Research Article

BIOMEDICAL Journal of Scientific & Technical Research

ISSN: 2574 -1241

DOI: 10.26717/BJSTR.2022.45.007184

A Study on the Development of a Machine Learning Prediction Model on the Spread of COVID-19

Juhwan Moon[1], Hongsik Yoon[2], Hongsul Lee[1] and Jaejoon Lee[1]*

[1]*Interdisciplinary Program in Crisis, Disaster and Risk Management, Sungkyunkwan University, South Korea*

[2]*School of Civil, Architectural Engineering & Landscape Architecture, Sungkyunkwan University, South Korea*

***Corresponding author:** Jaejoon Lee, Interdisciplinary Program in Crisis, Disaster and Risk Management, Sungkyunkwan University, Seoul, South Korea

ARTICLE INFO

Received: June 22, 2022

Published: July 26, 2022

Citation: Juhwan Moon, Hongsik Yoon, Hongsul Lee, Jaejoon Lee. A Study on the Development of a Machine Learning Prediction Model on the Spread of COVID-19. Biomed J Sci & Tech Res 45(2)-2022. BJSTR. MS.ID.007184.

Keywords: Corona Pandemic; Mechine-Learning; Prediction Model

ABSTRACT

The corona pandemic has caused many human lives and economic losses. The number of confirmed cases, which slowed since February of this year during the availability of vaccines, increased rapidly due to mutated viruses and casual enforcement of the quarantine system. The difference in this study, as compared to those done previously, is that the most recent data was used, and sufficient learning data is used for training. In addition, the number of confirmed cases was predicted based on the latest information including those who received the primary vaccine and those who were fully vaccinated. In addition, we used a predictive model with information only from confirmed corona cases, subdivided it by parameter, and tried to propose an accurate and effective predictive model for the number of corona-confirmed cases. In this study, the machine-learning model used neural networks, ensembles, distance-based models, and linear regression as supervised learning models.

As for the model with excellent predictive power, Gradient Boosting and AdaBoosting had high training scores, and CatBoost showed the best predictive power among the Gradient Boosting models through cross-validation by model. About 94.8% of the predictions were accurate. CatBoost's predictive power was poor in the area where the number of confirmed cases rapidly increased due to the mutated virus. In particular, it was confirmed that the CatBoost model was effective in predicting small and irregular infections in the early stage, but that the prediction of the period of a rapid increase in the number of confirmed cases due to delta mutation was somewhat ineffective. As a future research task, it is necessary to implement and compare prediction algorithms using machine learning techniques trained in unsupervised learning. In addition, it is necessary to make a prediction using the policy variables to be considered, such as the stage and implementation of the social distancing movement.

Introduction

The coronavirus, first discovered in China in December 2019, caused a global pandemic via person-to-person transmission. In response to this pandemic, each country has established various health care policies. The need for national risk management is increasing as the coronavirus outbreak is affecting the efficacy of the health care system due to the unprecedented increase in patients, but also the survival and permanence of businesses. As of August 4, 2021, there have been more than 200 million confirmed cases and more than 4 million deaths worldwide. In addition, despite the primary inoculation of more than 40% of the coronavirus vaccine, the rate of spread according to the mutant is increasing. In this case, the importance of vaccines and therapeutics is on the rise, and the

only way to prevent the spread of Corona is strict personal hygiene management and intensive social distancing, and the number of confirmed cases continues to occur even though it is being practiced steadily.

Therefore, health authorities must decide on workforce planning and policy responses within a short period of time. Hence, accurately predicting the spread of the coronavirus that will occur in the near future at a sufficiently granular level will help the authorities to provide better information and more time to respond accurately. Effective policies based on that information will be of great help, not only to prevent the spread of infectious diseases but also to secure corporate continuity. The problem of this study is to develop a model that can accurately predict the spread of the disease using machine techniques. As in previous studies, external factors (seasonal, environmental, geographical) were not used. In addition, in previous studies, realistic data could not be constructed because the number of daily primary vaccine recipients and the number of completed daily vaccinations were not utilized. The structure of this paper is as follows: Chapter 2 reviews related prior research and predictive models based on machine learning. Chapter 3 compares training scores for each model, goes through model validation, selects an optimal predictive result model, and selects an optimal predictive model by changing detailed parameters. Finally, it includes conclusions and future plans.

Related Research

Prior Research

With the coronavirus spreading rapidly around the world, numerous studies have been conducted to develop a model that predicts the spread of the coronavirus. Looking at previous studies, Marvel (2020) and Metha (2020) evaluated the risk of infection in the United States through an ensemble of existing epidemiological models and machine learning techniques, respectively, as prior studies based on corona data. And Ceylan, et al. [1] proposed an ARIMA model to predict corona cases in Italy, Spain, and France, and as a result, found that the MAPE ranged from 4 to 6%. Chimmula and Zhang [2] used a deep learning model (LSTM) to predict the end of the coronavirus in Canada. conducted a study to predict the trend and end time of corona confirmed in Canada with a cyclic neural network model. Yang, et al. [3] used a model for corona prediction by combining population movement data and epidemiologic data in China. They reported that combining susceptible-exposed-infected-removed (SEIR) and LSTM models were effective in predicting the peak and magnitude of infectious diseases. He, et al. [4] simulated the spread of corona in Hubei, China through the SEIR disease spread model and particle swarm optimization algorithm. Alazab et al. showed that the Prophet model is effective in predicting the number of confirmed cases, recoveries, and deaths from coronavirus in Australia and Jordan.

Arora, et al. [5] conducted a study to predict the trend of corona confirmed in India with a circular neural network model. Pandey, Gaurav, et al. [6] conducted an experiment to predict the number of corona cases in India using an SEIR model and a regression model. Data from Johns Hopkins University [7] was used to predict the number of confirmed cases over a two-week period. RMSLE was used as the prediction result. The SEIR model achieved 1.52 and the regression model achieved 1.75, indicating satisfactory performance. Alzahrani, et al. [8] used the ARIMA model to predict the number of COVID-19 cases in Saudi Arabia. The research team used the combination of ARIMA to determine the best model fit and proposed ARIMA (2,1,1) as the most suitable predictive model for the daily number of confirmed cases in Saudi Arabia. Pinter, Gergo, et al. [9] proposed a hybrid machine learning approach to predict COVID-19. Using Hungarian data, the researchers proposed a model that combines an adaptive network-based fuzzy inference system (ANFIS) with a multi-layer perceptron competition algorithm (MLP-ICA). Looking at previous studies in Korea, Jeong (2020) used a mathematical epidemiologic model to estimate the domestic infection weight and evaluated the effectiveness of government policies, and Kim Jin-oh, et al. [10] 19 confirmed cases and deaths by country in the world in comparison and analysis of predicted cases. The SIR (Susceptible-Infected-Recovered) model is used for prediction through the epidemic model, and the curve fitting of the SIR model was performed with L-BFGS-B among the machine learning optimization algorithms.

In addition, Jin-soo Bae and Seong-beom Kim [11] proposed a methodology for predicting new confirmed cases four days later by using the information on confirmed cases of corona so far and considered legal holidays in predicting new cases of corona with a machine learning model. Myung-hui Kim (2021) proposed a deep learning-based model that combines CNN, Bi-LSTM, and Attention mechanisms to predict the number of confirmed cases for COVID-19 in a corona-confirmed patient prediction model using a deep learning-based prediction model. In addition, Hyeongju Seon (2021) predicted the daily number of confirmed cases by synthesizing external variables such as epidemiological data, demographic data, and search trends. As a result of experimenting with various models, it was proved that tree and regression-based machine learning models can predict the number of confirmed patients significantly. In addition, Seung-Yeol Lee and Myung-Ki Shin (2020) studied how to predict and control the number of confirmed cases coming from abroad in predicting the number of confirmed cases of COVID-19 using mathematical modeling. They proposed a mathematical model that can predict the number of overseas inflows, and the proposed model predicts the number of overseas inflows using roaming service data and the LSTM algorithm. There are also previous studies that investigated the relationship between climate, aviation, and web data, and corona.

First, as a study using climate and temperature, Mohammad, et al. Corona was judged as a seasonal respiratory virus and investigated how factors such as altitude, humidity, and temperature might apply. Peng Shi, et al. [12] investigated the relationship between corona and temperature based on weather and epidemiological data as an environmental factor in the outbreak of COVID-19 in China investigated the relationship between the transmission of coronavirus and ecological factors of tropical climate in Brazil and found that temperature had a negative linear relationship with the number of confirmed cases. As a study using the impact of aviation, To, et al. [13], analyzed the relationship between the number of passengers at Hong Kong International Airport and COVID-19. Coelho et al. tested the effects of climate and aviation for the prediction of COVID-19. Kumar, et al. [14] explained that India is a country in which mobility between countries is very diverse and infection cases vary dynamically from region to region. For accurate prediction, they studied population migration data and monthly data of airline passengers. Researchers at KAIST in Korea [15] have designed a Hi-COVID Net for monitoring COVID-19. The proposed model solved the problem of monitoring inbound travelers in each country and predicting cases of COVID-19 coming from abroad. It also showed practicality and effectiveness through real-world experiments and predicted the number of imported COVID-19 cases in the future much more accurately than the baseline.

Finally, as a study using web data, Qin, Lei, et al. [16] proposed a model for predicting the number of cases through the Social Media Search Index (SMSI) for COVID-19. Kia Jahanbin, et al. [17] proposed the FAMEC system to collect unstructured data from COVID-19 on Twitter to monitor the spread of the epidemic. Li, Civilian, et al. [18] predicted epidemic outbreaks in China using Internet searches and social media data. Based on the fact that web data occurs earlier than the spread of COVID-19, they developed a model to monitor a new epidemic using Google Trends and Baidu Index and found that the data had a high correlation with real COVID-19 data [19].

Predictive Model

Looking at the prediction methodology for predicting corona confirmed, a number of prior studies made predictions by using the hydrodynamic methodology first. Recently, mathematical mechanics and machine learning have been applied to epidemiological research. Mathematical mechanics has been used as an effective tool for monitoring and predicting the prevalence of infectious diseases. Still, machine learning is being used because of problems that cannot be solved due to too many variables and computational amounts. In epidemiological information, population information of three groups (S (Sensitive) group, I (Infected) group, and R (Recovered) group) is used. The three groups are called SIR using acronyms. Group S is the sum of the population at risk of infection because they do not have immunity to the disease and the population of the infected who do not know that they are infected yet. Group I is the population that recognized and confirmed the infection. Group R refers to the population in a state in which the disease is cured or dies and will not become ill or become a spreader. The second is a machine learning-based prediction model, which is a Long Short-Term Memory (LSTM) model applied as well as a neural network model. LSTM is a structure in which a hidden layer unit is added as an LSTM cell in the existing RNN structure. The LSTM cell determines the data output by weighting the value of each cell when the distance between the input data and the output data of the previous step increases. A decision tree is a model that makes predictions using several decision rules of a hierarchical structure. Random forest, which is one of the ensemble models using bagging, is a method of calculating a final prediction value by learning multiple decision trees and then synthesizing (voting, averaging, multiplying, etc.) the result values of each decision tree. A model combining several decision trees has better prediction performance than a single decision tree.

The advantage of integrating the random forest into multiple decision tree models is that even if some decision trees make incorrect predictions, accurate prediction is possible by synthesizing the prediction results of multiple decision tree models. Gradient boosting is the same as random forest in that it predicts by synthesizing the prediction results of several decision trees, but the most prominent feature is that it trains decision trees sequentially. The next decision tree learns the error that the previous decision tree incorrectly predicted, and the subsequent decision tree gradually reduces the error. The model structure and searched hyperparameters must be adjusted. XGBoost, an improved model of Gradient Boosting, is one of the ensemble models using boosting. Because it is based on CART (Classification and Regression Tree), it can be used for both classification and regression problems like the random forest. Since it uses the same methodology as Gradient Boosting, the weights among several internal models are determined by gradient descent. The K-nearest neighbor method is generally used for classification problems, but it can solve regression problems with the same principle as classification. In the same way as the KNN classification model, the distance between each variable in the N-dimensional space is calculated. (In this case, N is the number of independent variables used.) In the subsequent classification problem, the categories of the K nearest neighbors are taken by the voting method, but in the regression problem, the average of the values of the K nearest neighbors is taken.

In this case, fine adjustment is possible through the formula for calculating the distance and the weighted average. Tree-based

models used in this study include Tree model, Random Forest, and Gradient Boosting models, Extreme Gradient Boosting (XGBoost), Gradient Boosting (Sickit-Learn), Extreme Gradient Boosting Random Forest (XGBoost), and Gradient Boosting (CatBoost). AdaBoost was also used. Linear Regression was used as the regression-based model. Ridge Regression (L2), Lasso Regression (L1), and Elastic Net Regression were used. The distance-based model was used for each metric option as a KNN regressor. Prediction after learning was attempted using SGD, SVM, and Neural Network. However, their result predictions were either too unstable or under-fitted, so they failed to learn. Therefore, KNN, Tree, Neural Network, Random Forest, Gradient Boosting, Linear Regression, and AdaBoost were used in this study. The hardware and software used were ANACONDA.NAVIGATOR's Orange 3 3.26.0, RStudio, and Excel. The limitation of previous studies is that the existing prediction models generated a prediction model with only daily data of a simple corona confirmed patient and divided it into sections to improve the prediction power. The overall predictive power can be high, but the realism of the forecast is insufficient. In addition, since infectious diseases such as corona have non-linearities, there are limits to accurate predictions assuming a simple data set is used Previous studies used a univariate regression model using only the variables of confirmed coronavirus cases.

They did not reflect changes over time, such as vaccinated persons (the number of primary vaccinated persons including those released from isolation and the number of persons who completed vaccination). In addition, it was not possible to secure adequate data for sections where the number of confirmed cases changed rapidly and irregular sections, so the reliability to evaluate the prediction performance was insufficient. As a past study, recent trends could not be utilized. In this study, realistic data were constructed using the number of daily primary vaccinations and the number of completed daily vaccinations. External factors (seasonal, environmental, geographical factors) used in the previous studies, mainly corona analysis in 2020, are gradually not being used. Propagation and diffusion due to external factors should be identified and used as parameters, but this study does not consider external factors. In many previous studies, temperature and regional characteristics were used by utilizing the information that seasonal factors, climate, environment, and geographical factors are related to the spread of corona. However, it was excluded in this study because it is not a highly correlated part due to 2 years of group learning of the corona pandemic. Instead, the number of daily deaths, the number of daily testers, and the infection rate compared to the testers were reinforced to increase the predictive power. In addition, the data set was composed of a balanced data set because uncertainty in the data could lead to the failure of machine learning model learning. Unlike previous studies, various models (Gradient Boosting, XGBoost, CatBoost, etc.) were used to increase the prediction accuracy.

Data Collection and Preprocessing

The original data used in this paper was obtained from the Johns Hopkins University public data set (https://github.com/CSSEGISandData/COVID-19) and coronaboard.kr. The data were from March 04, 2020, up to August 04, 2021, (Table 1) shows the contents of target variables and feature variables with the collected data.

Table 1: Variables and Configurations.

Variable	Code	Explanation
Date	Date	Date
Target	K_D_Cnfrm	Number of daily confirmed cases in Korea
Feature	K_D_Dth	Daily death toll in Korea
	K_D_Rcvr	Number of daily recoveries in Korea (those who completed vaccination)
	K_F_Vac	Number of daily primary vaccinations in Korea
	K_D_Exm	Number of daily inspections in Korea
	C_Cnfrm	Cumulative confirmation rate
	Under_Care	Number of patients in treatment per day
Meta	G_d_Cnfrm	Global daily number of confirmed cases
	G_D_Rcvr	Global daily recoveries

In this study, Input Total Data is 519 instances, Training Data is 416 instances, while Test Data is 103 instances, and they were divided 8 to 2. For cross-validation data for each model, 10-folds or 20-folds of training data were used. KNN, Tree, Random Forest, Gradient Boosting, Linear Regression, AdaBoost, SVM, Neural Network, etc., are used as models. (Table 2) shows the descriptive statistics of the variables used in this study. The x-axis is Date (start=2020-03-04) and corresponds to the distribution of KD_Cnfrm, KD_Dth, KD_Rcvr, KF_Vac, KD_Exm, C_Cnfrm, Under_Care, Gd_Cnfrm, and GD_Rcvr data in column units. (Figures 1 & 2) shows the correlation of variables visualized by heatmap. The correlation between the number of daily tests (K_D_Exm) and the number of daily confirmed cases (K_D_Cnfrm) was 0.87, which was high. The correlation between the number of patients during daily treatment (Under_Care) and the number of examiners per day (K_D_Exm) was also high at 0.92. In addition, the correlation between the number of daily confirmed cases (K_D_Cnfrm) and the number of patients during treatment (Under_Care) was high at 0.87.

Table 2: Descriptive statistics.

	K_D_ Cnfrm	G_d_ Cnfrm	K_D_ Dth	K_D_ Rcvr	K_F_ Vac	G_D_ Rcvr	K_D_ Exm	C_ Cnfrm	Under_ Care
Average	386.697	385633.886	4.015	14609.304	39190.131	252121.071	72051.367	1.434	6047.023
Standard Deviation	386.933	231637.037	5.223	40825.079	124291.695	328010.438	68617.795	0.553	5030.412
minimum	2	2310	0	7	0	-5989997	8009	0.902205	623
maximum	1896	1498044	40	288379	894835	1504943	398300	4.919986	22697
number of observations	519	519	519	519	519	519	519	519	519

Note: The data used in this study is visualized as a date.

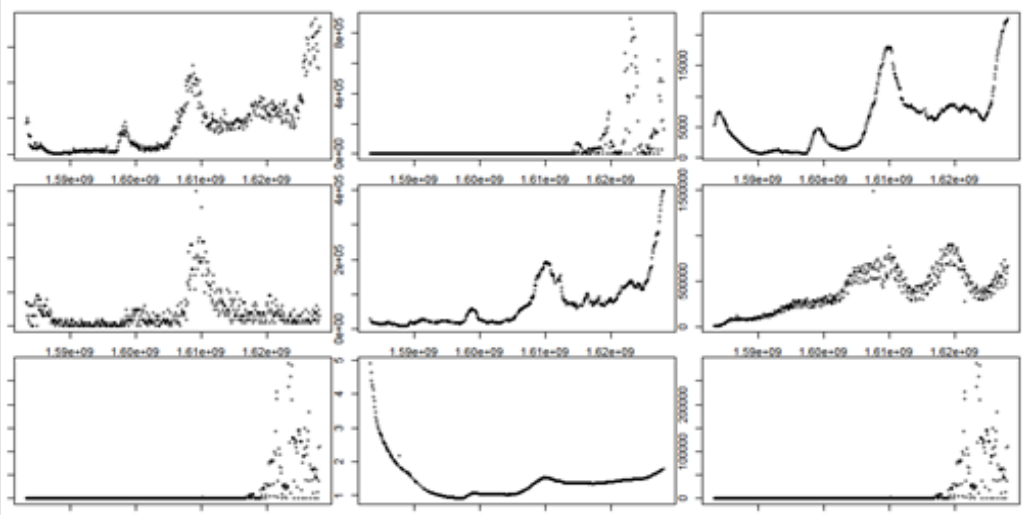

Figure 1: Scatterplot by Date.

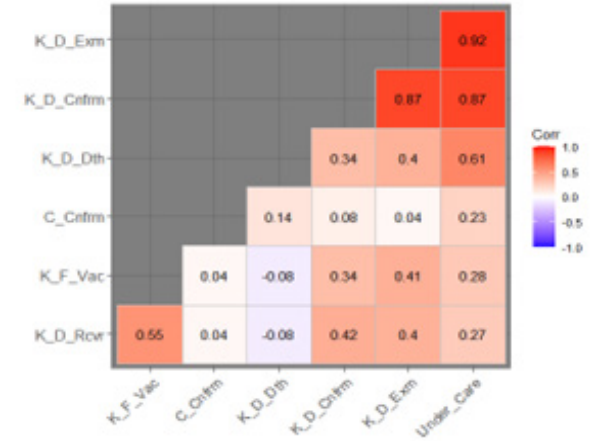

Figure 2: Correlation.

Prediction Results

Model selection

In this study, the machine learning model was trained using 80% of the training data in the set format.

Test on Training Data: The machine learning models used for training include KNN, Tree, SVM, SGD, Random Forest, Neural Network, Linear Regression, Gradient Boosting, and AdaBoost. The machine learning model results using the training data are shown in (Table 3) below. In the table above, SVM (Support Vector Machine), SGD (Stochastic Gradient Descent), and Neural Network were not trained. Therefore, it did not fit the model of this study. First, in the case of SGD, the values of MSE, RMSE, MAE, and R_Squared, which are indices representing the explanatory degree of the model, do not converge to 2.290E+62, 1.513E+31, 1.513E+31, and -1.598E+57. Therefore, it cannot be used in this study. Also, it can be seen that the values of MSE, RMSE, MAE, and R_Squared of the Neural Network do not converge to 150853.729, 388.399, 291.663, and -0.053. This model cannot be used in this study. In SGD, Hinges were selected for Loss Functions, ε was set to 1.10, Regression was set to Squared Loss, and ε was also set to 0.10. In Regularization, Strength(α) was 0.00001, and Ridge(L2) was relatively suitable, but the learning effect did not converge. In the SVM (Support Vector Machine) model, SVM and v-SVM were used. In the former, cost(c) was set to 1.00, and Regression Loss Epsilon(ε) was set to 0.10. The

regression cost(c) of v-SVM was set to 1.00, and the complexity bound(v) was set to 1.00. Also, SVM's Kernel had better RBF than Linear, Polynomial, and Sigmoid. The kernel in v-SVM uses Linear. This was better explained than RBF, Sigmoid, and Polynomial. Numeric Tolerance of Optimization Parameters was set to 0.0010. However, this model also showed no tendency to converge. In the case of Neural Network (NN), the number of Neuron in Hidden Layers was 1000, and among the activation functions, ReLu was relatively superior to tanh, Identity, and Logistic. Among the solvers, Adam and SGD did not converge with the ReLu function, and L-BFGS-B provided the most appropriate value. Regularization, α = 0.002, Maximal Number of Iterations was set to 200, and replicable training was set. However, the overall model fit was very poor. These three models were not used because they did not fit the model of this study.

Table 3: Performance of Test on the training Data.

Sampling type: No sampling, test on training data				
Scores				
Model	MSE	RMSE	MAE	R2
kNN	4566.716	67.577	42.012	0.968
Tree	776.722	27.870	16.890	0.995
SVM	146353.529	382.562	290.764	-0.021
SGD	2.290E+62	1.513E+31	1.513E+31	-1.598E+57
Random Forest	1629.676	40.369	25.107	0.989
Neural Network	150853.729	388.399	291.663	-0.053
Linear Regression	21071.120	145.159	98.025	0.853
Gradient Boosting	2.815	1.678	1.175	1.000
AdaBoost	43.450	6.592	2.954	1.000

Results by Model

In the tree model, the values were obtained by dividing them by parameters first. For the Induces Binary Tree and Min, the number of Instances in Leaves is set to 2, and the Do not Split Subsets smaller than is set to 5. Limit the Maximal Tree Depth to was set to 1000 and calculated by giving other options. In Classification, Stop when Majority Reaches [%] was set to 90. (Table 4) shows that Training Score was 0.995 to 0.996, and Predictions dropped from 0.929 to 0.930. In the KNN model, the Number of Neighbors is set to 3. (Table 5) shows that the Training Score was 0.962 to 0.981, and the learning was good. Predictions showed reasonable predictive power from 0.875 to 0.947. Looking at the settings and results of Linear Regression, Fit Intercept was set as the parameter, while regularization is the same as in (Table 6), and the resulting values were obtained. (Table 6) shows the Training Score and Predictions of Linear Regression, and the Training Score is 0.853. It also shows under-fitting. The predictive power was 0.851, which was also under-fitting in the predictive power, and the predictive power was relatively low.

Table 4: Tree's Training Score and Predictions.

Training Score	MSE	RMSE	MAE	R2
A1	776.722	27.870	16.890	0.995
A2	503.949	22.449	13.660	0.996
Predictions	MSE	RMSE	MAE	R2
A1	12243.895	110.652	68.031	0.929
A2	12052.538	109.784	66.152	0.930

Table 5: Training Score and Predictions of KNN.

Training Score		MSE	RMSE	MAE	R2
Metric	Euclidean	2907.959	53.925	32.930	0.980
	Manhattan	2756.596	52.503	32.538	0.981
	Chebyshev	3009.527	54.859	32.236	0.979
	Mahalanobis	5515.704	74.268	45.646	0.962
Weight	Uniform	2907.959	53.925	32.930	0.980
	Distance	0.000	0.000	0.000	0.000

Predictions		MSE	RMSE	MAE	R2
Metric	Euclidean	9366.723	96.782	58.233	0.946
	Manhattan	9288.819	96.067	58.152	0.947
	Chebyshev	9544.562	97.696	58.210	0.945
	Mahalanobis	21505.904	146.649	88.657	0.875

Table 6: Training Scores and Predictions.

Training Score	MSE	RMSE	MAE	R2
No Regularization (alpha 0.0001)	21071.2	145.159	98.025	0.853
Ridge Regression(L2) (alpha 0.0001)	21071.2	145.159	98.025	0.853
Lasso Regression(L1) (alpha 0.0001)	21071.2	145.159	98.025	0.853
Elastic Net Regression (L1:L2=50:50)	21071.2	145.159	98.025	0.853
Predictions	MSE	RMSE	MAE	R2
No Regularization (alpha 0.0001)	25691.091	160.284	104.691	0.851
Ridge Regression(L2) (alpha 0.0001)	25691.091	160.284	104.691	0.851
Lasso Regression(L1) (alpha 0.0001)	25691.091	160.284	104.691	0.851
Elastic Net Regression (L1:L2=50:50)	25691.091	160.284	104.691	0.851

As a basic property of Random Forest, the Number of Trees is set to 10, as shown in the table below. The training was divided by parameters. By default, the Random Forest condition and Number of Attributes Considered are each Split were set to 5, and the Balance Class Distribution option was chosen, but it was not used because it was a factor that lowered the value. Using the Replicable Training function, the results were different each time. In the Growth Control function, the Limit Depth of Individual Trees was set to 3, while the Do not Split Subsets Smaller than was set to 5. The resulting values are shown in (Table 7). Training Scores and Predictions of Random Forest show that training scores ranged from 0.921 to 0.989, indicating overfitting. The predictive power is 0.871, 0.943, which is the best fitting in the predictive power, and the predictive power is high. 3.3.

Table 7: Training Score and Predictions of Random Forest.

Number of Trees 10				
Training Score	**MSE**	**RMSE**	**MAE**	**R2**
R1	1109.509	33.309	19.960	0.992
R2	1629.676	40.369	25.107	0.989
R3	976.288	31.246	18.629	0.993
R4	1123.190	33.514	19.196	0.992
R5	908.839	53.925	32.930	0.980
R6	1603.255	40.041	25.318	0.989
R7	11283.312	106.223	78.002	0.921
Predictions	**MSE**	**RMSE**	**MAE**	**R2**
R1	9857.872	99.287	61.183	0.943

R2	10271.630	101.349	61.514	0.940
R6	12798.497	113.130	66.342	0.926
R7	22241.533	149.136	99.762	0.871

Comparison of Model Performance

The training score for the training data shows that the model learning rate and AdaBoost show 1.000 over-fitting for MSE 62.952, RMSE 7.934, and MAE 2.514 R2. Gradient Boosting MSE 2.815 RMSE 1.678 MAE 1.175 R2 1.000 is also showing over-fitting. In terms of predictive score, AdaBoost is less accurate with MSE 14414.078. The explanatory power of R2 also fell to 0.916. On the other hand, Gradient Boosting is also high at MSE 9441.362, but R2 is relatively high at 0.945 (Table 8). In the case of Stratified Shuffle Split with 20 Random Samples for 80% Training Data, AdaBoost's MSE is 7253.901, and R2 is 0.943. The learning error was the smallest with MSE 6930.701 of Gradient Boosting. The model explanatory power R2 was also the best at 0.946. Gradient Boosting was the best in terms of predictive power, with an MSE of 9441.362 and an R2 of 0.945 (Table 9). Comparing the Training Score and Prediction Score according to the cross-validation condition, the training score of 10-fold cross-validation is MSE 6530.401, RMSE 80.811, MAE 51.575, and R2 0.954, as in the above case, for AdaBoost. Gradient Boosting is MSE 5819.615, R2 0.959. The training score for 20-fold cross-validation was AdaBoost MSE R2 0.952, and Gradient Boosting MSE 6222.771, R2 0.957.

Table 8: Training score for the training data.

Sampling type: No sampling, test on training data					Predictions Scores Test on Training Data				
Model	**MSE**	**RMSE**	**MAE**	**R2**	**Model**	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	21071.120	145.159	98.025	0.853	Linear Regression	25691.212	160.285	104.691	0.851
Random Forest	11283.312	106.223	78.002	0.921	Random Forest	22241.533	149.136	99.762	0.871
KNN	5515.704	74.268	45.646	0.962	kNN	21505.904	146.649	88.657	0.875
Tree	776.722	27.870	16.890	0.995	Tree	12243.895	110.652	68.031	0.929
AdaBoost	62.952	7.934	2.514	1.000	AdaBoost	14414.078	120.059	66.777	0.916
Gradient Boosting	2.815	1.678	1.175	1.000	Gradient Boosting	9441.362	97.167	56.252	0.945

Table 9: Stratified Shuffle Split.

Sampling type: Stratified Shuffle split with 20 random samples with 80% data					Predictions Scores Random Sampling				
Model	**MSE**	**RMSE**	**MAE**	**R2**	**Model**	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	22833.655	151.108	101.026	0.821	Linear Regression	25691.212	160.285	104.691	0.851
Random Forest	14104.888	118.764	83.391	0.889	Random Forest	22241.533	149.136	99.762	0.871
kNN	12556.134	112.054	68.986	0.901	kNN	21505.904	146.649	88.657	0.875
Tree	9375.781	96.829	62.412	0.926	Tree	12243.895	110.652	68.031	0.929
AdaBoost	7253.901	85.170	54.628	0.943	AdaBoost	14414.078	120.059	66.777	0.916
Gradient Boosting	6930.701	83.251	52.825	0.946	Gradient Boosting	9441.362	97.167	56.252	0.945

The training score of 10-fold cross-validation was higher. However, the training score of 10-fold cross-validation and the training score of 20-fold cross-validation were the same, and in prediction score, MSE 8977.716 RMSE 94.751 MAE 60.367 of gradient boosting, and explanatory power R2 was the best with 0.948 (Table 10). (Figure 3) is a graph visualized by the actual K_D_Cnfrm data remaining at random and the predicted model. The upper left graph shows K_D_Cnfrm - kNN; The lower left graph shows K_D_Cnfrm - Tree, and the upper right graph shows K_D_Cnfrm - Random Forest. The lower right graph shows K_D_Cnfrm-Linear Regression.

Table 10: Comparing the Training Score and Prediction Score.

Sampling type: 10-fold Cross-Validation					Predictions Scores 10-fold Cross Validation				
Model	**MSE**	**RMSE**	**MAE**	**R2**	**Model**	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	22587.375	150.291	100.634	0.842	Linear Regression	25691.212	160.285	104.691	0.851
Random Forest	12934.578	113.730	81.425	0.910	Random Forest	22241.533	149.136	99.762	0.871
kNN	12043.001	109.741	68.295	0.916	kNN	21505.904	146.649	88.657	0.875

Tree	9294.824	96.410	61.643	0.935	Tree	12243.895	110.652	68.031	0.929
AdaBoost	6530.401	80.811	51.575	0.954	AdaBoost	14414.078	120.059	66.777	0.916
Gradient Boosting	5819.615	76.286	49.184	0.959	Gradient Boosting	8977.716	94.751	60.367	0.948
Sampling type: 20-fold Cross Validation					**Predictions Scores 20-fold Cross Validation**				
Model	**MSE**	**RMSE**	**MAE**	**R2**	**Model**	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	22640.705	150.468	100.838	0.842	Linear Regression	25691.212	160.285	104.691	0.851
Random Forest	13896.291	117.883	83.460	0.903	Random Forest	22241.533	149.136	99.762	0.871
kNN	12418.700	111.439	67.874	0.913	kNN	21505.904	146.649	88.657	0.875
Tree	9346.386	96.677	60.840	0.935	Tree	12243.895	110.652	68.031	0.929
AdaBoost	6881.454	82.955	51.276	0.952	AdaBoost	14414.078	120.059	66.777	0.916
Gradient Boosting	6222.771	78.885	48.431	0.957	Gradient Boosting	8977.716	94.751	60.367	0.948

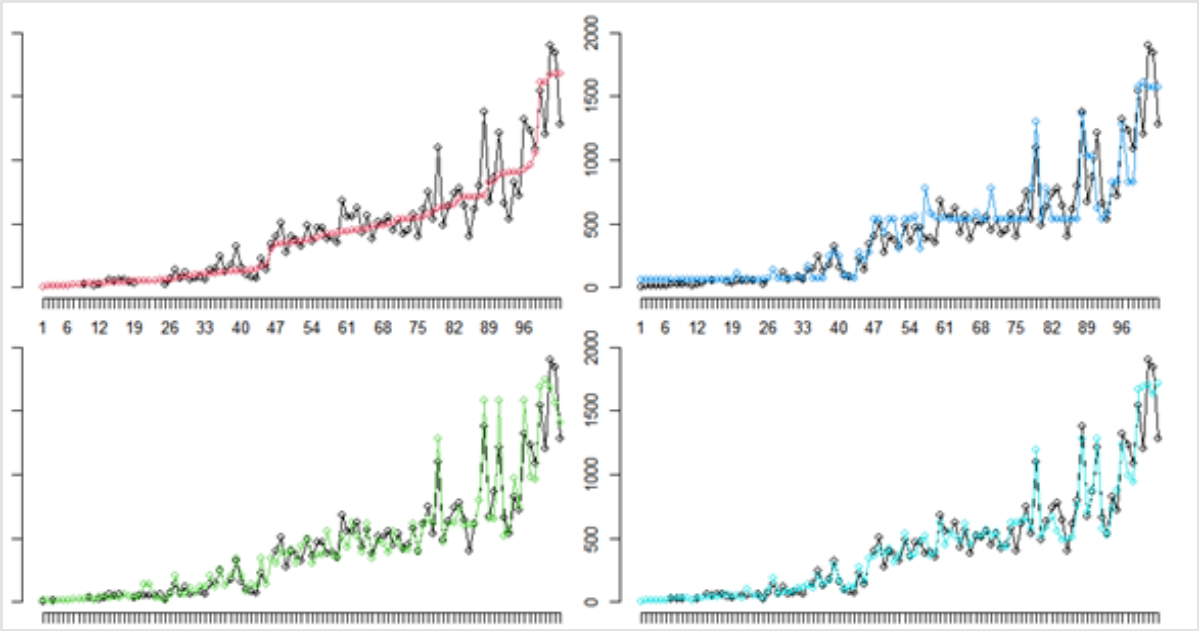

Figure 3: Actual Values and Predicted Values by Model.

Optimal Model Selection

Even in the results of random sampling, gradient boosting has an excellent model fit. R_Squared was 0.946, which was superior to AdaBoost. (Table 11) corresponds to the results of Random Sampling. In Predictions, Gradient Boosting was the best with 0.945. In cross-validation, 10-fold Cross-Validation and 20-fold Cross-Validation showed a higher model explanatory degree (Table 12). AdaBoost and Gradient Boosting were excellent in comparison by the model compared with CVRMSE and MSE. (Table 8) corresponds to Model comparison by CVRMSE and MSE. According to cross-validation, when comparing the models with MSE, it is 0.685:0.315, which shows that the gradient boosting is excellent, and in CVRMSE, it is also 0.749: 0.251, which shows that the gradient boosting is also excellent (Table 13). In the AdaBoost model, the base estimator is set to Tree as a parameter. The number of Estimators is set to 50.

As good results can be derived from a small number, it is sensitively responding. Learning Rate is 1.00000, and the Fixed Seed for Random Generator is not set. SAMME. R and SAMME were used for the Classification Algorithm as a Boosting Method. Also, the regression loss function was calculated using Linear, Square, and Exponential. The obtained values are shown in the table below (Table 14). The table above compares the performance of the SAMME and SAMME.R algorithms. According to Zhu et al. (2009), SAMME.R uses probability estimates to update additive models, whereas SAMME is characterized only for classification. Because the number of iterations is the same, the values are the same. In addition, the SAMME.R algorithm generally converges faster than SAMME, achieving lower test errors with fewer boosting iterations. (Table 15) shows AdaBoost's Training Score and Predictions, and the Training Score is overfitting from 0.997 to 1.000. The predictive power decreased slightly from 0.916 to 0.917.

Table 11: Random Sampling Result.

Sampling type: Stratified Shuffle split, 20 random samples with 80% data				
Training Score				
Model	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	22833.655	151.108	101.026	0.821
kNN	14104.888	118.764	83.391	0.889
Random Forest	12556.134	112.054	68.986	0.901
Tree	9375.781	96.829	62.412	0.926
AdaBoost	7253.901	85.170	54.628	0.943
Gradient Boosting	6930.701	83.251	52.825	0.946
Predictions				
Model	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	25691.212	160.285	104.691	0.851
kNN	21505.904	146.649	88.657	0.875
Random Forest	22241.533	149.136	99.762	0.871
Tree	12243.895	110.652	68.031	0.929
AdaBoost	14414.078	120.059	66.777	0.916
Gradient Boosting	9441.362	97.167	56.252	0.945

Table 12: 10-fold Cross-Validation.

Sampling type: 10-fold Cross-validation				
Training Scores				
Model	**MSE**	**RMSE**	**MAE**	**R2**
Linear Regression	22587.38	150.291	100.6344	0.84238
Random Forest	12934.58	113.7303	81.42456	0.909739
kNN	12043	109.7406	68.29487	0.915961
Tree	9294.824	96.40967	61.64263	0.935139
AdaBoost	6530.401	80.8109	51.57452	0.954429
Gradient Boosting	5819.615	76.2864	49.18379	0.959389

Table 13: Model comparison by CVRMSE and MSE.

Model comparison by MSE						
	Linear Regression	**Random Forest**	**kNN**	**Tree**	**AdaBoost**	**Gradient Boosting**
Linear Regression		0.999	0.999	1.000	1.000	1.000
Random Forest	0.001		0.710	0.988	1.000	1.000
kNN	0.001	0.290		0.967	0.999	0.991
Tree	0.000	0.012	0.033		0.968	0.910
AdaBoost	0.000	0.000	0.001	0.032		0.685
Gradient Boosting	0.000	0.000	0.009	0.009	0.315	
Model comparison by CVRMSE						
	Linear Regression	**Random Forest**	**kNN**	**Tree**	**AdaBoost**	**Gradient Boosting**
Linear Regression		0.999	0.999	0.999	1.000	1.000
Random Forest	0.001		0.784	0.980	1.000	0.999
kNN	0.001	0.216		0.968	1.000	0.994
Tree	0.001	0.020	0.032		0.981	0.924
AdaBoost	0.000	0.000	0.000	0.019		0.749
Gradient Boosting	0.000	0.001	0.006	0.076	0.251	

Table 14: The obtained values.

Training Score	MSE	RMSE	MAE	R2
No Regularization (alpha 0.0001)	21071.2	145.159	98.025	0.853
Ridge Regression(L2) (alpha 0.0001)	21071.2	145.159	98.025	0.853
Lasso Regression(L1) (alpha 0.0001)	21071.2	145.159	98.025	0.853
Elastic Net Regression (L1:L2=50:50)	21071.2	145.159	98.025	0.853
Predictions	MSE	RMSE	MAE	R2
No Regularization (alpha 0.0001)	25691.091	160.284	104.691	0.851
Ridge Regression(L2) (alpha 0.0001)	25691.091	160.284	104.691	0.851
Lasso Regression(L1) (alpha 0.0001)	25691.091	160.284	104.691	0.851
Elastic Net Regression (L1:L2=50:50)	25691.091	160.284	104.691	0.851

Table 15: Performance of Test on the training Data.

Training Score		MSE	RMSE	MAE	R2
SAMME	Linear	43.450	6.592	2.954	1.000
	Square	444.151	21.075	11.305	0.997
	Exponential	21.216	4.606	1.909	1.000
SAMME. R		MSE	RMSE	MAE	R2
	Linear	43.450	6.592	2.954	1.000
	Square	444.151	21.075	11.305	0.997
	Exponential	21.216	1.606	1.909	1.000
Predictions		**MSE**	**RMSE**	**MAE**	**R2**
SAMME	Linear	14414.078	120.059	66.777	0.916
	Square	14463.233	120.263	70.592	0.916
	Exponential	14249.961	119.373	72.155	0.917
SAMME. R	Linear	14414.078	120.059	66.777	0.916
	Square	14463.233	120.263	70.592	0.916
	Exponential	14249.961	119.373	72.155	0.917

Gradient Boosting methods include Extreme Gradient Boosting (XGBoost), Gradient Boosting (Sickit-Learn), Extreme Gradient Boosting Random Forest (XGBoost), and Gradient Boosting (CatBoost). Estimated values for each of these methods are shown in the table below. For Basic Properties, the Number of Trees is 100, and the learning rate is 0.300. Replicable allowed. For regularization, Lambda was set to 1 or 3. The Limit Depth of Individual Trees for Growth Control is 6. For SubSampling, Fraction of Training Instances, Features for each Tree, Features for each Level, and Features for each Split were set to 1.00. (Table 16) shows Training Score and Predictions. In Training Score, Extreme Gradient Boosting (XGBoost) is 1.000, Gradient Boosting (CatBoost) is 0.995, while Gradient Boosting (Sickit-Learn) is 0.990. The above AdaBoost was also 1.000, and the performance was excellent. However, there is a risk of overfitting. An appropriate model is selected by looking at the values in Predictions (Figure 4).

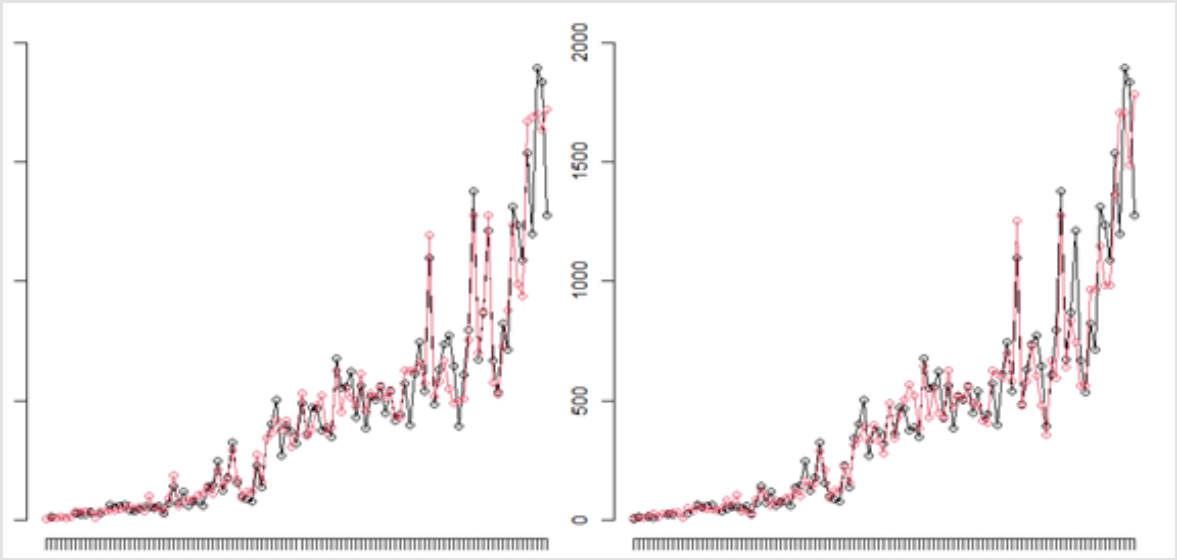

Figure 4: Gradient Boosting (CatBoost) and Predictions of AdaBoost.

Table 16: Training Score and Predictions of Gradient Boosting.

Training Score	MSE	RMSE	MAE	R2
Extreme Gradient Boosting(XGBoost)	2.815	1.678	1.175	1.000
Gradient Boosting(Sickit-Learn) Lambda : 1	1476.145	38.421	27.120	0.990
Extreme Gradient Boosting Random Forest(XGBoost) Lambda : 1	147574.533	384.154	267.973	0.030
Gradient Boosting(CatBoost) Lambda :3	750.561	27.396	20.610	0.995
Predictions	MSE	RMSE	MAE	R2
Extreme Gradient Boosting(XGBoost)	9441.362	97.167	56.252	0.945
Gradient Boosting(Sickit-Learn) Lambda : 1	11636.327	107.872	65.377	0.933
Extreme Gradient Boosting Random Forest(XGBoost) Lambda : 1	180297.184	424.614	99.762	-0.045
Gradient Boosting(CatBoost) Lambda :3	8977.716	94.751	60.367	0.948

Extreme Gradient Boosting (XGBoost) is 0.945, Gradient Boosting (CatBoost) is 0.948, Gradient Boosting (Sickit-Learn) is 0.933. The above AdaBoost is also 0.916~0.917. In this study, Gradient Boosting (CatBoost), which has the best predictive power, was relatively good at 0.948, so it was estimated using this model. The graph on the left compares the actual test data with the predicted value of Gradient Boosting (CatBoost), the graph on the right compares the predicted value of AdaBoost with the actual tester, and it shows that the predicted value of Gradient Boosting (CatBoost) is more accurate. According to CatBoost's prediction results, the predictive power was poor in the area where the number of confirmed cases due to the mutated virus rapidly increased. In particular, it was confirmed that the CatBoost model was effective in predicting small and irregular infections in the early stage, but it was confirmed that the prediction of the period of the rapid increase in the number of confirmed cases due to delta mutation was somewhat ineffective.

Conclusion

Sufficient learning data was used for training using data corresponding to the most recent period. The number of confirmed cases was predicted based on the latest information, including

those who received primary vaccination and those who completed vaccination. We used a predictive model that learned only corona confirmed cases information, subdivided it by parameters, and proposed an accurate and effective predictive model for the number of corona confirmed cases. Neural networks, ensembles, distance-based models, and linear regression were used as supervised learning models for various machine learning models. As for the model with excellent predictive power, the training scores such as Gradient Boosting and AdaBoosting had high training scores. CatBoost showed the best predictive power among the Gradient Boosting models through cross-validation by model. About 94.8% of the predictions were accurate.

According to CatBoost's prediction results, the predictive power was poor in the area where the number of confirmed cases due to the mutated virus rapidly increased. In particular, it was confirmed that the CatBoost model was effective in predicting small and irregular infections in the early stage, but it was confirmed that the prediction of the period of the rapid increase in the number of confirmed cases due to delta mutation was somewhat ineffective. As a future research task, it is necessary to implement and compare prediction algorithms using machine learning techniques trained in unsupervised learning. In addition, it is necessary to make a prediction using the policy variables to be considered, such as the stage and implementation of the social distancing movement.

Author Contributions

Conceptualization, J.Moon and J.Lee; Methodology, J.Moon; Project Administration, H.Lee and H.Yun; Data curation, J.Moon and H.Lee; Result data acquisition, J.Moon and J.Lee; Wrighting-original draft preparation, J.Moon; Visualization, H.Lee; Funding acquisition, J. Moon and H.Yun; Supervision, J. Moon and H.Yun. All authors have read and agreed to the published version of the manuscript."

Funding

This research was supported by a grant (2021-MOIS61-02-0000-2021) of Development of location oriented virus safety map funded by the Ministry of Interior and Safety (MOIS, Korea).

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. Ceylan Z (2020) Estimation of COVID-19 prevalence in Italy, Spain, and France. Sci Total Environ 729: 138817.
2. Chimmula VKR, Zhang L (2020) Time series forecasting of COVID-19 transmission in Canada using LSTM networks. Chaos Solitons & Fractals 135: 109864.
3. Yang Z, Zeng Z, Wang K, Wong SS, Liang W, et al. (2020) Modified SEIR and AI prediction of the epidemics trend of COVID-19 in China under public health interventions. J Thorac Dis 12: 165-174.
4. He S, Peng Y, Sun K (2020) SEIR modeling of the COVID-19 and its dynamics. Nonlinear Dyn 101: 1667-1680.
5. Arora P, Kumar H, Panigrahi BK (2020) Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India. Chaos, Solitons & Fractals 139: 110017.
6. Pandey G, Chaudhary P, Gupta R, Pal S (2020) SEIR and Regression Model based COVID-19 outbreak predictions in India. arXiv: 200400958.
7. (2021) Cssegisand Data COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.
8. Alzahrani S, Aljamaan I, Al-Fakih E (2020)Forecasting the Spread of the COVID-19 Pandemic in Saudi Arabia Using ARIMA Prediction Model Under Current Public Health Interventions. Journal of Infection and Public Health 13: 914-919.
9. Pinter G, Felde I, Mosavi A, Ghamisi P, Gloaguen R (2020) COVID-19 Pandemic Prediction for Hungary, A Hybrid Machine Learning Approach. Mathematics 8: 890.
10. Jino K, Jisu K, Heewon K, Hyosang P, et al. (2020) Comparison and Analysis of COVID-19 Confirmed Cases Based on the SIR model and LSTM. Korea Intelligent Information Systems Society, p. 59-64
11. Baejinsoo, Bum KS (2021) Predictions of COVID-19 in Korea Using Machine Learning Models. Journal of the Korean Institute of Industrial Engineers 47: 272-279.
12. Shi P, Dong Y, Yan H, Zhao C, Li X, et al. (2020) Impact of temperature on the dynamics of the COVID-19 outbreak in China. Science of The Total Environment 728: 138890.
13. To WM (2020) How Big is the Impact of COVID-19 (and Social Unrest) on the Number of Passengers of the Hong Kong International Airport?,
14. Kumar A(2020) Modeling geographical spread of COVID-19 in India using network-based approach.
15. Kim M, Kang J, Kim D, Song H, Min H, et al. (2020) Hi-COVIDNet: Deep Learning Approach to Predict Inbound COVID-19 Patients and Case Study in South Korea. 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2020: 3466-3473.
16. Qin L, Sun Q, Wang Y, Wu KF, Chen M, et al. (2020) Prediction of Number of Cases of 2019 Novel Coronavirus (COVID-19) Using Social Media Search Index. Int J Environ Res Public Health 17(7): 2365.
17. Jahanbin K, Rahmanian V (2020) Using twitter and web news mining to predict COVID-19 outbreak. Asian Pacific Journal of Tropical Medicine 13: 378-380.
18. Li C, Chen LJ, Chen X, Zhang M, Pang CP, et al. (2020) Retrospective analysis of the possibility of predicting the COVID-19 outbreak from Internet searches and social media data, China, 2020. Euro Surveill 25(10): 2000199.
19. Prata DN, Rodrigues W, Bermejo PH (2020) Temperature significantly changes COVID-19 transmission in (sub)tropical cities of Brazil. Science of The Total Environment 729: 138862.

ISSN: 2574-1241

DOI: 10.26717/BJSTR.2022.45.007184

Jaejoon Lee. Biomed J Sci & Tech Res