single-day horizon.

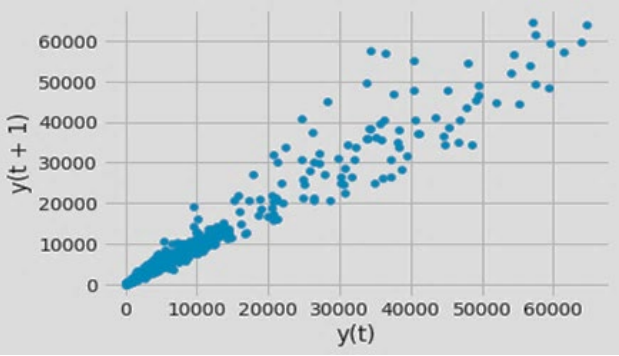

Fig. 3. Scatter plot depicting daily Covid-19 cases on day t and day t+1 (lag = 1)

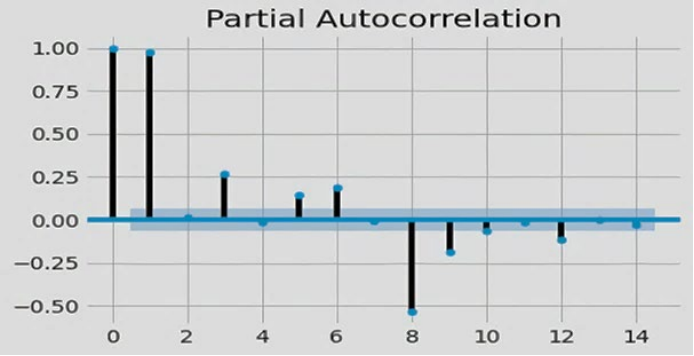

Fig. 4. Partial Autocorrelation Function (PACF) plotted for lag intervals ranging from 0 to 14

3.3 Methods

The primary aim of this study is to construct a Deep Learning (DL) model for forecasting the daily incidence of new Covid-19 cases in Indonesia, employing both univariate and multivariate methodologies. Prior to delving further, it is imperative to establish two terms that will be frequently referenced: “horizon”, denoting the duration of time leading up to the future point to be predicted, and “window size”, representing the number of preceding time intervals utilized in model construction.

Deep Learning Architectures for Time Series Prediction:

This study employs four Deep Learning (DL) architectures for time series forecasting: deep feedforward neural networks, Long Short-Term Memory (LSTM), one-dimensional convolutional neural networks (Conv1D), and Gated Recurrent Units (GRU). The constructed model incorporates both univariate and multivariate time series data.

3.3.1 Univariate Experimental Setup

The experimental procedures utilizing univariate data are depicted in Fig. 5. The complete dataset consisting of 25 columns was employed, although solely the active cases column was utilized. This experimental configuration, utilizing only active cases as input, is designated as Scenario 1. Referring to the PACF plot, it is anticipated that the model's predictions will extend up to a horizon of 8. Dataset windowing was conducted to format the dataset appropriately for input into each designated model. The resultant dataset size after this procedure (n^') is contingent upon the window size and horizon, defined as follows:

$$n' = n - w - h + 1 \tag{11}$$

where

n' = the length of the new data set

n = the length of the original data set

w = window size

h = horizon length

The optimal window size was determined via an initial experiment, yielding a range spanning from 6 to 21 days. The study was executed with a training-testing split of 80% and 20%, respectively. The final row of testing data is utilized for forecasting purposes employing the most effective model. The configurations of the four Deep Learning models are depicted in Figures 6 to 9, whereas the parameters employed by each model are outlined in Table 1.

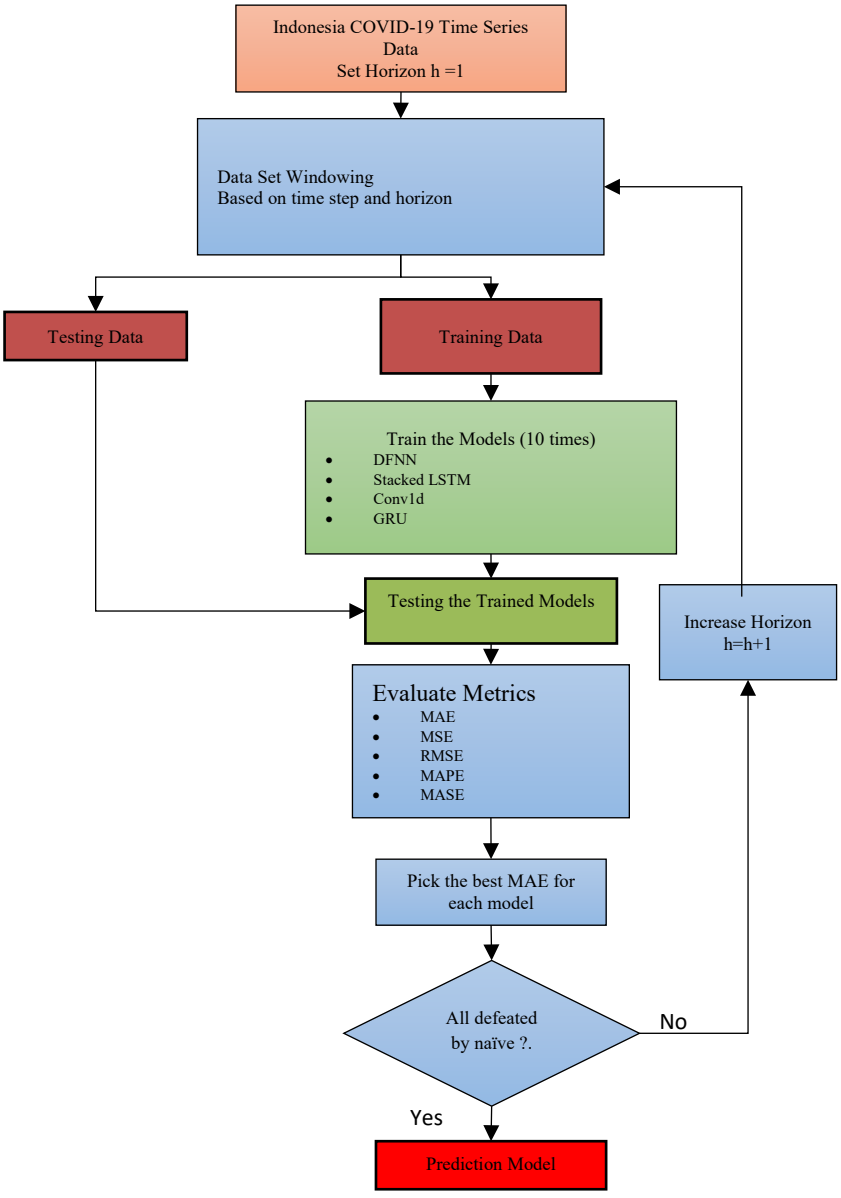

Fig. 5. Schematic illustrating the procedural steps involved in Univariate Experiments

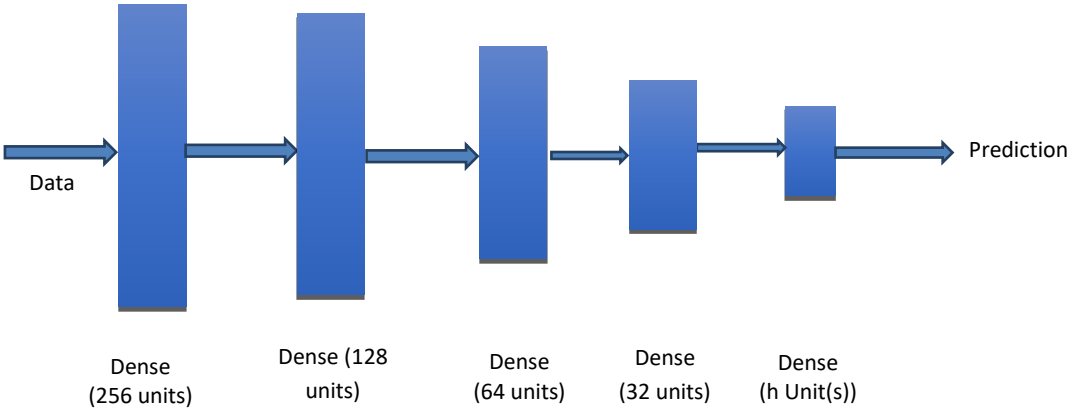

Fig. 6. Structural layout of the Deep Feedforward Neural Network (DFNN)

Fig. 6 illustrates the architecture of the Deep Feedforward Neural Network (DFNN). The input layer receives time series data, configured based on the input sequence length (input_length) and the number of variables in the time series (input_dim), represented as (w, d), where w is the window size and d is the dimension (with d being 1 for univariate time series). Subsequently, four hidden layers with neurons (units) employing rectified linear unit ('relu') activation functions are integrated. The output layer predicts time series values, with the number of units matching the forecast horizon (h). Further details on the parameters used in this architecture are provided in Table 1.

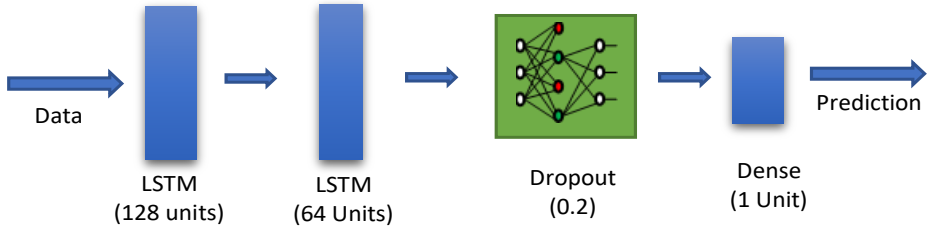

Fig. 7. Structural configuration of the Stacked Long Short-Term Memory (LSTM) architecture

Fig. 7 depicts the configuration of the stacked Long Short-Term Memory (LSTM) model, consisting of two LSTM layers arranged sequentially and augmented with dropout layers to mitigate overfitting. The input layer maintains the same structure as the Deep Feedforward Neural Network (DFNN), denoted as (w, d), where w represents the window size and d the dimension of the input time series data. Stacking LSTM layers enables the model to capture intricate temporal dependencies within the data, with each layer transmitting its hidden state to the subsequent layer, facilitating the acquisition of diverse levels of abstraction. Dropout layers are positioned after the LSTM layers to address overfitting and improve generalization. Finally, the output layer predicts the time series values. For comprehensive parameter details, please refer to Table 1.

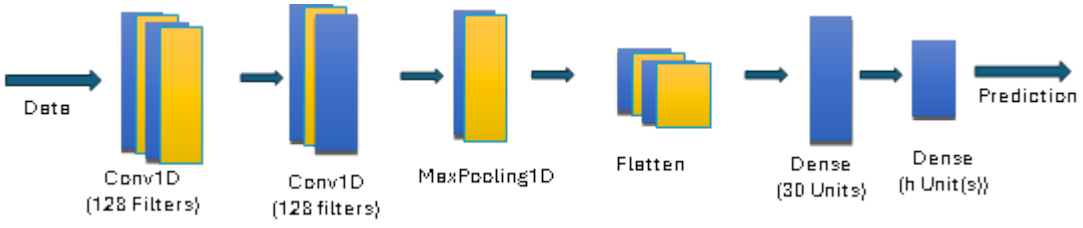

Fig. 8. Structural layout of the Convolutional 1D (Conv1D) architecture

Fig. 8 illustrates our one-dimensional Convolutional Neural Network (CNN), featuring a convolutional hidden layer designed to process a 1D sequence. Subsequently, a second convolutional layer and a pooling layer are applied to distill the output of the convolutional layer, emphasizing the most pertinent elements. Following the convolutional and pooling layers, a dense, fully connected layer interprets the features extracted by the convolutional segment of the model. A flatten layer is interposed between the convolutional and dense layers to condense the feature maps into a singular one-dimensional vector. Each sample's input shape comprises the number of time steps and the number of features. Comprehensive details regarding the parameters utilized in our experiments are provided in Table 1.

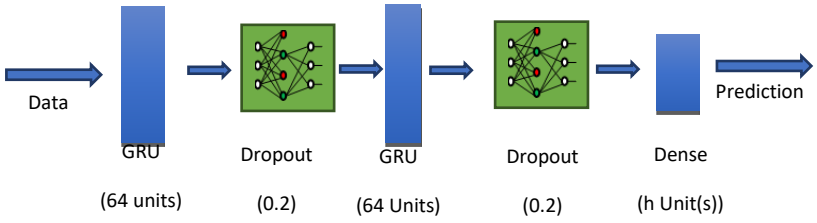

Fig. 9. Structural configuration of the Gated Recurrent Unit (GRU) architecture

Fig. 9 illustrates the architecture of the GRU model employed in this study. Beginning with an input layer structured as (w, d) to accommodate sequential input data, GRU layers play a pivotal role in capturing temporal dependencies within the time series data. Comprising multiple GRU units, these layers process sequential information, with outputs transmitted hierarchically to subsequent layers, facilitating the representation of intricate patterns. The model integrates two GRU layers to capture enduring dependencies effectively. To address overfitting and enhance generalization, two dropout layers follow the GRU layers, and a dense layer is incorporated thereafter to facilitate nonlinear transformations and feature extraction. Details regarding the number of units in the dense layers and other parameters are provided in Table 1. The final output layer comprises h neurons, representing predictions.

Table 1
Hyperparameters of Four Deep Learning Models

	Naive	DFNN	LSTM	Conv1D	GRU
Split Percentage	(80,20)	(80,20)	(80,20)	(80,20)	(80,20)
Window Size	N/A	7 – 21	7 - 21	7 – 21	7 - 21
# Trial	1	10	10	10	10
Hidden Layers	N/A	4	Default	Default	Default
Hidden Units	N/A	(256,128,64,32)	(128,128)	128,128	32,32
Activation Function	N/A	Relu	Default	Relu	Relu
Convolutional Filters	N/A	N/A	N/A	128,128	N/A
Stride	N/A	N/A	N/A	5,3	N/A
Padding	N/A	N/A	N/A	Causal	N/A
Learning Rate	N/A	Default (0.001)	Default	Default	Default
Optimizer	N/A	Adam	Adam	Adam	Adam
Metric evaluation	N/A	MAE	MAE	MAE	MAE
Epoch	N/A	150	150	150	150
Batch Size	N/A	2	2	2	2

Note: In this study, all algorithms are implemented utilizing TensorFlow Keras API version 2.6.0. In this context, the term "Default" pertains to the default settings provided by the Keras library.

3.3.2 Multivariate Experimental Setup

The multivariate experiment follows a methodology akin to that depicted in Figure 6. Diverging from univariate datasets, multivariate datasets necessitate preprocessing steps such as scaling and variable selection. Consequently, two additional processes are introduced in the second box of Figure 5: dataset scaling and filtering. The dataset employed encompasses the full dataset comprising 25 columns. Data scaling becomes imperative due to the substantial disparities in the scales of the variables within the dataset. To address this, Minimax scaling within the range [0,1] is implemented. Variable filtering employs the XGB Regressor algorithm, entailing multiple regression computations between all predictors at time t against new_cases at t+h, where h varies within the range h = 1,2,…,28. Subsequently, the three variables exhibiting the highest weights in each horizon are retained for further analyses. Based on these outcomes, experiments are conducted across several scenarios predicated on the optimal number of variables with superior weights. Dataset windowing is undertaken to format the dataset appropriately for input into each designated model. The resultant dataset's row count (n') post this process hinges on the window size and horizon, as illustrated in Equation (17). Optimal window size determination is guided by preliminary experiments. This study adopts an 80% training and 20% testing data splitting scheme. Notably, the last row of the validation data is reserved for testing purposes, offering insights into the forecasting efficacy of the optimal model. The architectural configurations for the four Deep Learning (DL) models employed are elucidated in Fig. 6 to Fig. 9, while the respective parameter specifications are delineated in Table 1.

4. Results

4.1 Univariate Experimental Results

Table 2 delineates the outcomes of the experiment, employing deep learning methodologies for forecasting new_cases h horizon days, employing window steps and a defined hyperparameter configuration as illustrated in Fig. 5 and Table 1. These experimental procedures were conducted utilizing a laptop equipped with an AMD Ryzen 3 4300U processor featuring Radeon Graphics and 8 gigabytes of RAM. The duration of execution for the steps depicted in Fig. 5 spans from 2,000 to 13,000 seconds.

Fig. 10 (univariate analysis) displays the outcomes documented in Table 2. This figure exclusively presents the Mean Absolute Scaled Error (MASE) acquired by each model derived from univariate experiments (Scenario 1). Additionally, for comparison purposes, the results of experiments conducted under multivariate scenarios are also depicted within the same figure.

Table 2
Evaluation of Optimal Deep Learning Models on Test Data Utilizing Univariate Predictors (Scenario 1)

		MAE	MSE	RMSE	MAPE	MASE
Horizon = 1	NAÏVE	1313.317871	10995298.0	1313.317871	18.630119	1.00
	DFNN	883.65	5190366.5	2278.237549	15.36	0.672840
	LSTM	882.722168	5068792.0	2251.397705	14.835999	0.672131
	CONV1D	895.317444	5165272.5	2272.723633	14.668820	0.681722
	GRU	1309.246216	13459805.0	3668.760742	16.597067	0.996900
Horizon = 2	NAÏVE	1310.170532	10987072.0	1468.379028	18.548323	1.000000
	DFNN	1044.293945	6764670.5	1167.943481	17.072536	0.797067
	LSTM	1100.123779	7947832.5	1210.607788	17.667553	0.839680
	CONV1D	1094.034424	7990885.0	1201.172241	18.660189	0.835032
	GRU	1682.005859	18657478.0	1828.220581	22.052633	1.283807
Horizon = 3	NAÏVE	1323.616577	11097612.0	1528.525635	18.537531	1.000000
	DFNN	1091.961792	7873397.5	1246.030762	18.579967	0.824983
	LSTM	1153.723633	8594721.0	1301.843872	18.925718	0.871645
	CONV1D	1156.255981	8128375.0	1325.600708	18.581030	0.873558
	GRU	1843.574829	19850782.0	2044.575317	26.012411	1.392831
Horizon = 4	NAÏVE	1317.829956	11102246.0	1555.318726	18.570448	1.000000
	DFNN	1127.714966	7712999.0	1320.954102	19.878557	0.855736
	LSTM	1106.666870	7346445.5	1298.744629	19.927185	0.839765
	CONV1D	1203.325928	9252538.0	1399.098877	20.846319	0.913112
	GRU	1899.683960	21188162.0	2137.000244	27.994749	1.441524
Horizon = 5	NAÏVE	1347.910400	11647780.0	1625.956177	18.580898	1.000000
	DFNN	1211.099121	8487647.0	1430.666016	20.933155	0.898501
	LSTM	1248.208374	8969898.0	1469.298828	21.605707	0.926032
	CONV1D	1183.589111	7700390.5	1404.270996	21.439163	0.878092
	GRU	1981.558228	22565950.0	2287.288086	30.101984	1.470096
Horizon = 6	NAIVE	1375.160767	12073528.0	1676.785767	18.599396	1.000000
	DFNN	1342.517090	10676143.0	1606.338135	23.056026	0.976262
	LSTM	1304.960693	10187365.0	1594.224121	22.217001	0.948951
	CONV1D	1338.717163	11226568.0	1641.417603	23.464102	0.973499
	GRU	2051.931641	24781816.0	2358.406006	29.974390	1.492139
Horizon = 7	NAIVE	1367.746826	11978104.0	1672.518311	18.651089	1.000000
	DFNN	1323.395874	9491738.0	1601.843018	23.901087	0.967574
	LSTM	1342.807983	10784328.0	1634.744873	24.035267	0.981767
	CONV1D	1357.122925	10164708.0	1615.185669	24.194988	0.992232
	GRU	1874.111572	19125806.0	2195.093994	29.963703	1.370218
Horizon = 8	NAIVE	1434.678833	12928787.0	1759.389526	18.664747	1.000000
	DFNN	1510.288208	12948476.0	1831.603149	26.122417	1.052701
	LSTM	1492.555786	12930806.0	1793.618896	25.757946	1.040341
	CONV1D	1510.883789	13548019.0	1843.132446	25.796734	1.053116
	GRU	1854.183716	19952378.0	2163.464355	28.543400	1.292403
Horizon = 9	NAIVE	1422.790161	12776653.0	1743.760742	18.694855	1.000000
	DFNN	1536.508301	13578925.0	1828.404053	25.973475	1.079926
	LSTM	1525.955688	13263729.0	1829.734375	26.705345	1.072509
	CONV1D	1541.007324	13295836.0	1873.072998	27.124849	1.083088
	GRU	1841.795410	19304114.0	2174.532227	29.647469	1.294495
Horizon = 10	NAIVE	1445.514038	13118028.0	1776.539062	18.699409	1.000000
	DFNN	1696.067627	15974404.0	2030.316162	28.503838	1.173332
	LSTM	1618.077026	14907396.0	1944.740234	27.373291	1.119378
	CONV1D	1643.845703	15232460.0	1975.722046	26.785664	1.137205
	GRU	2055.106934	22788930.0	2423.059082	30.667458	1.421713
Horizon = 11	NAIVE	1466.781494	13417188.0	1807.333862	18.696201	1.000000
	DFNN	1673.251099	14484569.0	1995.095581	27.811123	1.140764
	LSTM	1752.379883	17315262.0	2078.024414	28.544985	1.194711
	CONV1D	1683.078125	16481523.0	2030.381348	28.125175	1.147464
	GRU	2100.281982	24985832.0	2513.043213	31.594790	1.431898
Horizon = 12	NAIVE	1451.733765	13230539.0	1792.781494	18.717451	1.000000
	DFNN	1768.781738	17602090.0	2115.720459	28.153948	1.218393
	LSTM	1745.918701	18106658.0	2107.490967	28.356607	1.202644
	CONV1D	1757.592285	18310492.0	2102.190918	28.323296	1.210685
	GRU	2164.490967	25652494.0	2598.680664	34.445511	1.490970
Horizon = 13	NAIVE	1473.809937	13521798.0	1825.368042	18.723145	1.000000
	DFNN	1846.863159	21246070.0	2245.180664	28.846254	1.253122
	LSTM	1790.528320	18113022.0	2166.532227	29.457363	1.214898
	CONV1D	1826.008911	19167842.0	2191.894287	28.528379	1.238972
	GRU	2324.121338	29955354.0	2758.261719	34.750858	1.576948
Horizon = 14	NAIVE	0.006651	0.000182	0.008408	19.087564	1.000000
	DFNN	0.021005	0.001440	0.037941	146.38478	1.160628
	LSTM	0.007741	0.000178	0.009360	29.057104	1.163898
	CONV1D	0.009003	0.000215	0.010833	36.719528	1.353781
	GRU	0.010261	0.000241	0.012397	73.416748	1.542806

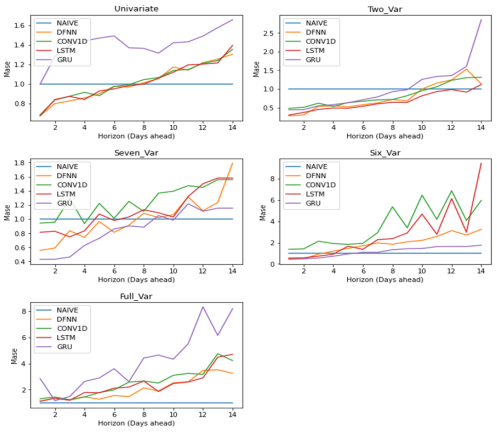

Fig. 10. Mean Absolute Scaled Error (MASE) for the NAÏVE, DFNN, CONV1D, LSTM, and GRU models across various datasets: Scenario 1 (Univariate), Scenario 2 (Two_Variable), Scenario 3 (Seven_Variable), Scenario 4 (Six_Variable), and Scenario 5 (Full_Variable).

To visually assess the efficacy of the top-performing models generated by each algorithm in predicting outcomes using both testing data and data from the latest horizon, the subsequent section presents visual representations of model performance solely for h = 3 and h = 7, which stand as representatives for other horizons. Fig. 11a illustrates the performance of deep learning (DL) models on the testing data employing Scenario 1 input with a horizon of 3. On the other hand, Fig. 11b illustrates the DL models' performance in forecasting for the final three days (h = 3).

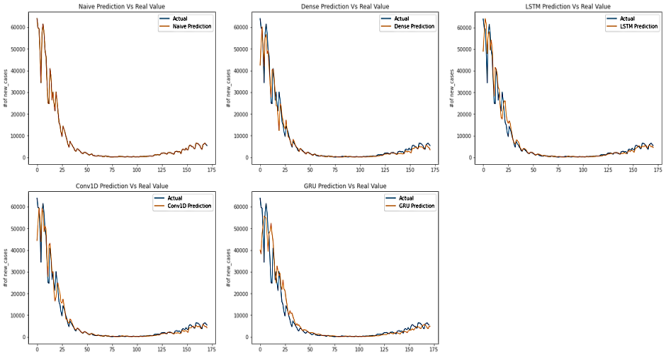

Fig. 11a Performance of DL models on the testing dataset with a horizon of 3, utilizing Scenario 1 input

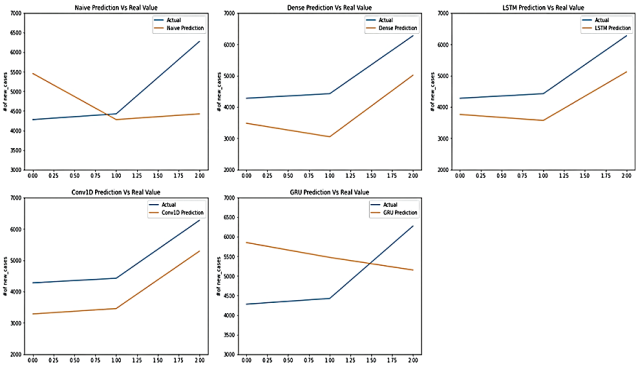

Fig. 11b. Performance of DL models for the final horizon (h = 3) utilizing Scenario 1

Additionally, Fig. 12a showcases the performance of DL models on the testing data, employing Scenario 1 input with a horizon of 7. Meanwhile, Fig. 12b illustrates the performance of DL models in forecasting the preceding seven days (h = 7).

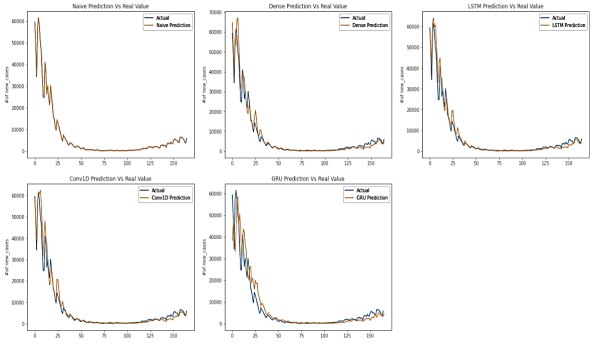

Fig. 12a Performance of DL models on the testing dataset with a horizon of 7, utilizing Scenario 1 input

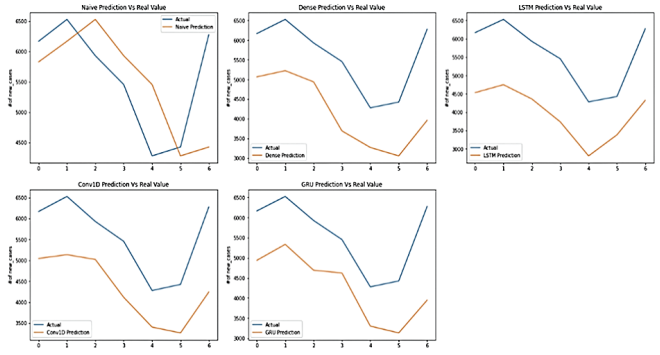

Fig. 12b. Performance of DL models for the final horizon (h = 7) employing Scenario 1

4.2 Multivariate Experimental Results

In contrast to the univariate analysis, multivariate experiments incorporated scaling and variable filtering techniques. The Minimax scaling method utilized in this investigation was confined within the range [0.1]. Variable selection was conducted using the XGB Regressor. Findings from these experiments revealed that, irrespective of the value of h, two predictors, namely new_cases and Rt_ave, consistently ranked among the top three, as demonstrated in Figures 13–15. These figures delineate the weights attributed to the ten most influential predictors for h values of 1, 14, and 21. Based on these observations, a series of four additional experiments were undertaken, resulting in a total of five distinct scenarios.

Scenario 1 is expounded upon in Subsection 5.1.

Scenario 2 entails the utilization of two predictors: new_cases and Rt_ave.

Scenario 3 involves the integration of seven predictors: new_cases, Rt_Ave, C8EV_International travel controls, new_cases_smoothed, population_vaccinated, StringencyIndex_weighted_average, and C5E_closed public transport.

Scenario 4 encompasses all 25 predictors.

Scenario 5 incorporates all predictors included in Scenario 3, with the exclusion of Rt_Ave.

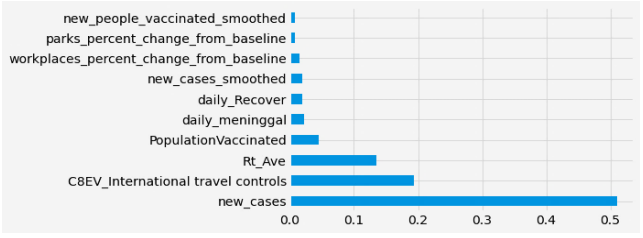

Fig. 13. Ten most significant features determined through XGB Regressor analysis for the period h = 7

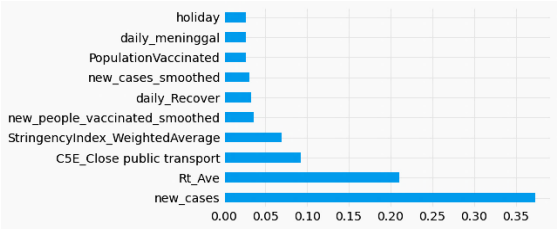

Fig. 14. Ten most influential features computed via XGB Regressor analysis for the duration h = 14

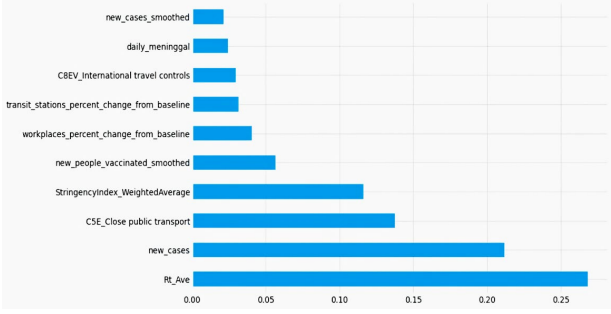

Fig. 15. Ten most significant features determined through XGB Regressor analysis for the period h = 21

A comprehensive outline of the experimental configuration is condensed and presented in the following Table 3.

Table 3
Overview of Experimental Scenarios

Input Scenario	Type	Number of Var /Predictors	Variables/ Predictors Name
Scenario 1	Univariate	1	new_cases
Scenario 2	Multivariate	2	new_cases, Rt_Ave
Scenario 3	Multivariate	6	new_cases,C8EV_International travel controls, new_cases_smoothed, population_vaccinated, StringencyIndex_WeightedAverage, C5E_Close Public Transport
Scenario 4	Multivariate	7	new_cases, Rt_Ave, C8EV_International travel controls, new_cases_smoothed, population_vaccinated, StringencyIndex_WeightedAverage, C5E_Close Public Transport
Scenario 5	Multivariate	25	new_cases, daily_Recover, daily_meninggal,C1E_School closing, C2E_Workplace closing,C3E_Cancel public events, C5E_Close public transport, C6E_Stay at home requirements, C7M_Restrictions on internal movement, C8EV_International travel controls, H2E_Testing policy, H3E_Contact tracing, H6E_Facial Coverings,H8E_Protection of elderly people, PopulationVaccinated,StringencyIndex_WeightedAverage, grocery_and_pharmacy_percent_change_from_baseline, parks_percent_change_from_baseline, transit_stations_percent_change_from_baseline, workplaces_percent_change_from_baseline, residential_percent_change_from_baseline, new_cases_smoothed, new_people_vaccinated_smoothed, holiday, Rt_Ave

Table 4

Experimental Findings Utilizing Multivariate Predictors from Horizon 1 to Horizon 14

Models/	Two Predictors					Six Predictors					Seven Predictors					Full (25) Predictors				
Horizon	MAE	MSE	RMSE	MAPE	MASE	MAE	MSE	RMSE	MAPE	MASE	MAE	MSE	RMSE	MAPE	MASE	MAE	MSE	RMSE	MAPE	MASE
Horizon =1																				
NAIVE	0.011	0.001	0.011	19.000	1.000	0.013	0.001	0.015	19.218	1.000	0.015	0.001	0.015	19.232	1.000	0.007	0.000	0.014	18.959	1.000
DFNN	0.003	0.000	0.003	5.379	0.283	0.006	0.001	0.007	9.355	0.487	0.008	0.001	0.008	10.708	0.557	0.008	0.000	0.016	21.515	1.135
LSTM	0.003	0.000	0.003	5.744	0.302	0.007	0.001	0.008	10.672	0.555	0.012	0.001	0.012	15.645	0.813	0.008	0.000	0.011	35.698	1.121
CONV1D	0.005	0.000	0.005	9.186	0.483	0.018	0.002	0.020	26.452	1.376	0.014	0.001	0.014	18.127	0.943	0.010	0.000	0.014	35.669	1.313
GRU	0.005	0.000	0.005	8.468	0.446	0.006	0.001	0.006	8.306	0.432	0.006	0.001	0.006	8.275	0.430	0.021	0.001	0.031	102.151	2.853
Horizon =2																				
NAIVE	0.011	0.001	0.012	19.078	1.000	0.014	0.001	0.016	19.238	1.000	0.014	0.001	0.016	19.238	1.000	0.009	0.000	0.010	19.000	1.000
DFNN	0.006	0.000	0.011	28.411	0.305	0.010	0.000	0.016	52.353	0.487	0.012	0.000	0.018	40.497	0.591	0.012	0.001	0.013	25.559	1.345
LSTM	0.004	0.000	0.005	9.384	0.372	0.008	0.000	0.009	20.931	0.566	0.012	0.000	0.012	31.655	0.827	0.012	0.000	0.013	60.832	1.354
CONV1D	0.006	0.000	0.006	17.769	0.508	0.020	0.001	0.021	91.096	1.416	0.014	0.000	0.014	46.646	0.956	0.013	0.000	0.013	49.355	1.447
GRU	0.005	0.000	0.005	30.293	0.449	0.007	0.000	0.007	31.600	0.479	0.006	0.000	0.006	26.134	0.432	0.011	0.000	0.011	71.053	1.169
Horizon = 3																				
NAIVE	0.011	0.001	0.012	19.039	1.000	0.014	0.001	0.016	19.180	1.000	0.014	0.001	0.016	19.180	1.000	0.014	0.001	0.016	19.224	1.000
DFNN	0.006	0.000	0.006	10.182	0.535	0.018	0.001	0.029	37.326	0.918	0.011	0.001	0.013	16.040	0.836	0.017	0.002	0.020	23.327	1.213
LSTM	0.005	0.000	0.006	15.184	0.453	0.010	0.000	0.011	48.115	0.730	0.010	0.000	0.011	39.782	0.748	0.017	0.001	0.018	89.018	1.198
CONV1D	0.007	0.000	0.007	22.441	0.620	0.029	0.002	0.031	78.987	2.133	0.018	0.001	0.019	37.434	1.290	0.018	0.001	0.019	103.189	1.240
GRU	0.006	0.000	0.006	34.339	0.549	0.008	0.000	0.009	27.808	0.557	0.006	0.000	0.008	25.850	0.463	0.021	0.001	0.023	87.128	1.485
Horizon = 4																				
NAIVE	0.010	0.001	0.012	19.046	1.000	0.013	0.001	0.016	19.195	1.000	0.013	0.001	0.016	19.195	1.000	0.008	0.000	0.010	18.993	1.000
DFNN	0.006	0.000	0.007	10.446	0.548	0.024	0.002	0.040	43.040	1.198	0.015	0.001	0.023	48.920	0.740	0.012	0.000	0.014	27.611	1.454
LSTM	0.005	0.000	0.006	22.283	0.490	0.012	0.000	0.014	32.131	0.910	0.011	0.000	0.012	34.094	0.834	0.015	0.000	0.016	98.823	1.789
CONV1D	0.006	0.000	0.007	18.910	0.533	0.026	0.002	0.029	62.509	1.917	0.013	0.001	0.014	39.942	0.936	0.012	0.000	0.013	60.624	1.456
GRU	0.006	0.000	0.007	38.970	0.582	0.010	0.000	0.011	45.177	0.721	0.008	0.000	0.010	35.076	0.627	0.022	0.001	0.024	114.597	2.630
Horizon = 5																				
NAIVE	0.010	0.001	0.012	19.059	1.000	0.013	0.001	0.015	19.190	1.000	0.013	0.001	0.015	19.190	1.000	0.008	0.000	0.010	19.007	1.000
DFNN	0.011	0.001	0.025	49.687	0.524	0.029	0.003	0.057	73.851	1.458	0.019	0.001	0.026	55.558	0.966	0.025	0.001	0.038	129.886	1.280
LSTM	0.005	0.000	0.006	17.400	0.485	0.022	0.001	0.026	43.662	1.669	0.014	0.000	0.015	46.069	1.072	0.015	0.000	0.015	48.833	1.770
CONV1D	0.006	0.000	0.008	28.849	0.635	0.024	0.002	0.025	42.091	1.840	0.016	0.001	0.018	40.242	1.222	0.015	0.001	0.016	75.208	1.783
GRU	0.006	0.000	0.007	43.407	0.632	0.012	0.001	0.014	46.581	0.935	0.009	0.000	0.011	37.602	0.728	0.024	0.001	0.027	123.807	2.899
Horizon = 6																				
NAIVE	0.010	0.000	0.012	19.086	1.000	0.013	0.001	0.015	19.211	1.000	0.013	0.001	0.015	19.211	1.000	0.008	0.000	0.010	19.016	1.000
DFNN	0.012	0.001	0.027	34.276	0.579	0.034	0.004	0.063	242.584	1.704	0.016	0.001	0.026	72.402	0.813	0.030	0.002	0.050	305.668	1.565
LSTM	0.005	0.000	0.006	15.018	0.538	0.017	0.001	0.020	37.944	1.371	0.012	0.000	0.014	55.841	0.980	0.017	0.001	0.019	94.535	2.121
CONV1D	0.007	0.000	0.008	29.780	0.672	0.025	0.002	0.028	83.145	1.930	0.013	0.000	0.014	37.057	1.011	0.016	0.001	0.018	83.932	1.995
GRU	0.007	0.000	0.008	40.673	0.712	0.014	0.001	0.016	43.889	1.093	0.011	0.001	0.013	45.810	0.860	0.029	0.002	0.033	130.806	3.608
Horizon = 7																				
NAIVE	0.010	0.000	0.012	19.108	1.000	0.012	0.001	0.015	19.218	1.000	0.012	0.001	0.015	19.218	1.000	0.008	0.000	0.009	19.047	1.000
DFNN	0.013	0.001	0.028	31.696	0.629	0.040	0.004	0.065	301.956	1.970	0.018	0.001	0.034	38.575	0.914	0.028	0.003	0.050	157.379	1.473
LSTM	0.006	0.000	0.007	22.579	0.602	0.028	0.002	0.032	56.360	2.278	0.013	0.000	0.014	40.258	1.030	0.017	0.001	0.019	78.746	2.205
CONV1D	0.007	0.000	0.008	18.942	0.711	0.036	0.004	0.039	60.510	2.936	0.015	0.001	0.017	43.330	1.250	0.020	0.001	0.021	138.315	2.583
GRU	0.008	0.000	0.009	47.691	0.789	0.013	0.001	0.016	58.797	1.087	0.011	0.000	0.013	45.725	0.903	0.020	0.001	0.024	103.515	2.615
Horizon = 8																				
NAIVE	0.009	0.000	0.011	19.114	1.000	0.012	0.001	0.014	19.215	1.000	0.012	0.001	0.014	19.215	1.000	0.008	0.000	0.009	19.047	1.000
DFNN	0.014	0.001	0.026	121.137	0.719	0.037	0.006	0.078	74.341	1.848	0.013	0.001	0.016	20.801	1.083	0.041	0.005	0.073	175.603	2.137
LSTM	0.006	0.000	0.007	21.561	0.644	0.029	0.003	0.032	65.234	2.373	0.014	0.000	0.015	56.356	1.132	0.020	0.001	0.022	66.514	2.666
CONV1D	0.007	0.000	0.008	26.636	0.718	0.064	0.016	0.069	171.825	5.363	0.013	0.000	0.015	54.424	1.109	0.020	0.001	0.021	173.140	2.662
GRU	0.009	0.000	0.010	58.540	0.928	0.016	0.001	0.018	62.016	1.334	0.011	0.001	0.013	49.246	0.885	0.034	0.002	0.040	156.622	4.420

Table 4
Experimental Findings Utilizing Multivariate Predictors from Horizon 1 to Horizon 14 (Continued)

Models/	Two Predictors					Six Predictors					Seven Predictors					Full (25) Predictors				
Horizon = 9																				
NAIVE	0.009	0.000	0.011	19.120	1.000	0.012	0.001	0.014	19.215	1.000	0.012	0.001	0.014	19.215	1.000	0.007	0.000	0.009	19.050	1.000
DFNN	0.013	0.001	0.029	46.204	0.684	0.042	0.008	0.087	177.116	2.074	0.021	0.001	0.033	46.747	1.020	0.037	0.004	0.062	292.487	1.905
LSTM	0.006	0.000	0.007	21.124	0.642	0.034	0.003	0.040	103.095	2.905	0.013	0.000	0.015	44.175	1.089	0.014	0.000	0.015	80.244	1.865
CONV1D	0.008	0.000	0.009	38.339	0.821	0.040	0.005	0.042	80.469	3.390	0.016	0.001	0.018	57.928	1.367	0.019	0.001	0.020	84.198	2.514
GRU	0.009	0.000	0.011	57.565	0.979	0.017	0.001	0.019	81.156	1.430	0.012	0.001	0.014	53.989	1.050	0.035	0.003	0.041	178.788	4.644
Horizon =10																				
NAIVE	0.007	0.000	0.009	19.036	1.000	0.011	0.001	0.014	19.211	1.000	0.011	0.001	0.014	19.211	1.000	0.007	0.000	0.009	19.036	1.000
DFNN	0.007	0.000	0.009	19.032	1.000	0.045	0.007	0.082	99.358	2.209	0.021	0.001	0.033	59.934	1.062	0.046	0.005	0.072	302.980	2.525
LSTM	0.006	0.000	0.007	20.142	0.819	0.053	0.008	0.060	120.734	4.681	0.012	0.000	0.013	31.654	1.028	0.018	0.001	0.019	67.285	2.475
CONV1D	0.007	0.000	0.008	31.513	0.954	0.074	0.010	0.078	362.595	6.452	0.016	0.001	0.018	51.338	1.394	0.022	0.001	0.024	147.697	3.100
GRU	0.009	0.000	0.011	61.389	1.258	0.017	0.001	0.019	65.090	1.446	0.011	0.000	0.013	65.924	0.984	0.031	0.003	0.036	117.357	4.332
Horizon = 11																				
NAIVE	0.007	0.000	0.009	19.037	1.000	0.011	0.001	0.013	19.208	1.000	0.011	0.001	0.013	19.208	1.000	0.007	0.000	0.009	19.037	1.000
DFNN	0.008	0.000	0.010	22.056	1.159	0.052	0.007	0.085	82.521	2.579	0.015	0.001	0.018	25.205	1.312	0.047	0.009	0.096	231.427	2.603
LSTM	0.007	0.000	0.008	22.445	0.932	0.031	0.002	0.034	199.341	2.782	0.015	0.000	0.016	56.658	1.319	0.018	0.001	0.020	101.385	2.593
CONV1D	0.007	0.000	0.009	32.087	1.060	0.047	0.007	0.051	162.983	4.189	0.016	0.001	0.018	37.204	1.473	0.023	0.001	0.027	69.810	3.248
GRU	0.009	0.000	0.011	70.505	1.337	0.018	0.002	0.021	78.724	1.623	0.014	0.000	0.016	65.521	1.217	0.039	0.004	0.044	167.970	5.500
Horizon = 12																				
NAIVE	0.007	0.000	0.009	19.047	1.000	0.011	0.001	0.013	19.214	1.000	0.011	0.001	0.013	19.214	1.000	0.007	0.000	0.009	19.047	1.000
DFNN	0.023	0.002	0.042	100.999	1.244	0.063	0.014	0.118	434.306	3.127	0.023	0.001	0.032	99.505	1.117	0.063	0.016	0.128	620.330	3.471
LSTM	0.007	0.000	0.008	25.564	0.984	0.066	0.011	0.075	168.353	6.124	0.016	0.001	0.018	43.240	1.499	0.020	0.001	0.022	98.659	2.906
CONV1D	0.009	0.000	0.010	33.640	1.233	0.074	0.011	0.081	207.281	6.863	0.016	0.001	0.018	46.246	1.448	0.022	0.001	0.025	159.878	3.175
GRU	0.009	0.000	0.011	68.555	1.359	0.018	0.001	0.021	80.019	1.637	0.012	0.000	0.015	58.251	1.111	0.058	0.006	0.066	305.848	8.327
Horizon = 13																				
NAIVE	0.007	0.000	0.009	19.064	1.000	0.011	0.001	0.013	19.226	1.000	0.011	0.001	0.013	19.226	1.000	0.007	0.000	0.009	19.064	1.000
DFNN	0.028	0.003	0.052	81.221	1.530	0.052	0.013	0.114	120.315	2.706	0.013	0.001	0.016	23.735	1.235	0.064	0.024	0.154	350.805	3.523
LSTM	0.006	0.000	0.008	25.641	0.916	0.031	0.002	0.035	79.684	2.974	0.017	0.001	0.019	70.354	1.581	0.030	0.003	0.034	98.938	4.486
CONV1D	0.009	0.000	0.010	48.085	1.303	0.043	0.006	0.048	97.811	4.074	0.016	0.001	0.018	42.317	1.558	0.032	0.002	0.035	201.634	4.745
GRU	0.011	0.000	0.013	70.769	1.603	0.017	0.001	0.020	79.098	1.631	0.012	0.000	0.015	63.526	1.154	0.042	0.004	0.049	186.452	6.144
Horizon = 14																				
NAIVE	0.007	0.000	0.014	18.959	1.000	0.010	0.001	0.013	19.239	1.000	0.011	0.001	0.013	19.226	1.000	0.007	0.000	0.008	19.088	1.000
DFNN	0.008	0.000	0.016	21.515	1.135	0.063	0.018	0.135	359.546	3.248	0.019	0.001	0.023	34.408	1.790	0.059	0.015	0.122	403.080	3.260
LSTM	0.008	0.000	0.011	35.698	1.121	0.097	0.028	0.111	223.660	9.444	0.017	0.001	0.019	70.354	1.581	0.031	0.002	0.035	225.961	4.702
CONV1D	0.010	0.000	0.014	35.669	1.313	0.061	0.009	0.069	320.693	5.957	0.016	0.001	0.018	42.317	1.558	0.028	0.002	0.030	132.534	4.226
GRU	0.021	0.001	0.031	102.151	2.853	0.018	0.002	0.021	82.549	1.762	0.012	0.000	0.015	63.526	1.154	0.054	0.007	0.064	252.104	8.180

Fig. 10 (10b–10e) illustrates the Mean Absolute Scaled Error (MASE) comparison across all comprehensive multivariate experiments. In order to offer a comprehensive assessment of the predictive capabilities of the models utilizing Scenario 2 input, Fig. 16 displays the models' performance on the test dataset for h = 7, while Fig. 17 showcases their performance evaluated on the final horizon for h = 7.

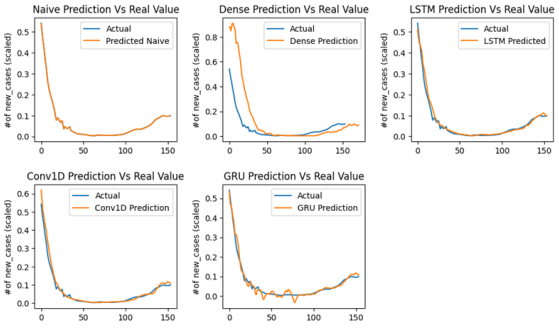

Fig. 16. Prediction Performance of All Models for h = 7 on the Entire Testing Dataset Utilizing Scenario 2

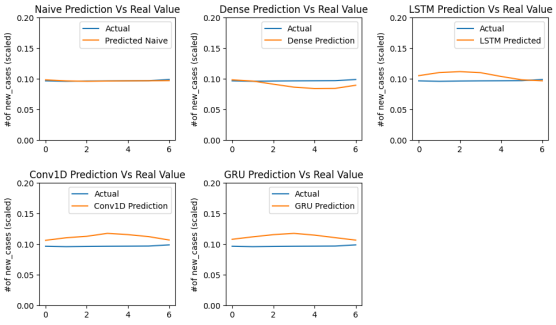

Fig. 17. Prediction Performance of All Models for the Final Seven Days Utilizing Scenario 2 Predictors

To offer a comprehensive assessment of the predictive capacity of the models employing the Scenario 3 input, Fig. 18 depicts the performance of the models on the test dataset for h = 7, while Fig. 19 demonstrates their performance evaluated on the final horizon for h = 7.

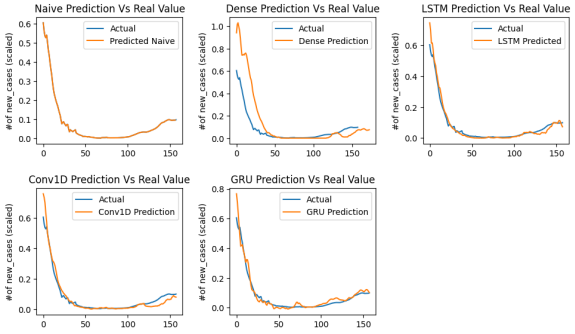

Fig. 18. Prediction Performance of All Models for h = 7 on the Entire Testing Dataset Utilizing Scenario 3

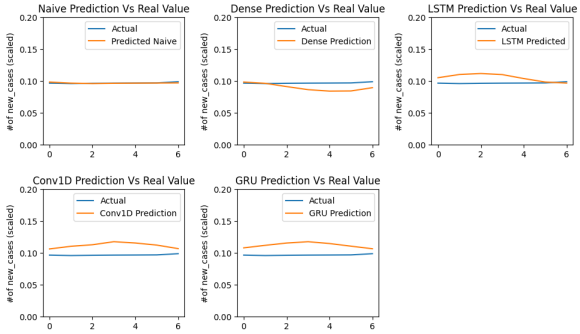

Fig. 19. Prediction Performance of All Models for the Final Seven Days Utilizing Scenario 3 Predictors

In order to present a comprehensive evaluation of the predictive capability of the models utilizing the Scenario 4 input, Fig. 20 illustrates the performance of the models on the test dataset for h = 7, while Fig. 21 demonstrates their performance assessed on the final horizon for h = 7.

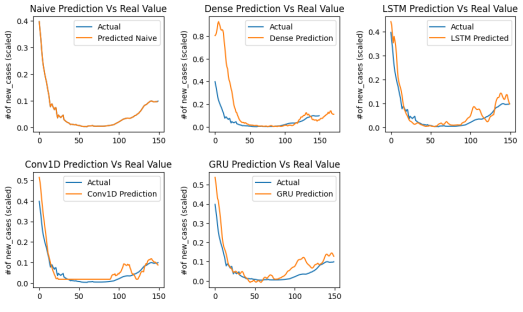

Fig. 20. Prediction Performance of All Models for h = 7 on the Entire Testing Dataset Utilizing Scenario 4

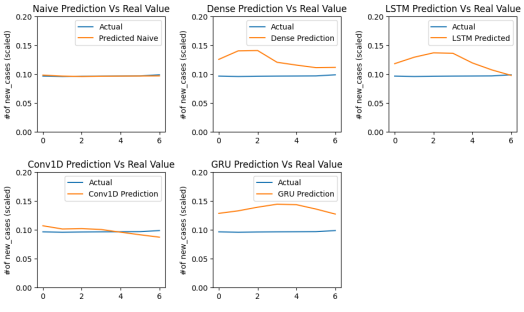

Fig. 21. Prediction Performance of All Models for the Final Seven Days Utilizing Scenario 4 Predictors

5. Discussion

This section highlights several key points derived from a series of experiments and subsequent analysis of the results. Firstly, it is imperative to investigate whether the dataset exhibits specific patterns conducive to estimating various aspects of time series data. Scatter diagrams depicting time series with varying lags offer an initial informative depiction prior to modeling. Fig. 3 provides insight into the temporal relationships within the data, where smaller lags denote greater information sharing. Furthermore, the Partial Autocorrelation Function (PACF) analysis, as depicted in Figure 4, yields equally crucial insights. Beyond lag = 1, Figure 4 distinctly reveals a substantial correlation within the Covid-19 time series data in Indonesia, notably with a lag of eight days. This correlation suggests a close association between present case numbers and those occurring eight days prior or subsequent. Further investigation is warranted to ascertain whether this phenomenon is linked to the virus's incubation period. Based on the insights gleaned from the Scatter diagrams and PACF analysis, we infer that the naïve model exhibits sufficient strength in predicting Covid-19 new cases one day in advance. Any model capable of enhancing the accuracy of this prediction is deemed suitable for real-world forecasting scenarios. However, it is noteworthy that by its inherent nature, the naïve model is unsuitable for predicting real-world problems with a horizon equal to or greater than 2. The DNN-based model and its DL variant demonstrated superior performance compared to the Naive model. In univariate scenarios, the prediction of future cases relies solely on the present or past case counts, limited by the window size. As depicted in Figure10a, DL models exhibit proficient forecasting capabilities for up to 7 days ahead. Notably, the figure indicates that at least one algorithm achieves a Mean Absolute Scaled Error (MASE) value below 1.0, indicating superior performance relative to the Naive model. With the exception of GRU, all DL algorithms exhibited similar performance, as evidenced by the proximity of their respective curves in Figure10a.

Experiments conducted with multivariate time-series data demonstrate the potential for enhancing the capabilities of DFNN and DL models. This enhancement necessitates the incorporation of supplementary information during the model training phase. However, the selection of additional variables for model inclusion must be approached with caution. Findings from these experiments indicate that augmenting models with more variables does not invariably enhance their predictive efficacy. As illustrated in Figure 10d, the inclusion of all variables in the model results in diminished performance, even when trained under identical conditions and hyperparameters. Despite variations in selected horizons, none of the models succeeded in surpassing the accuracy achieved by the Naive model. The subsequent challenge revolves around the selection of suitable variables for incorporation into the model. Utilizing an XGB Regressor for variable selection in this context yields effective outcomes. Findings from the five experimental scenarios indicate that employing only two variables as inputs yields optimal results. The inclusion of these two variables enhances the predictive capacity of DL models from seven days (Figure 10a) to twelve days (Figure 10b) ahead. Once again, the performance of the three DL algorithms appeared to be comparable to that of LSTM, with the latter exhibiting slightly superior performance. However, this predictive capability diminishes with the addition of more variables to the model (Figure 10c). Under this scenario, DL models can predict up to eight days ahead using GRU, demonstrating superior performance compared to other algorithms. Furthermore, our experiments underscored the significance of Rt in enhancing prediction accuracy, as previously discussed. The inclusion of Rt alongside daily case counts notably augmented the predictive capabilities of DL-based models. In univariate scenarios where Rt was not accounted for, DL models were limited to predicting the subsequent seven days. However, with Rt incorporated into the model in a 2-variable multivariate scenario, DL predictions extended to 13 days. Likewise, in the 7-variable scenario, DL could forecast up to eight days ahead. Conversely, when Rt was excluded in the 6-variable scenario, DL predictions were constrained to the following five days. Ultimately, our experiments unveiled an unexpected discovery: governmental measures appeared ineffective in stemming the disease's spread, and community mobility behavior seemed to have no discernible impact on daily case counts. However, this conclusion was deemed unsatisfactory and imprecise. We postulated that the functions of all aforementioned variables might have been subsumed within the Rt value, which has been demonstrated to notably enhance the predictive capacity of DL models. Consequently, Rt emerges as a mediating variable encompassing all independent variables examined in this study.

6. Conclusion and Future Research

Since early 2020, human life has faced widespread disruptions across the globe due to the Covid-19 pandemic, exerting a profound impact on various facets of existence. Comprehending the dynamics of Covid-19 transmission within a given area holds significant importance in mitigating and forestalling the adverse consequences of this pandemic. In this investigation, deep learning methodologies were employed by researchers to forecast the forthcoming number of active Covid-19 cases over the following days. The experiments were conducted across diverse input scenarios tailored to available data inputs, yielding several notable insights. Deep learning techniques exhibited the capability to forecast new daily cases for a duration extending up to 13 days, evaluated

relative to the predictive capacity of a naive model. The efficacy of deep learning methodologies varied contingent upon the specific algorithm employed and the input variables provided. Notably, the investigation revealed that the count of daily cases and the Rt value at a given juncture played pivotal roles in predicting the number of active cases in the ensuing days. Consequently, avenues for further research can be explored in several directions:

1. Further investigation may entail conducting analogous experiments utilizing alternative deep learning algorithms, including but not limited to the bidirectional variant. Additionally, extensive hyperparameter tuning remains an area that warrants exploration to ascertain optimal model configurations.
2. Beyond the prediction of new cases, the target variable could be diversified to include other pertinent factors directly influencing pandemic mitigation efforts. These factors may encompass metrics such as Bed Occupancy Ratio (BOR), mortality rates, requirements for intensive care unit (ICU) admissions, and availability of isolation facilities.
3. The elucidation of Rt's role in enhancing the predictive capacity of DL models for active case prediction represents another significant finding of this study. However, the authors advocate for the exploration of Rt prediction, positing that governmental interventions and community mobility behaviors likely exert influence on Rt values, thereby impacting disease transmission dynamics.

Acknowledgment

The authors extend their sincere gratitude to Bina Nusantara University for the invaluable support extended throughout this research endeavor under Initiative Project number 189/Proyek. Inisiatif/XII/2021 Project Code: PID33189.

References

Abolmaali, S., & Shirzaei, S. (2021). A comparative study of SIR Model, Linear Regression, Logistic Function and ARIMA Model for forecasting COVID-19 cases. *AIMS Public Health*, *8*(4), 598–613. https://doi.org/10.3934/publichealth.2021048

Absar, N., Uddin, N., Khandaker, M. U., & Ullah, H. (2022). The efficacy of deep learning based LSTM model in forecasting the outbreak of contagious diseases. *Infectious Disease Modelling*, *7*(1), 170–183. https://doi.org/10.1016/j.idm.2021.12.005

Alassafi, M. O., Jarrah, M., & Alotaibi, R. (2022). Time series predicting of COVID-19 based on deep learning. *Neurocomputing*, *468*, 335–344. https://doi.org/10.1016/j.neucom.2021.10.035

Blavatnik School of Government. (2022). *COVID-19 Government Response Tracker*. https://www.bsg.ox.ac.uk/research/covid-19-government-response-tracker

Bsat, R., Chemaitelly, H., Coyle, P., Tang, P., Hasan, M. R., Kanaani, Z. Al, Kuwari, E. Al, Butt, A. A., Jeremijenko, A., Kaleeckal, A. H., Latif, A. N., Shaik, R. M., Nasrallah, G. K., Benslimane, F. M., Khatib, H. A. A., Yassine, H. M., Kuwari, M. G. A., Romaihi, H. E. Al, Al-Thani, M. H., … Ayoub, H. H. (2022). Characterizing the effective reproduction number during the COVID-19 pandemic: Insights from Qatar's experience. *Journal of Global Health*, *12*. https://doi.org/10.7189/JOGH.12.05004

Cinaglia, P., & Cannataro, M. (2022). Forecasting COVID-19 Epidemic Trends by Combining a Neural Network with Rt Estimation. *Entropy*, *24*(7), 1–17. https://doi.org/10.3390/e24070929

Cortés-Carvajal, P. D., Cubilla-Montilla, M., & González-Cortés, D. R. (2022). Estimation of the Instantaneous Reproduction Number and Its Confidence Interval for Modeling the COVID-19 Pandemic. *Mathematics*, *10*(2). https://doi.org/10.3390/MATH10020287

Delli Compagni, R., Cheng Id, Z., Russo, S., & Van Boeckel, T. P. (2022). A hybrid Neural Network-SEIR model for forecasting intensive care occupancy in Switzerland during COVID-19 epidemics. *PLoS ONE*, *3*. https://doi.org/10.1371/journal.pone.0263789

Gnanvi, J. E., Salako, K. V., Kotanmi, G. B., & Glèlè Kakaï, R. (2021). On the reliability of predictions on Covid-19 dynamics: A systematic and critical review of modelling techniques. *Infectious Disease Modelling*, *6*, 258–272. https://doi.org/10.1016/j.idm.2020.12.008

Goel, R., Bonnetain, L., Sharma, R., & Furno, A. (2021). Mobility-based SIR model for complex networks: with case study Of COVID-19. *Social Network Analysis and Mining*, *11*(1). https://doi.org/10.1007/S13278-021-00814-3

Google. (2022). *Community Mobility Reports*. https://www.google.com/covid19/mobility/

Inthamoussou, F. A., Valenciaga, F., Núñez, S., & Garelli, F. (2022). Extended SEIR Model for Health Policies Assessment Against the COVID-19 Pandemic: the Case of Argentina. *Journal of Healthcare Informatics Research*, *6*(1), 91–111. https://doi.org/10.1007/s41666-021-00110-x

Jo, W., & Kim, D. (2023). Neural additive time-series models: Explainable deep learning for multivariate time-series prediction. *Expert Systems with Applications*, *228*. https://doi.org/10.1016/j.eswa.2023.120307

John-Otumu, A. M. G., Ikerionwu, C., Olaniyi, O. O., Dokun, O., Eze, U. F., & Nwokonkwo, O. C. (2024). Advancing COVID-19 Prediction with Deep Learning Models: A Review. *International Conference on Science, Engineering and Business for Driving Sustainable Development Goals, SEB4SDG 2024*. https://doi.org/10.1109/SEB4SDG60871.2024.10630186

Kermack, W. O., & McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772), 700-721.

Khalifa, N. E., Mawgoud, A. A., Abu-Talleb, A., Taha, M. H. N., & Zhang, Y. D. (2023). A COVID-19 Infection Prediction Model in Egypt Based on Deep Learning Using Population Mobility Reports. *International Journal of Computational Intelligence Systems*, *16*(1). https://doi.org/10.1007/s44196-023-00272-z

Kim, D., Ali, S. T., Kim, S., Jo, J., Lim, J. S., Lee, S., & Ryu, S. (2022). Estimation of Serial Interval and Reproduction Number to Quantify the Transmissibility of SARS-CoV-2 Omicron Variant in South Korea. *Viruses*, *14*(3). https://doi.org/10.3390/V14030533

Livieris, I. E. (2023). A novel forecasting strategy for improving the performance of deep learning models. *Expert Systems with Applications*, *230*. https://doi.org/10.1016/j.eswa.2023.120632

Locatelli, I., Trächsel, B., & Rousson, V. (2021). Estimating the basic reproduction number for COVID-19 in Western Europe. *PLOS ONE*, *16*(3), e0248731. https://doi.org/10.1371/JOURNAL.PONE.0248731

Lucas, B., Vahedi, B., & Karimzadeh, M. (2023). A spatiotemporal machine learning approach to forecasting COVID-19 incidence at the county level in the USA. *International Journal of Data Science and Analytics*, *15*, 247–266. https://doi.org/10.1007/s41060-021-00295-9

Luqmanul, A., Achmad, H., & Purwani, S. (2021). A Susceptible-Infected-Removed Epidemiological Model for COVID-19 Spreading in Indonesia. *World Scientific News*, *153*(January), 55–64.

Marinov, T. T., & Marinova, R. S. (2022). Inverse problem for adaptive SIR model: Application to COVID-19 in Latin America. *Infectious Disease Modelling*, *7*(1), 134–148. https://doi.org/10.1016/J.IDM.2021.12.001

Masum, M., Masud, M. A., Adnan, M. I., Shahriar, H., & Kim, S. (2022). Comparative study of a mathematical epidemic model, statistical modeling, and deep learning for COVID-19 forecasting and management. *Socio-Economic Planning Sciences*, *80*(December 2021), 101249. https://doi.org/10.1016/j.seps.2022.101249

Nabi, K. N., Tahmid, M. T., Rafi, A., Kader, M. E., & Haider, M. A. (2021). Forecasting COVID-19 cases: A comparative analysis between recurrent and convolutional neural networks. *Results in Physics*, *24*, 104137. https://doi.org/10.1016/j.rinp.2021.104137

Our World in Data. (2022). *Indonesia: Coronavirus Pandemic Country Profile*. https://ourworldindata.org/coronavirus/country/indonesia

Qu, Z., Li, Y., Jiang, X., & Niu, C. (2023). An innovative ensemble model based on multiple neural networks and a novel heuristic optimization algorithm for COVID-19 forecasting. *Expert Systems with Applications*, *212*. https://doi.org/10.1016/j.eswa.2022.118746

Satuan Tugas Penanganan COVID-19. (2022). *Data Sebaran*. https://covid19.go.id/

Shuai, C., Zhao, B., Chen, X., Liu, J., Zheng, C., Qu, S., Zou, J. P., & Xu, M. (2024). Quantifying the impacts of COVID-19 on Sustainable Development Goals using machine learning models. *Fundamental Research*, *4*(4), 890–897. https://doi.org/10.1016/J.FMRE.2022.06.016

Trajanoska, M., Trajanov, R., & Eftimov, T. (2022). Dietary, comorbidity, and geo-economic data fusion for explainable COVID-19 mortality prediction. *Expert Systems with Applications*, *209*. https://doi.org/10.1016/j.eswa.2022.118377

Verma, H., Mandal, S., & Gupta, A. (2022). Temporal deep learning architecture for prediction of COVID-19 cases in India. *Expert Systems with Applications*, *195*. https://doi.org/10.1016/j.eswa.2022.116611

Wathore, R., Rawlekar, S., Anjum, S., Gupta, A., Bherwani, H., Labhasetwar, N., & Kumar, R. (2023). Improving performance of deep learning predictive models for COVID-19 by incorporating environmental parameters. *Gondwana Research*, *114*, 69–77. https://doi.org/10.1016/j.gr.2022.03.014

Wei, W. W. (2018). *Multivariate time series analysis and applications*. John Wiley & Sons.

Wintachai, P., & Prathom, K. (2021). Stability analysis of SEIR model related to efficiency of vaccines for COVID-19 situation. *Heliyon*, *7*(4), e06812. https://doi.org/10.1016/j.heliyon.2021.e06812

Witten, I. H., Frank, E., & Hall, M. A. (2011). Data Mining. In *Data Mining*. https://doi.org/10.1016/C2009-0-19715-5

Xu, L., Magar, R., & Barati Farimani, A. (2022). Forecasting COVID-19 new cases using deep learning methods. *Computers in Biology and Medicine*, *144*, 105342. https://doi.org/10.1016/J.COMPBIOMED.2022.105342

Yang, X., Wang, S., Xing, Y., Li, L., Xu, R. Y. Da, Friston, K. J., & Guo, Y. (2022). Bayesian data assimilation for estimating instantaneous reproduction numbers during epidemics: Applications to COVID-19. *PLoS Computational Biology*, *18*(2). https://doi.org/10.1371/JOURNAL.PCBI.1009807

Zhang, P., Feng, K., Gong, Y., Lee, J., Lomonaco, S., & Zhao, L. (2022). Usage of Compartmental Models in Predicting COVID-19 Outbreaks. *AAPS Journal*, *24*(5), 1–12. https://doi.org/10.1208/s12248-022-00743-9

Zheng, N., Du, S., Wang, J., Zhang, H., Cui, W., Kang, Z., Yang, T., Lou, B., Chi, Y., Long, H., Ma, M., Yuan, Q., Zhang, S., Zhang, D., Ye, F., & Xin, J. (2020). Predicting COVID-19 in China Using Hybrid AI Model. *IEEE Transactions on Cybernetics*, *50*(7), 2891–2904. https://doi.org/10.1109/TCYB.2020.2990162