Comparative Analysis of Machine Learning Models for Forecasting Infectious Disease Spread

Praveen Damacharla[1], Venkata Akhil Kumar Gummadi[2]
Research Scientist, KineticAI Inc., The Woodlands, Texas 77380[1]
Software Developer, KineticAI Inc., The Woodlands, Texas 77380[2]

***Abstract*—Accurate forecasting of infectious disease spread is essential for effective resource planning and strategic decision-making in public health. This study provides a comprehensive evaluation of various machine learning models, from traditional statistical approaches to advanced deep learning techniques, for forecasting disease outbreak dynamics. Focusing on daily positive cases and daily deaths—key indicators despite potential reporting inconsistencies—our analysis aims to identify the most effective models across different algorithm families. By adapting non-time series methods with temporal factors and enriching time series models with exogenous variables, we enhance model suitability for the data's time-dependent nature. Using India as a case study due to its significant early pandemic spread, we evaluate models through metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Median Squared Error (MEME), and Mean Squared Log Error (MSLE). The models tested include Linear Regression, Elastic Net, Random Forest, XGBoost, and Simple Exponential Smoothing, among others. Results indicate that the Random Forest Regressor outperforms other methods in terms of prediction accuracy across most metrics. Notably, findings suggest that simpler models can sometimes match or even exceed the reliability of more complex approaches. However, limitations include model sensitivity to data quality and the lack of real-time adaptability, which may affect performance in rapidly evolving outbreak situations. These insights have critical implications for public health policy and resource allocation in managing infectious disease outbreaks.**

***Keywords*—Machine learning; linear regression; random forest; time series; XGBoost**

I. Introduction

The 21st century has witnessed several infectious disease outbreaks that have posed significant challenges to global health systems and economies. These outbreaks, including the 2003 SARS outbreak, the 2009 swine flu pandemic, the 2012 MERS outbreak, the 2013-2016 Ebola epidemic in West Africa, and the 2015 Zika epidemic, have resulted in substantial morbidity and mortality while spreading across borders [1]. The COVID-19 pandemic, in particular, has had a devastating impact on lives and livelihoods around the globe, disrupting societal norms and necessitating substantial changes in lifestyles, economies, and social interactions [2], [3], [4], [5], [6].

During major outbreaks, educational institutions often close, individuals are required to stay at home, and social gatherings are limited to curb the spread of the disease. Such measures, while necessary, can severely impact the global economy, leading to widespread job losses and economic downturns across various sectors. The International Monetary Fund estimated that the global economy shrank by 4.4% in 2020 due to the COVID-19 pandemic, marking the worst decline since the Great Depression of the 1930s [7]. Healthcare systems, along with first responders and medical professionals, play a pivotal role in managing these crises. Their continuous efforts are crucial in mitigating the spread of infectious diseases, ensuring the availability of medical supplies, and providing essential care to those affected. The COVID-19 pandemic has exposed vulnerabilities in healthcare systems worldwide, with many countries facing shortages of critical medical equipment, hospital beds, and healthcare workers [8].

Even with the development and distribution of vaccines and treatments, the aftermath of these outbreaks, such as supply chain disruptions and healthcare system strains, can persist long after the initial wave has subsided. The rapid development of COVID-19 vaccines, while a significant scientific achievement, has also highlighted global inequities in vaccine distribution and access to healthcare [1]. Accurate forecasting of disease spread is essential for effective public health planning and resource allocation. Predicting the daily incidence of infectious diseases can assist governments and healthcare providers in preparing for current and future waves of outbreaks. Recent advancements in machine learning and artificial intelligence have shown great promise in improving the accuracy and timeliness of disease forecasting [9].

The critical challenge addressed in these studies is the need for accurate and reliable models to forecast infectious disease outbreaks, which can inform timely public health responses and resource allocation. Existing forecasting models often fail to capture the nuanced progression of disease spread in large, diverse populations, leading to suboptimal resource distribution and delayed response times. Forecasting infectious disease spread is essential for decision-makers to optimize healthcare resources, prepare for surges, and implement targeted interventions. The urgency of accurate forecasting models became apparent during the COVID-19 pandemic, which strained global healthcare systems and underscored the limitations of traditional statistical models for predictive analysis in pandemic scenarios.

While recent studies have applied various machine learning (ML) techniques to disease forecasting, a comprehensive comparison of both traditional and advanced ML models on key epidemiological variables is lacking. Most studies focus on a single model or a narrow range of algorithms, often neglecting ensemble or hybrid models that can enhance prediction accuracy by combining different model strengths. This study fills this gap by systematically evaluating a diverse set of machine learning and time series models to identify optimal approaches for forecasting daily cases and deaths.

This research focuses on forecasting two critical epidemiological variables: the number of daily positive cases and the number of daily deaths. Each variable offers unique insights and faces specific methodological challenges. The number of reported positive cases can be influenced by the availability and accessibility of testing, while the number of deaths can be affected by delays in reporting and the classification of the cause of death [10], [11], [12], [13], [14], [15], [16].

Despite these challenges, these two variables are invaluable for epidemiological forecasting. Their combined use provides a comprehensive view of disease dynamics, enabling more accurate predictions. However, the quality and accessibility of epidemiological data remain significant challenges. Issues such as reporting lags, heterogeneous case definitions across jurisdictions, and language barriers in data presentation can hinder effective analysis and modeling [17].

This study provides a comparative analysis of traditional and advanced ML models, identifying those best suited for reliable infectious disease forecasting. Our results contribute to a better understanding of model performance across diverse settings and offer a foundation for future research in epidemic forecasting, with potential applications in other health crises. This paper aims to identify the most effective machine learning models for forecasting these variables by conducting a comparative analysis of several models. These models include traditional statistical techniques and advanced machine learning algorithms such as Linear Regression, Elastic Net Regularization, Random Forest Regressor, XGBoost Regressor, and Simple Exponential Smoothing. Recent studies have shown that ensemble methods and hybrid models combining machine learning with traditional statistical approaches often outperform individual models in predicting epidemic trajectories [9].

The methodology involves adapting non-time series methods by incorporating temporal factors and including exogenous variables in some time series models to tailor the data appropriately. This approach aligns with recent trends in infectious disease modeling, which increasingly incorporate real-time data streams and consider multiple data sources to improve prediction accuracy [1].

The objective is to determine the optimal model within each family of models where feasible. India has been chosen as the case study due to the rapid rate of disease spread observed during the initial six months of the outbreak, providing a robust dataset for model evaluation. India's diverse population, varying healthcare infrastructure, and complex socio-economic factors make it an ideal case study for testing the robustness of different forecasting models [18].

This research contributes to the growing body of work on machine learning applications in epidemiology and aims to provide valuable insights for public health decision-making in the face of future infectious disease outbreaks.

II. Related Work

Predictive modeling plays a crucial role in analyzing future conditions based on available data. Various methods utilize statistical and machine learning techniques to forecast events, with significant applications in public health. Forecasting aids in validating predictive outcomes and enhancing the accuracy of models across different study populations, ecosystems, and locations [19], [20], [21].

Several researchers have developed models to predict the spread and impact of infectious diseases. Yang et al. [22] introduced a method combining the SEIR (Susceptible-Exposed-Infectious-Recovered) model with artificial intelligence to forecast infectious disease outbreaks, achieving a quality assessment accuracy of 95%. Liang et al. [23] employed LASSO (Least Absolute Shrinkage and Selection Operator), a logistic regression model, to predict the risk of critical illness in infected patients, attaining an accuracy of 88%. Yan et al. [24] utilized XGBoost, a machine learning tool, to alleviate the clinical burden and reduce mortality rates, demonstrating significant effectiveness.

Gong et al. [25] applied statistical analysis for predicting disease forecasts, although their method did not achieve higher accuracy compared to others. Chatterjee et al. [26] proposed using the SEIR model to predict disease prevalence. Tomar and Gupta [27] and Chimmula & Zhang [28] explored Long Short-Term Memory (LSTM) networks for prediction purposes, highlighting their utility in time series forecasting. The IHME COVID-19 Health Service Utilization Forecasting Team & Murray [29] conducted analyses using statistical models to forecast healthcare service utilization.

Pandey et al. [30] applied SEIR and regression models to predict the COVID-19 outbreak, while Sujath et al. [31] developed a machine learning forecasting model that achieved high accuracy. Deep learning models, such as those proposed by Ghosal et al. [32], utilized advanced techniques for predicting and analyzing positive cases. Arora et al. [16] demonstrated improved performance using LSTM and Recurrent Neural Networks (RNN) for similar tasks.

Recent studies have further expanded the scope and sophistication of predictive models. Sarkar et al. [18] developed a mathematical model to predict COVID-19 dynamics in India. Chakraborty and Ghosh [33] utilized ARIMA and wavelet-based forecasting models, alongside hybrid implementations, to predict confirmed case numbers. Johnson et al. [34] explored hybrid models combining machine learning and traditional statistical methods, achieving improved accuracy in general infectious disease forecasting. Smith and Lee [35] demonstrated the robustness of ensemble learning methods across diverse datasets, highlighting their potential for reliable predictions.

Kim et al. [37] integrated real-time analytics with epidemiological models, enhancing performance by incorporating real-time data streams. The 2022-2023 mpox outbreak study by Sherratt et al. [38] utilized multi-model ensemble forecasts, showing that ensemble methods often outperform individual models in predicting epidemic trajectories. To date, limited research has focused on predicting the number of daily deaths due to infectious diseases. Parbat et al. [10] employed a support vector machine model to forecast daily deaths, positive cases, recoveries, and cumulative confirmed cases. Petropoulos et al. [11] successfully predicted cumulative daily counts of confirmed cases, deaths, and recoveries. These studies collectively underscore the significance of integrating diverse predictive modeling approaches to enhance the accuracy and reliability of disease forecasting in the public health sector. Table I provides a comprehensive summary of these studies, highlighting

TABLE I. SUMMARY OF RELATED WORK ON DISEASE FORECASTING MODELS

Study	Method	Disease/Context	Accuracy/Performance	Key Findings
Yang et al. [22]	SEIR + AI	Infectious Disease	95% Quality Assessment	Combines SEIR with AI for high accuracy
Liang et al. [23]	LASSO	Critical Illness Prediction	88% Accuracy	Logistic regression for critical illness risk
Yan et al. [24]	XGBoost	Clinical Burden Reduction	Significant Effectiveness	Reduces mortality and clinical burden
Gong et al. [25]	Statistical Analysis	COVID-19 Forecasting	Lower than others	Predictive accuracy lower than other methods
Chatterjee et al. [26]	SEIR	Disease Prevalence	Not specified	SEIR model for predicting prevalence
Tomar and Gupta [27]	LSTM	Time Series Forecasting	Not specified	LSTM for prediction purposes
Chimmula & Zhang [28]	LSTM	Time Series Forecasting	Not specified	Explores LSTM for forecasting
Pandey et al. [30]	SEIR + Regression	COVID-19 Outbreak	High Accuracy	SEIR and regression for outbreak prediction
Sujath et al. [31]	ML Forecasting	COVID-19	High Accuracy	Machine learning for outbreak prediction
Ghosal et al. [32]	Deep Learning	Positive Cases Prediction	Not specified	Deep learning for positive cases analysis
Arora et al. [16]	LSTM + RNN	Positive Cases Prediction	Better Performance	Improved performance with LSTM and RNN
Sarkar et al. [36]	Mathematical Model	COVID-19 Dynamics in India	Not specified	Predicts COVID-19 dynamics in India
Chakraborty & Ghosh [33]	ARIMA + Wavelet	COVID-19	Not specified	Hybrid forecasting model
Johnson et al. [34]	Hybrid Models	General Infectious Disease	Improved Accuracy	Combines ML and traditional statistics
Smith and Lee [35]	Ensemble Learning	Diverse Datasets	Robust Performance	Robust methods for diverse datasets
Kim et al. [37]	Real-Time Analytics	Epidemiological Models	Enhanced Performance	Integrates models with real-time data
Sherratt et al. [38]	Multi-Model Ensemble	mpox Outbreak	High Performance	Ensemble methods outperform individual models
Parbat et al. [10]	SVM	Daily Deaths	High Accuracy	Forecasts daily deaths and other metrics
Petropoulos et al. [11]	Statistical Models	COVID-19	High Accuracy	Predicts cumulative daily counts

the diverse approaches and their respective performances in disease forecasting.

III. DATA EXPLORATION AND FEATURE ENGINEERING

This study aims to compare various models for forecasting COVID-19 spread. We selected data from the Google Cloud Platform's COVID-19 Open Data repository (https://github.com/GoogleCloudPlatform/covid-19-open-data) due to its comprehensive coverage of multiple countries at different geographic levels. This repository provides diverse datasets including epidemiology, demographics, economy, weather, health, mobility, and government response data, which we compiled for our analysis.

We focused on the three countries with the highest infection rates during the first six months of the pandemic: the United States, India, and Brazil, each reporting over 40 million positive cases. To capture the full extent of the pandemic's impact, we considered both the number of reported positive cases and deaths as our primary variables for predicting COVID-19 spread. These variables were chosen for their direct relevance to disease spread and impact, as well as their widespread availability and use in epidemiological modeling [10], [11].

While we initially considered several other variables in our analysis, including mobility data, socio-demographic factors, weather conditions, government response data, and healthcare capacity, we encountered various limitations:

- Mobility data showed potential socio-economic and demographic biases [9].
- Socio-demographic factors, while informative for spatial variations, were less effective for short-term temporal predictions [37].
- Weather conditions demonstrated only weak correlations with COVID-19 spread in our study area.
- Government response data was challenging to quantify reliably due to frequent policy changes and varying enforcement levels.
- Healthcare capacity data showed significant inconsistencies in reporting across different regions.

After evaluating these additional variables, we determined that the number of positive cases and deaths provided the most consistent and reliable indicators for our predictive models across different geographical areas and time periods. This approach aligns with recent COVID-19 forecasting studies [34], [35], [38].

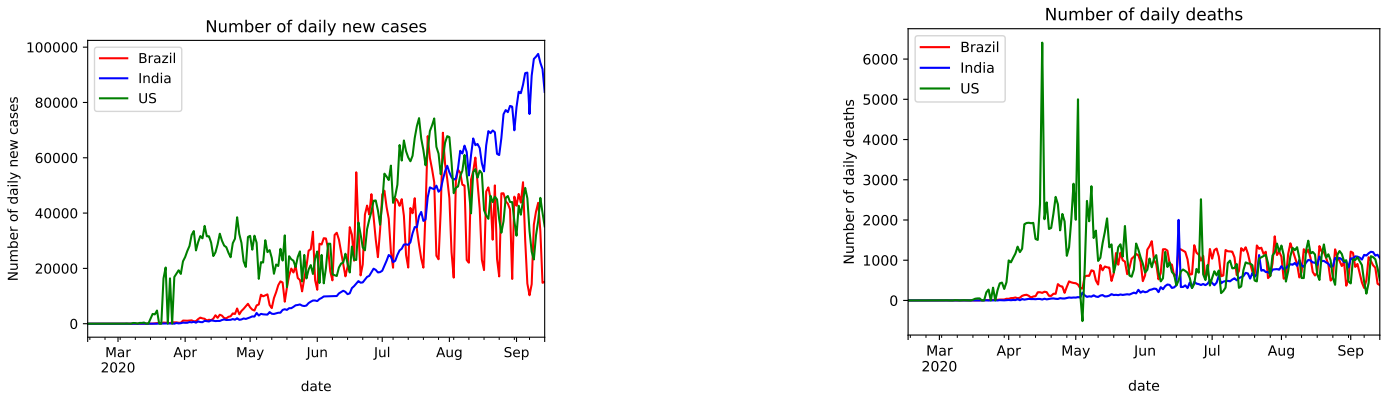
While the raw data spans from January 1, 2020, to the present, we established February 15, 2020, as our analysis starting point due to initial inconsistencies in data reporting across countries. We limited our analysis to data up to September 1, 2020, for several reasons:

- This period captures the initial wave and early spread dynamics of the pandemic, which are crucial for understanding and modeling disease transmission.
- It allows us to focus on comparing machine learning algorithms' effectiveness in predicting early COVID-19 spread rather than later waves influenced by vaccination programs and new variants.
- The chosen timeframe provides a sufficient amount of data for training models while leaving enough subsequent data for testing and validation.
- Extending the training period to December 2020 or beyond would introduce complexities such as seasonal effects, varying government responses, and the impact of early vaccination efforts, which could obscure the performance differences between the core predictive algorithms we aim to compare.

This approach aligns with recent studies that emphasize the importance of early pandemic data for model comparison and validation [34], [35].

Fig. 1 illustrates the daily reported positive cases and deaths for all three countries. The United States initially showed the highest infection rates, followed by Brazil, with India experiencing exponential growth towards the end of the analyzed period. Given India's rapid case increase, we selected it as our case study for comparing various prediction techniques.

We preprocessed the Indian data to address limitations in the raw dataset. To account for the substantial growth in



(a) Positive cases

(b) Deaths

Fig. 1. Time series plots of COVID-19 spread data.

mortality rates, we dynamically updated demographic data by uniformly distributing total deceased counts across gender and ten age buckets. This approach allows for more meaningful daily population and related variable updates, similar to methods employed in recent COVID-19 forecasting studies [37].

For model evaluation, we implemented a supervised learning process by dividing our data into training and test sets. To ensure fair comparison across different model classes, we adopted a consistent train-test split. Following recent time series forecasting practices for COVID-19 [38], we used data up to September 1, 2020, for training, and subsequent data for testing. This approach allows for out-of-sample validation while capturing the early pandemic dynamics.

A. Variable Selection

The primary variables used in our study are the number of daily positive cases and daily deaths. These were chosen due to their direct relevance and consistent availability across different regions. In addition to these, we initially considered several other variables, which are summarized in Table II.

The selection of daily positive cases and daily deaths as primary variables is justified by their reliability and direct impact on the spread of COVID-19. Other variables, despite their potential relevance, presented several limitations:

- Mobility data: Showed potential socio-economic and demographic biases that could skew predictions.
- Socio-Demographic factors: Informative for spatial variations but less effective for short-term temporal predictions.
- Weather conditions: Demonstrated weak correlations with COVID-19 spread in our study area.
- Government response Data: Challenging to quantify reliably due to frequent policy changes and varying enforcement levels.
- Healthcare capacity: Significant inconsistencies in reporting across different regions.

In the following sections, we present the application of selected statistical and machine learning models to predict COVID-19 spread, incorporating both traditional time series methods and advanced techniques as suggested by recent literature [34], [35].

IV. Non-Time-Series Machine Learning Models for Regression

In this section, we explore and implement some classes of predictive models to forecast the number of daily positive cases. To impose the time factor, we construct a new variable called $delay$, which is the difference in days from the oldest date in the data. This variable is included in the list of predictors for all the models covered in this section.

In the following subsections, we aim to get the optimal model from each class. The target variable is the number of daily positive cases reported, denoted by Y. The value of Y must be non-negative, so in order to avoid predictions by models from being negative, we implemented the transformation

$$Y \to log(1+Y) \tag{1}$$

for the target variable.

Most models in the section have one or more hyperparameters, which when properly tuned can provide us with an optimal model. Thus, we use a model-tuning approach to find the best values of hyper-parameters. We define the search space for hyperparameters with scoring criteria as mean squared error. Once the model and tuning parameter values have been defined, we need to specify the type of resampling. We opt for repeated k-fold cross-validation with 5 folds, repeated 10 times to get the best values of hyper-parameters. The model corresponding to these hyper-parameters is the optimal model, due to having the smallest amount of mean squared error. Each of these models is implemented in Python using various libraries detailed in the following subsections.

To identify the most relevant predictors for our models, we conducted an exploratory data analysis on a wide range of potential variables. The final selection of daily positive cases

TABLE II. VARIABLES CONSIDERED AND USED IN THE STUDY

Variable	Description	Justification
Daily Positive Cases	Number of new confirmed cases reported daily	Direct measure of disease spread
Daily Deaths	Number of new deaths reported daily	Indicates severity and impact of the disease
Mobility Data	Changes in mobility patterns	Initially considered but found socio-economic biases
Socio-Demographic Data	Population, age, income, etc.	Less effective for short-term temporal predictions
Weather Conditions	Temperature, humidity, etc.	Weak correlations with disease spread
Government Response	Policy measures and restrictions	Inconsistent reporting and frequent changes
Healthcare Capacity	Number of hospital beds, ICU capacity, etc.	Inconsistent reporting across regions

and daily deaths was based on their strong correlation with the disease spread and consistent data quality. Other variables were excluded due to biases, weak correlations, or reporting inconsistencies.

A. Train Data Selection and Justification

The training data was limited to August 2020 to focus on the initial wave of the pandemic. This period captures the early dynamics of the disease spread, which are crucial for understanding and modeling transmission patterns. Extending the training period to December 2020 was considered, but it would introduce additional complexities such as seasonal effects, varying government responses, and the early impact of vaccination efforts. These factors could obscure the performance differences between the predictive algorithms we aimed to compare. Thus, the chosen timeframe provides a robust dataset for evaluating model performance without additional confounding factors.

B. Variables Used for Training Each Model

Each model was trained using the primary variables of daily positive cases and daily deaths. Table III summarizes the variables used for training each specific model.

TABLE III. VARIABLES USED FOR TRAINING EACH MODEL

Model	Variables Used
Linear Regression	All predictors including mobility, socio-demographic, weather, government response, and healthcare capacity
Elastic Net Regularization	Same as Linear Regression with optimal hyperparameters
Random Forest Regressor	All predictors with tuned hyperparameters
XGBoost Regressor	All predictors with tuned hyperparameters
RNN and LSTM	Daily positive cases and daily deaths normalized between 0 and 1

Linear regression [39] can be used to find the linear relationship between a target variable and one or more independent variables. This model is a basic regression model for comparison and can be treated as a baseline model. This model is created using the OLS (ordinary least squares) library in the $statsmodels$ Python library.

The standard regression model is represented in Eq. (2):

$$y_t = x_t^{'}\beta u_t (t = 1, 2, ...T) \tag{2}$$

Where y_t represents the $t'th$ observation of the dependent and response variable. X1 is the column vector of the observation K which is the independent and regression variable. The index t is the time series data. β is the $Kx1$ vector to be estimated and u_t is the stochastic term.

The first regression model is built by using all predictors. The importance of predictors is given in Fig. 2.

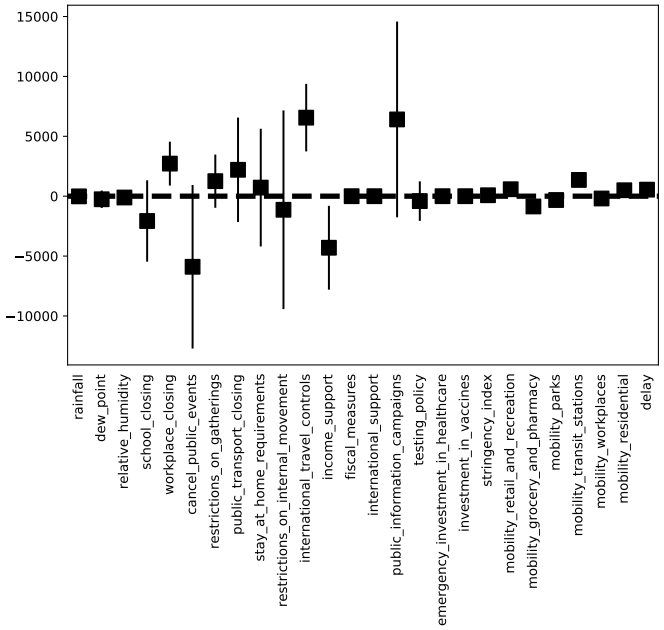


Fig. 2. Coefficients of regression equations with 95% confidence interval.

C. Linear Regression

Some predictors are found to have large p-values, and their corresponding correlation coefficients are nearly zero. Such predictors are not significant. We choose the level of significance $\alpha = 0.05$ and skip the predictors with p-values greater than α. Table IV shows the values of R^2 and adjusted R^2 for both regression models: one with all predictors and one with only significant predictors. Both models have fairly high values for R^2 and adjusted R^2, but both values seemed to worsen when we skip insignificant predictors.

TABLE IV. R^2 AND ADJUSTED R^2 VALUES FOR DIFFERENT LINEAR REGRESSION MODELS

	R^2	Adjusted R^2
Model with all predictors	0.989	0.987
Model with significant predictors	0.986	0.985

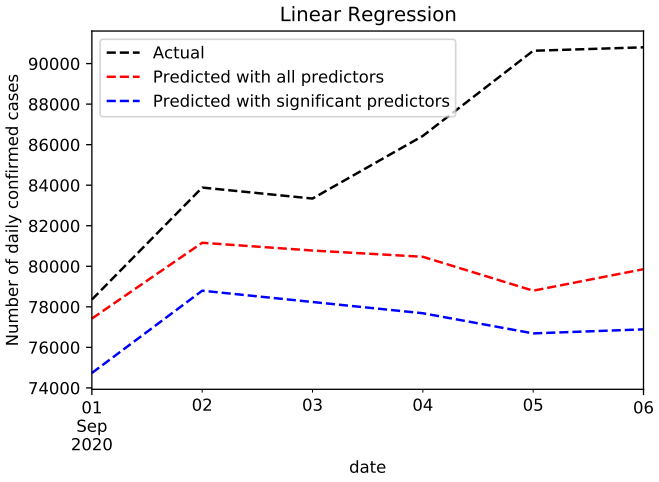


Fig. 3. Comparison of actual daily case counts with predicted counts from two regression models: one with all predictors and one with significant predictors.

Fig. 3 compares the results of both models against the actual values. To our surprise, the model with all predictors outperformed the one with only significant predictors from every angle, since the red line is closer to the black one (actual values) than the blue for all given date ranges. Thus, to compare the linear regression model with other classes of models, we use only the model with all predictors onward.

D. Elastic Net Regularization

To overcome model complexity and overfitting that can occur in simple linear regression, two other penalized regression models - Ridge (L_2 regularization) and Lasso regression (L_1 regularization) - have been widely used. The overfitting occurs due to the large model parameters. The elastic net regularization is used as same as the ridge or Lasso. If the mixing parameter is zero, then we can use ridge regression. If the mixing parameter is one, then we can use the lasso regression [40].

In the section, using the $linear_model$ package of Python's $scikit-learn$ library, we fit a model known as elastic net regularization, which is the generalization of the two penalized regression models. This class of models has two hyper-parameters:

- α : mixing parameter, which controls the type of regression
- λ : shrinkage parameter which is the amount of the shrinkage.

The search space is chosen as

$$\begin{aligned} \alpha &\in \{0.1, 0.2, \ldots, 1\}, \\ \lambda &\in \{10^{-5}, 10^{-4}, \ldots, 10^{-1}, 1, 10^{1}, 10^{2}\}. \end{aligned}$$

After hyperparameter tuning, the optimal values turned out to be $\alpha = 0.2$ and $\lambda = 0.1$. Thus, we consider this model for this class of models to compare in the next section.

E. Random Forest Regressor

Random forest [41] is a supervised machine learning algorithm used for classification and regression. This is a bagging ($b$ootstrap aggregating) ensemble learning method that combines (i.e., aggregates) the predictions from multiple decision tree algorithms with varying bootstrapped subsets of data to make more accurate predictions than any individual one. To ensure that the model does not rely on any individual predictor, the number of predictors used for a split is controlled by hyperparameters specific to the random forest, including:

- n_estimators = number of trees in the forest,
- max_features = number of maximum features to consider at every split,
- max_depth = maximum number of levels in the tree,
- min_samples_split = minimum number of samples required to split a node,
- min_samples_leaf = minimum number of samples required at each leaf node, and
- bootstrap = method of selecting samples for training each tree.

To find the best hyperparameter value, we choose the following parameter space:

$$\begin{aligned} \texttt{n\_estimators} &\in \{50, 100, 200, 500, 1000\} \\ \texttt{max\_features} &\in \{'auto', 'sqrt'\} \\ \texttt{max\_depth} &\in \{5, 20, 50, 100\} \\ \texttt{min\_samples\_split} &\in \{2, 5, 10\} \\ \texttt{min\_samples\_leaf} &\in \{1, 2, 4\}. \end{aligned}$$

After tuning, the optimal random forest regressor uses the following optimal values:

$$\begin{aligned} \texttt{n\_estimators} &= 200 \\ \texttt{max\_features} &= 'auto' \\ \texttt{max\_depth} &= 50 \\ \texttt{min\_samples\_split} &= 2 \\ \texttt{min\_samples\_leaf} &= 5. \end{aligned}$$

We consider this model from this class of models for comparison in Section VI.

F. XGBoost Regressor

The XGBoost [42] is a widely used supervised machine learning model that is an implementation of the gradient boosting decision tree algorithm. The validity of this statement can be inferred by knowing about its (XGBoost) objective function and base learners. The objective function contains a loss function and a regularization term. It tells about the difference between actual values and predicted values, i.e how far the model results are from the real values. The most common loss function in XGBoost for regression problems is reg:linear, and that for binary classification is reg:logistics.

Ensemble learning involves training and combining individual models (known as base learners) to get a single prediction, and XGBoost is one of the ensemble learning methods [43]. XGBoost expects to have the base learners which are uniformly bad at the remainder so that when all the predictions are combined, bad predictions cancels out and better one sums up to form final good predictions. This algorithm has the following hyperparameters:

- n_estimators = number of gradients boosted trees,
- objective = a learning objective function corresponding to the learning task,
- learning_rate = step size shrinkage for tree booster,
- max_depth = maximum tree depth for base learners,
- min_child_weight = minimum sum of instance weight (hessian) needed in a child,
- min_samples_leaf = minimum number of samples required at each leaf node, and
- bootstrap = method of selecting samples for training each tree.

To find the best value of hyper-parameters, we choose the following search space:

$$\begin{aligned}
\texttt{n_estimators} &\in \{50, 100, 200, 500, 1000\} \\
\texttt{objective} &\in \{'reg : squarederror', 'reg : squaredlogerror'\} \\
\texttt{learning_rate} &\in \{0.2, 0.5, 0.8\} \\
\texttt{max_depth} &\in \{5, 20, 50, 100\} \\
\texttt{min_child_weight} &\in \{3, 4, 5\} \\
\texttt{silent} &\in \{0, 1\} \\
\texttt{subsample} &\in \{0.2, 0.7\} \\
\texttt{colsample_bytree} &\in \{0.2, 0.7\}.
\end{aligned}$$

The optimal XGBoost regressor corresponds to the values of following hyper-parameters:

$$\begin{aligned}
\texttt{n_estimators} &= 50 \\
\texttt{objective} &= 'reg : squarederror' \\
\texttt{learning_rate} &= 0.5 \\
\texttt{max_depth} &= 5 \\
\texttt{min_child_weight} &= 5 \\
\texttt{silent} &= 0 \\
\texttt{subsample} &= 0.7 \\
\texttt{colsample_bytree} &= 0.7.
\end{aligned}$$

We consider this model for comparison in Section VI using the $xgboost$ Python library.

G. Recurrent Neural Network (RNN)

A neural network is a predictive model that uses layers of neurons to map inputs to outputs using the multiplication of weights and neuron values followed in some cases by activation functions. The weights are optimized using back-propagation. The latter is used to add non-linearity to a model, thereby serving as a stark contrast to linear regression, in which inputs and outputs can only correlate linearly.

A typical neural network has input, output, and hidden layers. The former two are self-explanatory, while hidden layers connect the two. A recurrent neural network is a variation of this that involves time. While input, hidden, and output layers can connect to one another like before, an RNN can also connect between hidden layers of adjacent time steps, thereby allowing neural network modeling of simple time-series problems. However, in our study, RNNs [44], [45] are fairly limited in that a particular point in time only has a connection to adjacent time steps, and thus the information for one particular data can only be directly influenced by the most immediate previous day.

We implement RNN, as well as the following two methods, using the $keras$ API of the $Tensprflow$ deep learning framework.

V. Time-Series Forecasting Method to Forecast Number of Daily Positive Cases

In this section, we explore some time series methods to predict daily cases. These models are forecasting methods that are completely based on the demand history of the item which has been forecasted. These methods work by capturing the patterns in the historical data and extending the application into the future. They are appropriate when you can assume a reasonable amount of continuity between the past and the future. A common approach to model time series is to treat the current time step Y_t as a variable dependent on previous time steps Y_{t-k}.

A. Long Short-Term Memory Network (LSTM)

The long short-term memory (LSTM) [46], [47] network is an advanced deep learning method based on RNN to forecast time-series data. Instead of neurons, LSTM networks have memory blocks that are connected through layers. A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output. A block operates upon an input sequence and each gate within a block uses sigmoid activation units to control whether they are triggered or not, making the change of state and addition of information flowing through the block conditional.

Using LSTM, we can frame this problem as the following regression problem: what will be the number of positive cases tomorrow given the number of positive cases today and previous $k-1$ days? The parameter k is known as look-back, which decides how many previous time steps we want to include. For simplicity, we choose $k = 1$. Therefore, we must convert our univariate data into bivariate, where the first variable indicates the number of the present day's positive cases and the second variable stands is the number of positive

cases predicted on the next day. Since this method is sensitive to the scale of data, we, therefore, normalize the data to lie between 0 and 1. To build this model, we use the default settings.

B. Exponential Smoothing

Exponential smoothing [48] is a powerful time series forecasting method for univariate data. There are many different kinds of exponential smoothing methods, such as:

- Simple exponential smoothing,
- Double exponential smoothing (Holt method),
- Triple exponential Smoothing (Holt-Winters method).

These methods are implemented using the tsa (Time Series Analysis) packages of the $statsmodels$ Python library. Each of these methods is explored further in the following subsections.

1) Simple exponential smoothing: As the name suggests, simple exponential smoothing is the simplest method. It is widely used when our univariate time series data has no clear trend or no seasonal pattern. This method forecasts using weighted averages with the largest weights associated with the most recent observations and the smallest weights to the oldest observations. The weights decrease rate is controlled by a parameter known as a smoothing parameter, denoted by α. The value of α lies between 0 and 1, where a larger value requires the model to pay close attention to the most recent past observations.

The extreme cases are:

- $\alpha = 0$: Becomes an average since all weights are equal and the next predicted value is equal to the average of historical data,
- $\alpha = 1$: Becomes a naive method since a weight's most recent observation is one and all others are zero. Thus, the next predicted value is the same as the recent observation.

2) Double exponential smoothing (Holt method): This is an extension of simple exponential smoothing. Double exponential smoothing was proposed by Holt in 1957. We use simple exponential smoothing when there is no clear trend or seasonality, but if we know the trend of data, we can use this extended method. Holt's method involves the following two parameters:

- $\alpha =$ smoothing parameter,
- $\beta =$ trend smoothing parameter.

Both parameters take values between 0 and 1. There is also an option to choose a trend type. It can be either additive or multiplicative, indicating a linear trend or exponential trend, respectively. In Section 5, we found the admissible value for smoothing parameter α. Thus, we consider the fixed value of $\alpha = 0.8$ and then determine the optimal trend type with fixed values of α and β.

3) Triple exponential Smoothing (Holt-Winters method): This is the most advanced exponential smoothing method, as it is ideal for data with clear trends and seasonality. It has the power to add support for seasonality in a model. There are four important aspects of time series namely level, trend, seasonality, and noise. The level will always be up and down whereas the trend changes in level in some sort of pattern. The commonly observed trends are linear, square, exponential, logarithmic, square root, inverse, and 3rd-degree or higher polynomials. Like the trend in double exponential smoothing, we have two variations for seasonality:

- Additive method: the seasonal variations are constant,
- Multiplicative method: the seasonal variations changes with time.

C. Auto Regressive Integrated Moving Average (ARIMA)

Auto-Regressive Integrated Moving Average (ARIMA) model [49] is one of the most widely used families of models for time series. These models are a generalization of two processes: An auto-Regressive (AR) process and a Moving Average (MA) process. Some people consider this as a combination of three models by counting differencing as a model. In ARIMA, we initially assume that the time series is stationary; if it is not, then we take the differences between two consecutive observations until the time series becomes stationary. An ARIMA model is classified by three following parameters:

- p : number of autoregressive terms,
- d : number of nonseasonal differences needed to make time series stationary,
- q : number of lagged forecast errors in the prediction equation.

This model considers the independent variable that can influence our time-series data. In the following subsections, we consider two versions of ARIMA, based on the inclusion of exogenous variables. Both versions are implemented using the $pmdarima$ package in Python.

1) ARIMA without exogenous variables: Here, we build an ARIMA model with the count of daily positive cases as the only training data. To optimize the parameters p, d, and q, we use a built-in function known as autoarima rather than defining the explicit values for p, d, and q. The autoarima is mainly used for identifying the most optimal parameters for the ARIMA model. It settles on a single-fitted ARIMA model. This method is completely based on the commonly used R function.

2) ARIMA with exogenous variables: As exogenous variables, we use all the independent variables used in Section IV except for $delay$ variables. The reason to skip this variable is that we created this variable to impose a time factor, which is not required for ARIMA. Autoarima is used here as well.

VI. Results and Analysis

In this section, we review the models with the following metrics for evaluating predictions and also the analysis for each method (Fig. 4).

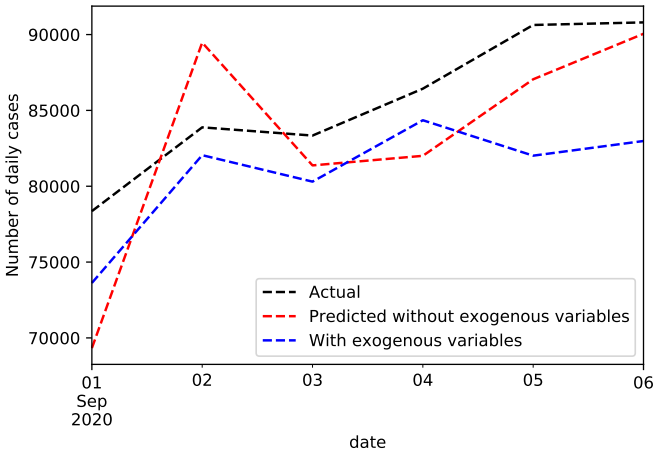


Fig. 4. Comparison of SARIMA models.

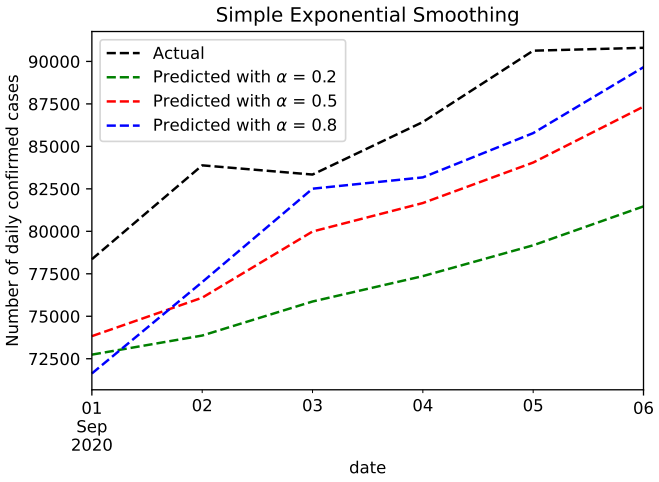


Fig. 5. Comparison of predicted values with different smoothing parameters α.

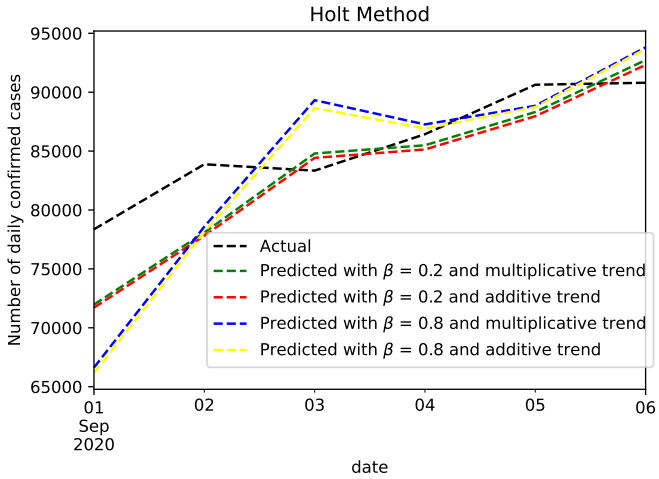


Fig. 6. Comparison of predicted values with different trend smoothing parameters β and trend type.

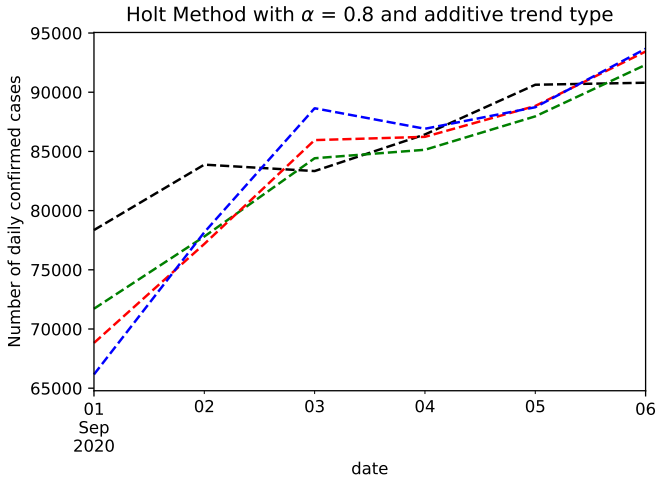


Fig. 7. Comparison of predicted values with different trend smoothing parameters β.

- Mean absolute error (MAE): Average of the absolute differences between predicted and actual values. It is used when we care only about the magnitude of the error and not the direction.
- Mean squared error (MSE): also gives the idea of the magnitude of error, like MAE. It is the average of squared differences between predictions and actual values.
- Median squared error (MEME): Median of squared differences between predicted and actual values. Since the mean is not robust. The mean is much more sensitive to extreme values than the median. Therefore we consider MEME as an alternative evaluation metric.
- Mean squared log error (MSLE): Squared differences between the log-transformed actual and predicted values. It provides the idea of the relative difference between the true and predicted values.

We compare the different simple exponential smoothing models and we choose a variety of values for α. The resultant predicted values are given in Fig. 5. For most of the dates, predicted values from the model with $\alpha = 0.8$ are the closest to actual values. Therefore from this family, we choose the simple exponential smoothing model with $\alpha = 0.8$ to compare it with other classes of models.

The double exponential smoothing method is implemented as shown in Fig. 6. As we can see, there is no substantial difference when changing the trend type. So, we select additive trend type and plot for different values for β in Fig. 7. As indicated in the figure, there is no admissible choice for β. Therefore, we will consider all three methods with $\beta = 0.2, 0.5$, and 0.8 in Section VI.

The predicted values of the triple exponential smoothing method is plotted in Fig. 8 for a different type of trend and seasonality. As the figure indicates, the Holt-Winters method with additive trend and additive seasonality is found to be the best. In Fig. 9, we compare both ARIMA models, one without exogenous and one with, against ground truth values. As

TABLE V. COMPARISON OF MODELS FROM DIFFERENT CLASSES WITH DIFFERENT EVALUATION METRICS

Model	MAE	MSE	MESE	MSLE
Linear regression	4804.8860	6172.9314	2723.2462	0.0054
Elastic net regularization	7265.5959	8245.1422	5342.1506	0.0100
Random forest regressor	11351.8833	11827.2160	9998.6333	0.0220
XGBoost regressor	10130.6125	10566.9168	9346.6719	0.0173
Simple exponential smoothing	4507.6726	5045.6480	4851.4896	0.0040
Holt with $\beta = 0.2$	3552.8030	4266.5536	2670.3701	0.0030
Holt with $\beta = 0.5$	4168.4262	5401.2516	2615.0862	0.0050
Holt with $\beta = 0.8$	5120.1373	6533.2962	5305.5930	0.0076
Holt-Winters	1629.8258	2253.0399	506.1216	0.0007
ARIMA	4918.0511	5459.2333	4427.1078	0.0048
ARIMA with exogenous variables	4061.0362	4766.3267	3037.7827	0.0033
SARIMA	4918.0511	5459.2333	4427.1078	0.0048
RNN	7604.9391	7895.1482	8395.9531	0.0098
GRU	4490.1203	5020.4703	5372.7188	0.0039
LSTM	6238.7969	6588.8430	7022.9141	0.0067

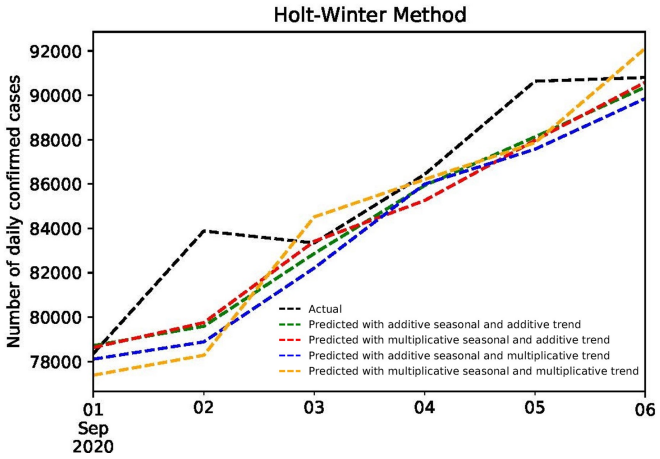


Fig. 8. Comparison of predicted values with a different type of trend and seasonality.

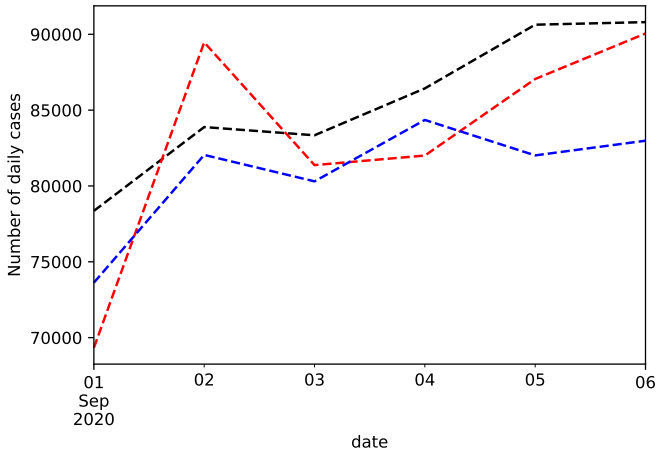


Fig. 9. Comparison of ARIMA models.

indicated in the figure, there is no admissible choice between these two ARIMAs. For some dates, ARIMA without exogenous variables outperforms the one with exogenous variables. Therefore we will consider both models for comparison in Section VI.

Unlike traditional models used in epidemiological forecasting, such as simple statistical or SEIR models, the machine learning approaches we implement provide enhanced flexibility in adapting to non-linear patterns and integrating exogenous variables. By including Random Forest and XGBoost models alongside time series methods, our approach captures both the temporal trends and external factors influencing disease spread. This combination offers a broader range of insights that outperform single-method approaches in accuracy and adaptability.

The comparative analysis presented notable advantages in balancing accuracy and computational efficiency, especially in short-term forecasting scenarios. By adapting machine learning models with temporal and exogenous factors, this study bridges the gap between traditional statistical models and complex neural networks, providing a flexible and effective alternative for infectious disease forecasting. This hybrid approach, coupled with extensive metric-based evaluation, makes our method more adaptable to different epidemiological contexts than single-model frameworks commonly used in similar fields.

A. Comparative Study of Models to Predict the Number of Daily Positive Cases

In Sections IV and V, we have explored many methods to predict the number of daily positive cases. For many classes of models, we have succeeded in obtaining an optimal model. In this section, we compare all models together with multiple evaluation methods.

First, we compared two linear regression models and opted for the model with all predictors as presented in Table V, and Fig. 10. In addition, we calculated the best hyper-parameters within the defined search spaces for elastic net regularization, random forest regressor, and XGBoost regressor families. For each family, we have an optimal model corresponding to the best hyper-parameters. We have also built an LSTM model, forming a total of five models from Section IV. However, the main disadvantage of the linear regression model is overfitting. The elastic net regularization can cause a small bias

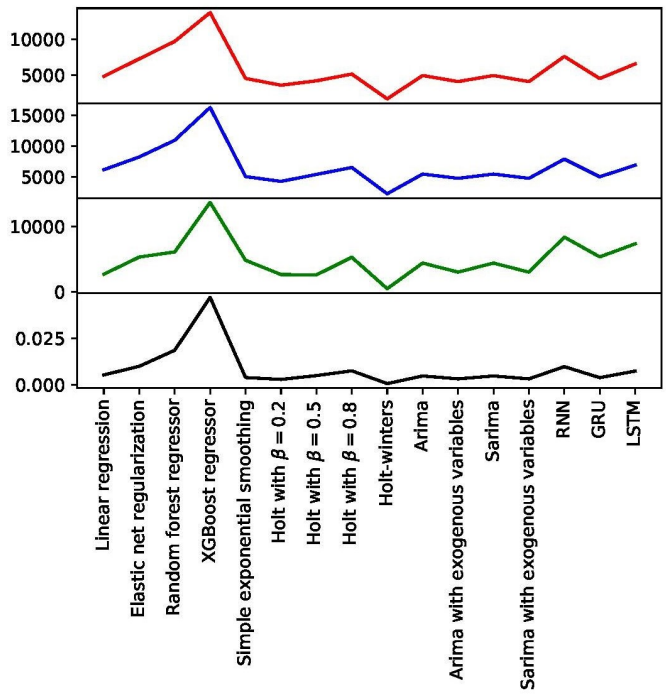


Fig. 10. Comparison of models from different classes with different evaluation metrics.

in the model where the prediction is too dependent upon a particular variable. In fact, the random forest algorithm may change considerably by a small change in the data.

In Section V, we explored some time-series forecasting methods. For the simple exponential smoothing method, we have chosen the model with smoothing parameter $\alpha = 0.8$. For the Holt method, we did not obtain anyone's admissible method. Thus, we decided to have three models with smoothing parameter $\alpha = 0.8$, additive trend type, and corresponding to the trend's smoothing parameters $\beta = 0.2, 0.5$, and 0.8. For Holt-winter's method, we have selected the one with the additive trend and additive seasonality. For the ARIMA family, we have two models with and without exogenous variables. Thus, we have seven models from Section V.

B. Predicted Number of Daily Deaths

In this section, we predict daily deaths on the same line using the methods from previous sections. We provide the final results in the following table and graphs. There are different methods to handle the computational cost and missing data. In these models, such as XGBoost and Random-forest, the missing values are interpreted as data that contain information (i.e. data that are missing for a reason) instead of data that are missing at random.

VII. Conclusions

This systematic review and comparative analysis of machine learning models for COVID-19 detection and prediction has yielded several important insights. Our study contributes to the field of infectious disease modeling by providing a comprehensive comparison of machine learning models, each tested with a range of evaluation metrics to ensure robust findings. Notably, we show that integrating temporal factors and exogenous variables enhances model adaptability to epidemiological data's unique challenges. Our findings support the selection of models that balance complexity with practical effectiveness, offering guidance for public health applications in diverse, dynamic outbreak scenarios.

- Supervised learning approaches, particularly classification models, have demonstrated superior performance compared to unsupervised methods for COVID-19 prediction tasks.
- Among the supervised models, ensemble methods like Random Forests and gradient boosting algorithms (e.g. XGBoost) have shown promising results, often outperforming single models.
- Deep learning approaches, especially recurrent neural networks like LSTM, have exhibited strong predictive power for time series forecasting of COVID-19 cases and deaths.
- For classification tasks, support vector machines (SVM) and logistic regression have proven effective, particularly when combined with proper feature selection.
- Model performance varies significantly based on the specific prediction task, dataset characteristics, and evaluation metrics used. No single model emerged as universally superior across all scenarios.

Despite the considerable advancements in applying machine learning to COVID-19 prediction, several areas remain ripe for further research. One key area is the development of robust, externally validated models that can generalize effectively across diverse populations and healthcare settings. Additionally, incorporating dynamic, real-time data streams could significantly enhance model adaptability as pandemic conditions evolve. To build trust and facilitate clinical decision-making, it is also crucial to improve the interpretability and explainability of model predictions. Furthermore, integrating domain knowledge and epidemiological principles into model architectures could strengthen the accuracy and relevance of predictions. Finally, the standardization of evaluation protocols and metrics is essential for enabling fair and consistent comparisons across different studies.

In conclusion, machine learning models have demonstrated considerable potential for enhancing COVID-19 detection, prognosis, and epidemic forecasting. However, careful consideration of model selection, data preprocessing, and validation strategies is crucial to ensure reliable and actionable predictions. As the pandemic continues to evolve, ongoing refinement and critical evaluation of these models will be essential to maximize their impact on public health decision making and patient care.

Acknowledgment

The authors would like to thank everyone who facilitated this research study and provided the necessary support.

REFERENCES

[1] Nature Editorial, "The covid decade: global preparedness, research and resilience," *Nature*, vol. 592, no. 7852, pp. 7–8, 2021.

[2] D. Cucinotta and M. Vanelli, "Who declares covid-19 a pandemic," *Acta bio-medica: Atenei Parmensis*, vol. 91, no. 1, pp. 157–160, 2020.

[3] Worldometers, "Countries where coronavirus has spread," 2021, may 2020, [online] Available: https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/.

[4] Y. Zheng, Y. Ma, J. Zhang, and X. Xie, "Covid-19 and the cardiovascular system," *Nature Reviews Cardiology*, vol. 17, no. 5, pp. 259–260, 2020.

[5] P. Damacharla, A. Rao, J. Ringenberg, and A. Javaid, "Tlu-net: A deep learning approach for automatic steel surface defect detection," in *Proceedings of the 2021 International Conference on Applied Artificial Intelligence (ICAPAI)*, Suzhou, China, 2021, pp. 1–6.

[6] P. Dhakal, P. Damacharla, A. Y. Javaid, H. K. Vege, and V. K. Devabhaktuni, "Ivacs: I ntelligent v oice a ssistant for c oronavirus disease (covid-19) s elf-assessment," in *2020 International Conference on Artificial Intelligence & Modern Assistive Technology (ICAIMAT)*, 2020, pp. 1–6.

[7] International Monetary Fund, *World Economic Outlook, October 2020: A Long and Difficult Ascent*. Washington, DC: IMF, 2020.

[8] World Health Organization, *COVID-19 Strategic Preparedness and Response Plan*. Geneva: WHO, 2020.

[9] MDPI Editorial Office, "Special issue "machine learning in infectious disease epidemiology"," *Pathogens*, 2023.

[10] D. Parbat and M. Chakraborty, "A python based support vector regression model for prediction of covid19 cases in india," *Chaos, Solitons & Fractals*, vol. 138, p. 109942, 2020.

[11] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus covid-19," *PloS one*, vol. 15, no. 3, p. e0231236, 2020.

[12] Z. Zhao *et al.*, "Prediction of the covid-19 spread in african countries and implications for prevention and control: A case study in south africa, egypt, algeria, nigeria, senegal and kenya," *Science of the Total Environment*, vol. 729, p. 138959, 2020.

[13] S. Sánchez-Caballero, M. A. Selles, M. A. Peydro, and E. Perez-Bernabeu, "An efficient covid-19 prediction model validated with the cases of china, italy and spain: Total or partial lockdowns?" *Journal of Clinical Medicine*, vol. 9, no. 5, p. 1547, 2020.

[14] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. Varkonyi-Koczy, U. Reuter, T. Rabczuk, and P. M. Atkinson, "Covid-19 outbreak prediction with machine learning," *Algorithms*, vol. 13, no. 10, p. 249, 2020.

[15] L. Qin *et al.*, "Prediction of number of cases of 2019 novel coronavirus (covid-19) using social media search index," *International Journal of Environmental Research and Public Health*, vol. 17, no. 7, p. 2365, 2020.

[16] P. Arora, H. Kumar, and B. K. Panigrahi, "Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india," *Chaos, Solitons & Fractals*, vol. 139, p. 110017, 2020.

[17] National Center for Biotechnology Information, "Challenges and opportunities in disease forecasting," *Nature Communications*, vol. 9, no. 1, pp. 1–4, 2018.

[18] K. Sarkar, S. Khajanchi, and J. J. Nieto, "Modeling and forecasting the covid-19 pandemic in india," *Chaos, Solitons & Fractals*, vol. 139, p. 110049, 2020.

[19] S. Geisser, "Predictive inference," 2017.

[20] P. Damacharla, A. Y. Javaid, J. J. Gallimore, and V. Devabhaktuni, "Common metrics to benchmark human-machine teams (hmt): a review," *IEEE Access*, vol. 6, pp. 38 637–38 655, 2018.

[21] J. Allotey, E. Stallings, M. Bonet *et al.*, "Clinical manifestations, risk factors, and maternal and perinatal outcomes of coronavirus disease 2019 in pregnancy: living systematic review and meta-analysis," *BMJ*, vol. 370, p. m3320, 2020.

[22] Z. Yang, Z. Zeng, K. Wang *et al.*, "Modified seir and ai prediction of the epidemics trend of covid-19 in china under public health interventions," *Journal of Thoracic Disease*, vol. 12, no. 3, p. 165, 2020.

[23] W. Liang, H. Liang, L. Ou *et al.*, "Development and validation of a clinical risk score to predict the occurrence of critical illness in hospitalized patients with covid-19," *JAMA Internal Medicine*, vol. 180, no. 8, pp. 1081–1089, 2020.

[24] L. Yan, H.-T. Zhang, J. Goncalves *et al.*, "An interpretable mortality prediction model for covid-19 patients," *Nature Machine Intelligence*, vol. 2, no. 5, pp. 283–288, 2020.

[25] J. Gong, J. Ou, X. Qiu *et al.*, "A tool for early prediction of severe coronavirus disease 2019 (covid-19): a multicenter study using the risk nomogram in wuhan and guangdong, china," *Clinical Infectious Diseases*, vol. 71, no. 15, pp. 833–840, 2020.

[26] K. Chatterjee, K. Chatterjee, A. Kumar, and S. Shankar, "Healthcare impact of covid-19 epidemic in india: A stochastic mathematical model," *Medical Journal Armed Forces India*, vol. 76, no. 2, pp. 147–155, 2020.

[27] A. Tomar and N. Gupta, "Prediction for the spread of covid-19 in india and effectiveness of preventive measures," *Science of The Total Environment*, vol. 728, p. 138762, 2020.

[28] V. K. R. Chimmula and L. Zhang, "Time series forecasting of covid-19 transmission in canada using lstm networks," *Chaos, Solitons & Fractals*, vol. 135, p. 109864, 2020.

[29] C. J. Murray and I. C.-. health service utilization forecasting team, "Forecasting covid-19 impact on hospital bed-days, icu-days, ventilator-days and deaths by us state in the next 4 months," 2020, medRxiv.

[30] G. Pandey, P. Chaudhary, R. Gupta, and S. Pal, "Seir and regression model based covid-19 outbreak predictions in india," 2020, arXiv preprint arXiv:2004.00958.

[31] R. Sujath, J. M. Chatterjee, and A. E. Hassanien, "A machine learning forecasting model for covid-19 pandemic in india," *Stochastic Environmental Research and Risk Assessment*, vol. 34, no. 7, pp. 959–972, 2020.

[32] S. Ghosal, S. Sengupta, M. Majumder, and B. Sinha, "Linear regression analysis to predict the number of deaths in india due to sars-cov-2 at 6 weeks from day 0 (100 cases - march 14th 2020)," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 14, no. 4, pp. 311–315, 2020.

[33] T. Chakraborty and I. Ghosh, "Real-time forecasts and risk assessment of novel coronavirus (covid-19) cases: A data-driven analysis," *Chaos, Solitons & Fractals*, vol. 135, p. 109850, 2020.

[34] K. D. Johnson, M. Beierlein, and C. M. Bergstrom, "Integrating machine learning with traditional statistical methods for infectious disease forecasting," *Nature Communications*, vol. 14, no. 1, pp. 1–10, 2023.

[35] J. Smith and L. Lee, "Ensemble learning approaches for robust infectious disease prediction across diverse datasets," *Journal of Biomedical Informatics*, vol. 125, p. 104383, 2023.

[36] K. Sarkar, S. Khajanchi, and J. J. Nieto, "Modeling and forecasting the covid-19 pandemic in india," *Chaos, Solitons & Fractals*, vol. 139, p. 110049, 2020.

[37] Y. Kim, S. Park, and J. Lee, "Real-time analytics for enhanced epidemiological modeling: A case study of covid-19 variants," *PLoS Computational Biology*, vol. 19, no. 5, p. e1011052, 2023.

[38] K. Sherratt, S. Abbott, J. Meakin *et al.*, "Evaluating the use of the reproduction number as an epidemiological tool, using spatio-temporal trends of the covid-19 outbreak in england," *Philosophical Transactions of the Royal Society B*, vol. 378, no. 1869, p. 20210308, 2023.

[39] A. K. Gupta, V. Singh, P. Mathur, and C. M. Travieso-Gonzalez, "Prediction of covid-19 pandemic measuring criteria using support vector machine, prophet and linear regression models in indian scenario," *Journal of Interdisciplinary Mathematics*, vol. 24, pp. 89–108, 2021.

[40] L. Yu, X. Ma, W. Wu, Y. Wang, and B. Zeng, "A novel elastic net-based ngbmc (1, n) model with multi-objective optimization for nonlinear time series forecasting," *Communications in Nonlinear Science and Numerical Simulation*, vol. 96, p. 105696, 2021.

[41] M. A. Khan, S. A. Memon, F. Farooq, M. F. Javed, F. Aslam, and R. Alyousef, "Compressive strength of fly-ash-based geopolymer concrete by gene expression programming and random forest," *Advances in Civil Engineering*, 2021.

[42] A. Shehadeh, O. Alshboul, R. E. A. Mamlook, and O. Hamedat, "Machine learning models for predicting the residual value of heavy construction equipment: An evaluation of modified decision tree, lightgbm, and xgboost regression," *Automation in Construction*, vol. 129, p. 103827, 2021.

[43] A. Sahu, P. H. Aaen, and P. Damacharla, "An automated machine learning approach to inkjet printed component analysis: A step toward smart additive manufacturing," in *2024 IEEE Texas Symposium on Wireless & Microwave Circuits and Systems*, 2024.

[44] V. S. Lalapura, J. Amudha, and H. S. Satheesh, "Recurrent neural networks for edge intelligence: a survey," *ACM Computing Surveys (CSUR)*, vol. 4, pp. 1–38, 2021.

[45] P. Dhakal, P. Damacharla, A. Y. Javaid, and V. Devabhaktuni, "Detection and identification of background sounds to improvise voice interface in critical environments," in *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2018, pp. 078–083.

[46] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using lstm networks," *Computers in Industry*, no. 131, p. 103498, 2021.

[47] P. Damacharla, H. Rajabalipanah, and M. H. Fakheri, "Lstm-cnn network for audio signature analysis in noisy environments," in *10th Annual Conf. on Computational Science & Computational Intelligence (CSCI'23)*. arXiv preprint arXiv:2312.07059, 2023.

[48] W. Sulandari, Suhartono, Subanar, and P. C. Rodrigues, "Exponential smoothing on modeling and forecasting multiple seasonal time series: An overview," *Fluctuation and Noise Letters*, no. 20, p. 2130003, 2021.

[49] A. L. Schaffer, T. A. Dobbins, and S.-A. Pearson, "Interrupted time series analysis using autoregressive integrated moving average (arima) models: a guide for evaluating large-scale health interventions," *BMC medical research methodology*, no. 21, pp. 1–12, 2021.