The quest for better machine learning models to forecast COVID-19-related infections: A case study in the state of Pará-Brazil

Renato Hidaka Torres (✉ renatohidaka@ufpa.br)
UFPA
Wilson Rogério Soares
IFPA
Orlando Shigueo Ohashi
UFRA
Gustavo Pessin
ITV

Research Article

Keywords: machine learning, forecasting, hypothesis test, COVID-19

Posted Date: June 26th, 2020

DOI: https://doi.org/10.21203/rs.3.rs-37547/v1

The quest for better machine learning models to forecast COVID-19-related infections: A case study in the state of Pará-Brazil

Renato H. Torres[1,*], **Wilson R. Soares**[2], **Orlando S. Ohashi**[3], **and Gustavo Pessin**[4]

[1]Federal University of Pará, Institute of Exact and Natural Sciences, Belém – PA, Brazil
[2]Federal Institute of Pará, IT department, Vigia – PA, Brazil
[3]Federal Rural University of Amazônia, Cyberspace Institute, Belém – PA, Brazil
[4]Instituto Tecnológico Vale, Robotics Lab, Ouro Preto–MG, Brazil
[*]e-mail: renatohidaka@gmail.com

ABSTRACT

COVID-19 disease has become an unprecedented public health crisis. Although a relatively small percentage of people require intensive care, due to the high degree of contagion of the disease, the public system quickly collapses. Due to the highly complex nature this disease and variation in its behavior depending on the characteristics of each geographic region, in this work we analyze data from the Amazon region in Brazil (Pará). We applied several machine learning models to forecast the contagious curve to up 10 days. The Linear SVM and Multilayer Perceptron presented the best overall performances. Until the discovery of a vaccine, every effort is needed to understand and anticipate this disease.

Introduction

COVID-19 disease has become an unprecedented public health crisis. Due to its speed of proliferation, the total of confirmed cases and deaths is increasing every day. According to the World Health Organization (WHO) situation report - 122[1], until May 21, 2020, worldwide, the total confirmed cases and deaths were 4.893.186 and 323.256 respectively. Comparing these data with the data released in the first WHO situational report[2], the number of deaths increased by 32.325.500% from January to May. With the exponential growth of the number of cases and deaths, Brazil is becoming the new epicenter of covid-19. These data are extremely worrying and, for this reason, researches and measures are being adopted to combat this pandemic. While the vaccine for this disease is not found, one of the most efficient ways to mitigate COVID-19 is information assist. Assist people with the importance of social isolation and preventive measures is essential to reduce the number of deaths. Besides, COVID-19-related issues can be analyzed by information systems to better understand the pattern of viral spread. For example, to analyze the growing trend in the number of confirmed cases and deaths can be provided by information systems that applied machine learning techniques. The prediction made by the machine learning models can be important information for strategic decisions performed by authorities, for example, to determine the institution of hospitals, acquisition of equipment, loosening or restriction of social isolation, or even the determination of lockdown.

In this context, some studies have proposed mathematical and machine learning models to predict the trend of the growth curve in the number of confirmed cases and deaths due to COVID-19. Works like[3–7] are examples of research that uses machine learning (ML) models along these lines. From the mentioned works, due to the highly complex nature of the COVID-19 disease and variation in its behavior from nation-to-nation, we note that two questions are central for the correctness models accuracy: a) the geographic context of the data, i.e. different countries and states show different patterns about the COVID-19 outbreak. b) the prediction window size in the number of days, i.e. short predictor windows have better accuracy. Considering these two factors, we understand that it is extremely relevant to further investigate ML models to predict cases and deaths of COVID-19-related in specific regions of Brazil. In this case, we investigated the pattern of infection in the state of Pará. According to IBGE data[8], Pará is a state of Brazil, located in the Amazon Rainforest that has 144 counties, distributed in an large area of 1.247.955 km^2, approximately 8 million inhabitants, and 13.720 hospital beds between public and private, 580 are UCI. Until May 21, 2020, the state of Pará had 20537 confirmed cases and 1893 deaths due to Covid-19[9]. In a statement on April 21, 2020, authorities said that 90% of the hospital beds in the state of Pará are already occupied. This news is alarming and highlights the idea that the information system as proposed in this work is promising to assist future public policy health care interventions.

This work aims to compare different machine learning models for forecasting COVID-19 infection cases in the Pará-Brazil state. Considering the collected data, we will also analyze the ideal prediction window size. In summary, this work evaluates different prediction windows Ω and the generalization capacity of machine learning models Ψ, in the task of estimating the number of infected due to COVID-19 in the state of Pará Φ, i.e., we evaluate how Φ can be solved by $\Psi(\Omega)$ and also which the influence of $\omega \in \Omega \; \forall \; \psi \in \Psi$. For the Ω set, we consider windows of up to 10 days, i.e., given the values of the attributes for a given day N, we built the target value to predict the total number of infected on day $N+d$, where d goes from 1 to 10. Regarding the Φ set, we consider six machine learning models and two statistical models, namely: Linear Regression, Linear SVM, Random Forest, Gradient Boost, Convolutional Neural Network, Multilayer Perceptron, Autoregression, and Prophet models. Among these machine learning models we have three classes of algorithms, which are: linear model (Linear Regression and Linear SVM), ensemble model (Random Forest and Gradient Boost), and neural network model (Convolutional Neural Network and Multilayer Perceptron).

Ensemble and deep learning models were taking into account mainly because the state of the art shows that these models are the best techniques to solve most problems where machine learning can be employed. Linear models were applied based on the grounds of Occam's razor principle[10], which states that the explanation of any phenomenon must assume the least number of premises possible. In machine learning, this often means that when faced with two classified algorithms with the same training performance and testing capacity, the simplest model will probably be the best choice. Taking into account the approach described, we consider that our work has three central contributions: (1) construction of efficient models to predict the number of infected due to COVID-19 in the state of Pará. (2) performance evaluation of models for different prediction windows. (3) comparison of different classes of machine learning models to apply Occam's razor principle. We use the metrics RMSE, RMSRE, MAE, MAPE, and R^2 to examine the performance of the models. The results show that Occam's razor principle was applied and that, despite the ensemble and deep learning models are the best techniques to solve most problems where machine learning can be employed, there are exceptions.

The remainder of the article is structured in the following way. In Section 2, there is a brief review of the literature on the most common ML employed to forecast COVID-19 outbreak. In Section 3, we describe how the data collection was carried out, and the database was formed. The designing and analysis of the models are carried out in Section 4. Finally, Section 5 summarizes the main conclusions of the research study.

Related Works

Machine learning is the field of artificial intelligence whose objective implies the development of computational algorithms that are capable of transforming experience into expertise[11]. In other words, machine learning algorithms aim to map patterns from an input domain to an output domain and subsequently recognize patterns from the output domain, even those not exemplified. Given the ability to generalize, ML techniques are applied in many areas to solve classification or prediction problems. For example, on the COVID-19 outbreak, ML techniques can be used to understand the pattern of viral spread, improve diagnostic accuracy, develop novel effective therapeutic approaches, and identify the most susceptible people based on personalized genetic and physiological characteristics[12]. In the review conducted by Bullock et al.[13], they present an overview of recent studies using Machine Learning and, more broadly, Artificial Intelligence, to tackle many aspects of the COVID-19 crisis at different scales including molecular, clinical, and societal applications. In total, 20 application areas of machine learning are presented, and 82 works that developed applications are presented in the review.

According to the applications cited by Bullock et al.[13], our work is in the category: Societal scale - Epidemiology and infodemiology. Researches in this subject want to understand how the virus is transmitted, and its likely effect on different demographics and geographic locations. These researches are therefore crucial for public policy health care interventions. In this type of problem, many well-established classical models, such as susceptible-infected-recovered (SIR) models are fine-tuned to the COVID-19 situation[14]. However, most ML applications developed for epidemiological modeling have presented promising results on forecasting national and local statistics such as the total number of confirmed cases, mortality, and recovery rates. In this context, we correlated our work with research that developed ML models with a methodology similar to that presented in this work. Table 1 shows researches that have improved machine learning models in the context of forecasting the COVID-19 global pandemic problem.

Ardabili et al.[3] showed promising results when using multi-layered perceptron, MLP, and adaptive network-based fuzzy inference system, ANFIS. In that work they collected data from five countries, including Italy, Germany, Iran, USA, and China on total cases over 30 days. The evaluation was conducted using the root mean square error (RMSE) and the correlation coefficient. The results show that the accuracy of the developed models is better than the accuracy of classic mathematical models such as logistic and linear.

Huang et al.[4] proposed a Convolutional Neural Network (CNN) to analyze and predict the number of confirmed cases in China. In this study, the input data were obtained from Surging News Network and WHO, respectively. To compare the overall efficacies of different algorithms, the indicators of mean absolute error (MAE) and root mean square error (RMSE) was applied

Reference	Technique	Dataset	Metrics
Ardabili et al.[3]	Multi-Layered Perceptron Adaptive Network-Based Fuzzy Inference System	own data collected in Italy Germany Iran USA China	RMSE correlation
Huang et al.[4]	Convolutional Neural Network	Surging News Network WHO situation reports	RMSE MAE
Al-qaness et al.[5]	Adaptive Network-Based Fuzzy Inference Flower Pollination Algorithm Salp Swarm Algorithm	WHO official data	MAE MAPE RMSE MAE R^2
Ceylan[6]	Auto-Regressive Integrated Moving Average	WHO official in Italy Spain France	RMSE MAPE
Dutta and Bandyopadhyay[7]	Long Short-Term Memory Gated Recurrent Unit	Kaggle dataset	RMSE Accuracy

Table 1. Related works that used methodology and metrics similar to the one used in our work.

in the experiment. The results show that the solution proposed by them achieves high predictive precision, even using small data sets.

Al-qaness et al.[5] conducted a case study in China to explore the COVID-19 prediction problem over a 10-day horizon. They developed an improved adaptive neuro-fuzzy inference system, (ANFIS) using an enhanced flower pollination algorithm (FPA) and salp swarm algorithm (SSA). For training and evaluation of the models, they used official data from the World Health Organization (WHO). To evaluate the performance of the models, they used the metrics mean absolute percentage error (MAPE), root mean squared relative error (RMSRE), and coefficient of determination (R^2). In comparison with different ML methods, the study shows that the proposed model has a high ability to forecast the COVID-19 dataset.

Ceylan[6] developed an Auto-Regressive Integrated Moving Average (ARIMA) model to predict the epidemiological trend of COVID-19. They took into account the cases reported in Italy, Spain, and France. In this study, they used official data released by the World Health Organization, from February 2020 to 15 April 2020. To evaluate the performance of the models, they used the metrics root mean square error (RMSE), mean absolute error(MAE) and mean absolute percentage error (MAPE). Considering the analysis of the results, they argue that the ARIMA model performs well to predict the cases of COVID-19 in the future.

Dutta and Bandyopadhyay[7] proposed a deep learning neural network to predict cases of confirmed, negative, released, deceased in Covid-19. In this study, the authors compare the performance of the Long short-term memory (LSTM) and Gated Recurrent Unit (GRU) models. According to the analysis of the results, they point out that the prediction of the models is compatible with the results predicted by clinical doctors. To conduct the training and evaluation of the models, the authors used a public dataset from the Kaggle repository that contains reported cases of COVID-19 in the period from 20th January 2020 to 12th March 2020. To evaluate the performance of the models, they used root mean square error (RMSE) and accuracy. The results show that the proposed deep learning models have a high prediction rate for the disease identification.

Looking at related work, we can see that neural networks and deep learning techniques are being used to solve the forecasting problem related to COVID-19. In addition, the evaluation metrics and the analysis of the results are performed in a similar way in all works. Considering this scenario, our work aims to compare different machine learning models for forecasting COVID-19 infection cases in the Pará-Brazil state. We will also analyze the ideal prediction window size and we will use the metrics RMSE, RMSRE, MAE, MAPE and R^2 to analyze the performance of the models.

Building dataset and choosing the models

COVID-19 cases are reported daily by the public health department of the state of Pará. So, we built our dataset collecting the cases reported by this department[9]. Data were collected from March 17, 2020 to May 21, 2020. In total we collected 66 instances containing the following attributes: total of confirmed cases, total of deaths, total of discarded cases, and total of recovered cases. Figure 1 shows the behavior of the data collected over the days. In the last ten days, we can see considerable growth in the number of confirmed cases, recoveries, and deaths. However, as the number of confirmed cases is greater than the number of people recovered, the situation is still worrying.

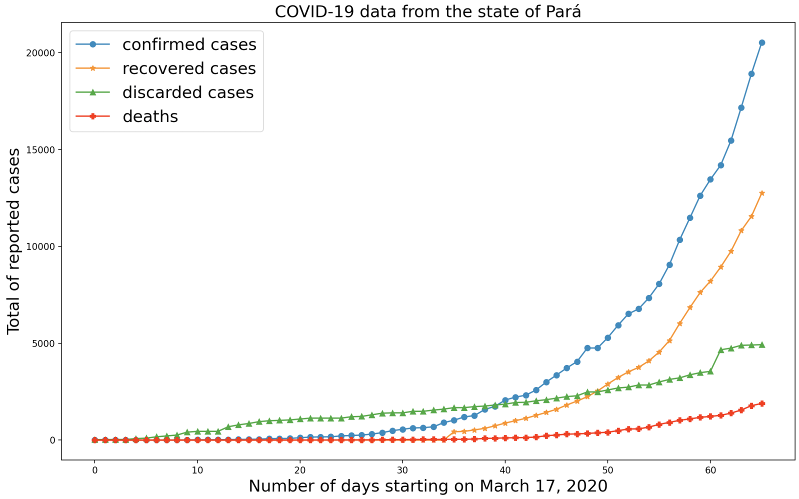

Figure 1. Official numbers of cases reported by the public health department of the state of Pará.

Analyzing Figure 1, it may not be possible to notice the growth in the number of deaths. However, when we analyze Figure 2, it is possible to observe that the number of deaths is also increasing in the last ten days. This behavior reinforces the hypothesis that COVID-19 related problems tend to increase in the state of Pará.

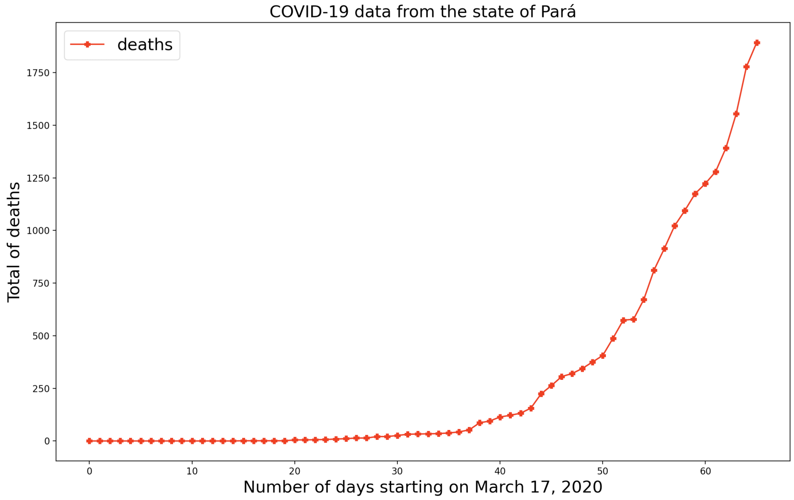

Figure 2. Official numbers of deaths reported by the public health department of the state of Pará.

Considering the information collected, we built our dataset with 8 attributes as shown in Table 2. To build the target variable, we re-frame the time series dataset as a supervised learning problem using the sliding window method. That is, given the values of the attributes for a given day N, we built the target value to predict the total number of infected on day $N+d$, where d goes from 1 to 10. Altogether we have 10 different target values which imply 10 training and analysis of the models.

To build the most appropriate machine learning models, we analyzed the linear correlation of the attributes of our dataset (see Figure 3).

1- Total confirmed cases	2- Total discarded cases	
3- Total deaths	4- Total recovered cases	5- Confirmed / 100k inhabitants
6- Deaths / confirmed	7- New cases in one day	8- New deaths in one day

Table 2. Dataset attributes.

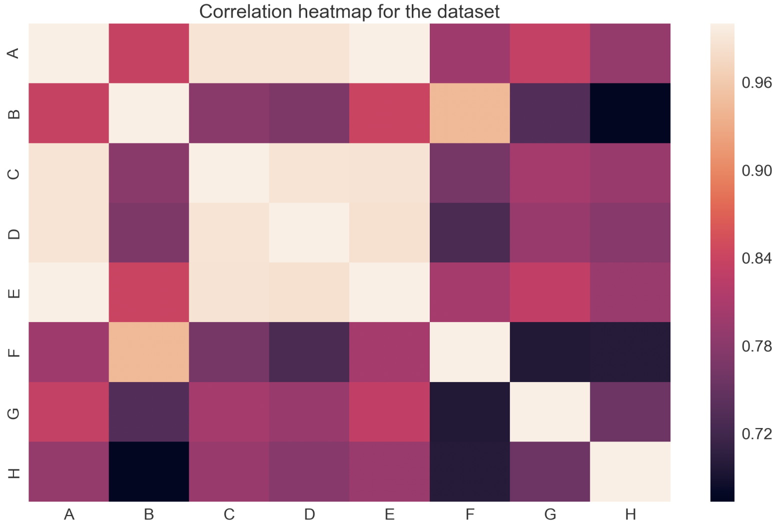

Figure 3. Correlation Heatmap. **A**=Confirmed cases; **B**=Discarded cases; **C**=Deaths; **D**=Recovered cases; **E**=Confirmed/100k inhabitants; **F**= Deaths/confirmed; **G**= New cases in one day; **H**= New deaths in one day.

As we can see in the heatmap, many attributes are linearly dependent on our target attribute (A). This behavior suggests that linear machine learning models will be more efficient in solving our problem. Although the state of the art shows that ensemble and deep learning models are the best techniques to solve most problems where machine learning can be employed, there are exceptions. In fact, the best ML depends on the patterns of input data. In our case, due to the correlation of the data, we assume the hypothesis that the linear models are better than the ensemble and deep learning models.

To prove our hypothesis we evaluated the performance of two linear models (Linear Regression and Linear SVM), two ensemble models (Random Forest and Gradient Boost), and two neural network models (Convolutional Neural Network and Multilayer Perceptron). Besides, we also compare these machine learning models with two statistical prediction models: AutoRegression and Prophet models[15].

Modeling and evaluation

To perform the modeling of ML models, we built a pipeline with grid-search and cross-validation techniques. In machine learning, we can configure hyperparameters manually from the designer's experience, using heuristics or using brute force techniques such as grid search. In the grid search, we must define the hyperparameters and the range of values that are combined when adjusting the model[16]. In summary, in the grid search technique, each combination of values represents a model that must be validated. In this experiment, we define the range of values of the hyperparameters to verify which configuration provides the best generalization capacity for the models.

Regarding cross-validation, we use it to evaluate the models and maximize the number of validation samples. The idea of cross-validation is that each sample of the data set is used to test the model's performance. Due to the nature of the data we are using, we apply the time series cross-validation known as walk-forward cross-validation. According to Hyndman and Athanasopoulos[17], in this procedure, there are a series of test sets, each consisting of a single observation. The corresponding training set consists only of observations that occurred before the observation that forms the test set. Thus, no future observations can be used in constructing the forecast. This approach is interesting because it allows the model to be evaluated with the largest number of samples and eliminates the chronological order problem caused by other cross-validation methods, such as k-fold cross-validation, for example. Formally, we can define walk-forward cross-validation as an iterative assessment that works as follows: let $\{t_1, t_2, ...t_N\}$ be a time series, the training and test set is given by training $= \{t_1, t_2, ...t_{k+i}\}$ and test $= \{t_{k+i+w}\}$

$\forall\ 0 \leq i \leq N - k - w$ where k is the initial size of the training set and w is the prediction window.

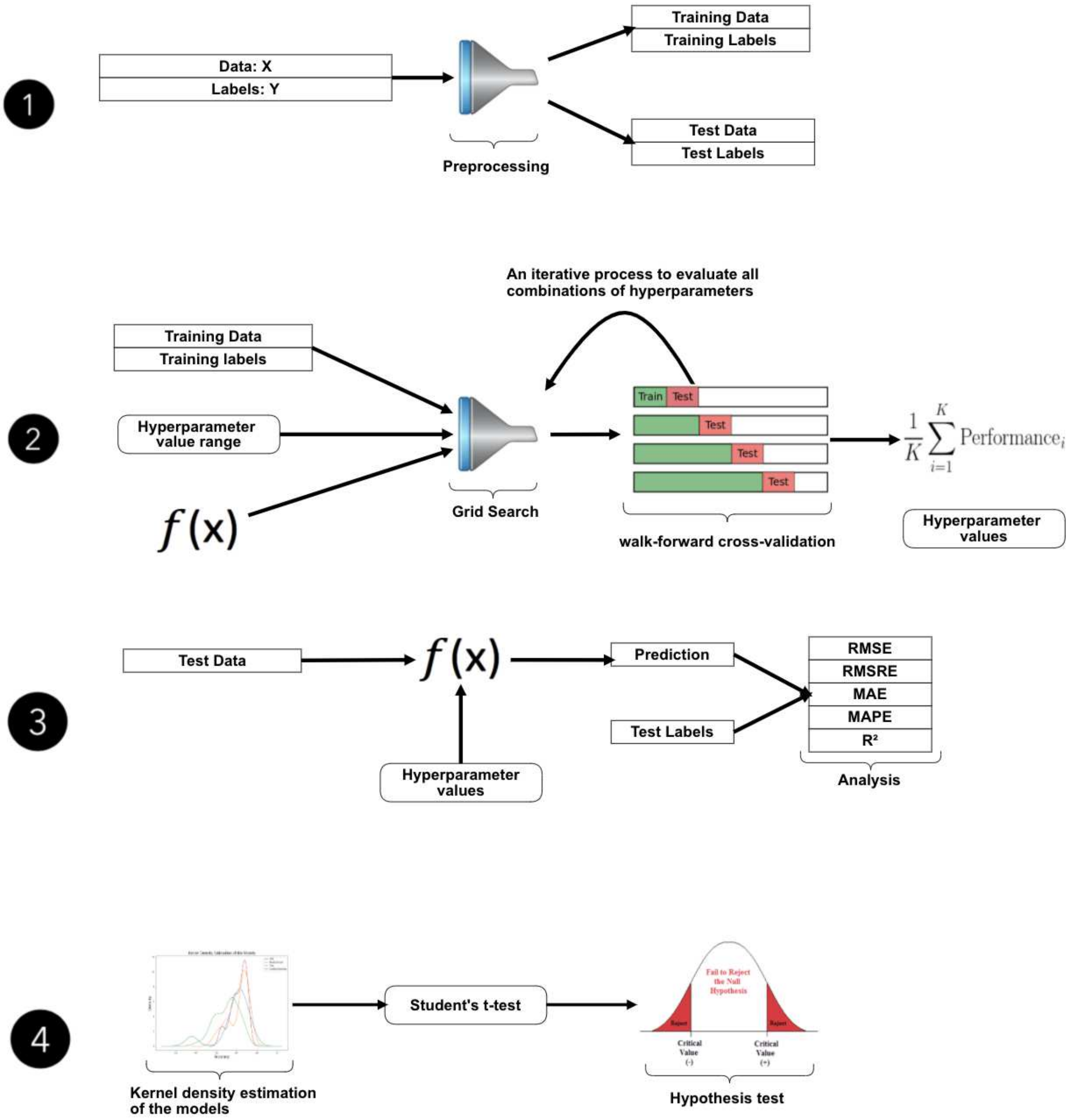

Figure 4. Methodology to assess the performance of the models. Step 1: split the data set into training and testing subsets. Step 2: application of grid search and cross-validation to train and select the hyperparameters of the models. Step 3: evaluation of the models with the test subset. Step 4: hypothesis test to verify the statistical significance of the best models evaluated.

Figure 4 describes the methodology that we use for the construction and evaluation of the prediction models. We use this methodology for each window prediction. In step 1, we split the data into training and test sets. The test set was built to test the prediction performance of the models for the last ten days. The set's shape depends on the prediction window to be tested. For example, when the window is N + 1, we use the data collected on May 1st to predict the number of cases on May 2nd, the data on May 2nd to predict the cases on May 3rd, and so on. When the window is N + 10, we use the data collected on April 29th to predict the number of cases on May 9th, the data on April 30th to predict the cases on May 1st and so on. This split procedure was used for each N + d window where $1 \leq d \leq 10$.

In step 2, we perform the pipeline for each machine learning model. In this step, we define the range of values for each hyperparameter and provide it as input data from the pipeline together with the training data and the model. The grid search is performed for each hyperparameter combination and evaluated with walk-forward cross-validation. At the end of the pipeline, for each ML, we have the best combination of hyperparameters, as well as the cross-validation evaluation performance. In step 3, we used the best configuration for each model and tested their generalizability. We use the data from the test set and evaluate the results with the metrics RMSE, RMSRE, MAE, MAPE and R^2. In this step, the Autoregressive and Prophet statistical models were also evaluated. Finally, in step 4, we conducted a hypothesis test to check the equivalence between the models. In this test, the objective is to verify whether the simplest models are equivalent to the more complex models. If so, we can apply Occam's razor principle and choose the simplest models to solve the problem.

Setting up the Hyperparameters

Table 3 shows the configuration parameters and the range of values that we pre-determine for each model. The Linear Regression model was the only one that we did not submit to the grid search because it does not have hyperparameters. The parameters of the Regression Linear are all adjusted during supervised training.

Model	**C**				
Support Vector Machine	range (0.001, 1000.0, 10)				
Model	**N estimators**	**Criterion**	**Min sample split**	**Max depth**	**Max features**
Random Forest	range (75, 200, 25)	MSE MAE	range (5, 45, 5)	range (2, 10, 2)	$\sqrt{n}$ $log_2(n)$
Model	**N estimators**	**Learning rate**		**Min sample split**	**Max depth**
Gradient Boosting	range (75, 200, 25)	range (0.1, 1.0, 0.1)		range (5, 45, 5)	range (2, 10, 2)
Model	**N layers**			**N filters**	
Convolutional Neural Network	range (1, 6, 1)			range (10, 60, 10)	
Model	**N layers**			**N units**	
Multilayer Perceptron	range (1, 6, 1)			range (10, 60, 10)	

Table 3. Models' grid search.

In the SVM model, we set up the C parameter. A large C value makes the model to use a lower margin hyperplane, while a small C value makes the model to look for a larger margin separating hyperplane. In Table 3, we configure the C range to test super-adjusted and smooth hyperplanes. As noted in Table 4, for all prediction windows, the grid search selected small values for C. These small values indicate that the hyperplanes of each SVM model are not overfitted and, for this reason, the built models can achieve a better generalization capacity.

In the Random Forest model, we have configured 4 hyperparameters. The N estimators parameter determines the number of trees in the forest. The higher the value of N estimators, the larger the ensemble. In the Criterion parameter, we configure the function to measure the quality of the splits that are performed during the construction of the trees. As it is a regression problem, we selected the MAE and MSE criteria. Max depth and Min sample split are parameters used to determine the tree configuration. In Table 3, we configure the hyperparameters to search for forests with shallow and deep trees, however, as noted in Table 4, the grid search selected trees with intermediate depth. trees.

In the Gradient Boosting models, the N estimator defines the number of boosting stages to perform. The boosting models are fairly robust to overfitting, so a large number usually results in better performance. In the learning rate, the set value has the purpose to define the contribution of each classifier. The nearer to zero is the learning rate, it means that the classifiers, individually, will have less contribution to the ensemble, i.e., for a small learning rate, the ensemble will take into account the joint performance of the classifiers. The parameters Min sample split and Max depth were also configured. In this case, these parameters have the same purpose as the configuration performed in the Random Forest model. Considering the ranges of values presented in Table 3, we can see that the grid search selected the values to configure Gradient Boosting models with many estimators of low learning rate and moderate depth. Table 4 shows the configuration selected for each prediction window.

For the Convolutional Neural Network and Multilayer Perceptron models, we configured the grid search to test the best architecture taking into account the number of layers and the number of neurons. As noted in Table 3, we varied the number of layers from 1 to 6 and the number of neurons from 10 to 60. To decrease the complexity of the grid search, we defined that for each tested architecture, the number of neurons would be the same in all layers. In the case of CNN, we do not perform the grid search for the filter size. Taking into account the observation of Szegedy et al[18], the number of each filter was fixed at 2. In the development of CNN, Szegedy et al. show that the use of small kernels is more efficient than the use of larger kernels. In addition to decreasing the processing load, they also emphasize that the use of multiple small filters can match the representativeness of larger filters. The concept of representativity is related to the capacity of the convolution to be able to detect structural changes in the analyzed problem. In Table 4, we can see the architecture of the neural networks defined for each prediction window.

Windows	SVM	Random Forest	Gradient Boosting	CNN	Multilayer Perceptron
1	C=0.001	N estimators = 75 Criterion = MAE Min sample split = 5 Max depth = 4	N estimators = 100 Learning rate = 0.2 Min sample split = 10 Max depth = 6	N layers = 6 N filters = 40	N layers = 1 N filters = 60
2	C=1	N estimators = 125 Criterion = MSE Min sample split = 5 Max depth = 6	N estimators = 150 Learning rate = 0.1 Min sample split = 20 Max depth = 4	N layers = 4 N filters = 50	N layers = 3 N filters = 30
3	C=1	N estimators = 125 Criterion = MSE Min sample split = 5 Max depth = 6	N estimators = 150 Learning rate = 0.2 Min sample split = 20 Max depth = 2	N layers = 3 N filters = 60	N layers = 4 N filters = 10
4	C=1	N estimators = 100 Criterion = MSE Min sample split = 5 Max depth = 4	N estimators = 125 Learning rate = 0.2 Min sample split = 35 Max depth = 6	N layers = 4 N filters = 60	N layers = 2 N filters = 10
5	C=0.1	N estimators = 100 Criterion = MSE Min sample split = 5 Max depth = 4	N estimators = 200 Learning rate = 0.1 Min sample split = 20 Max depth = 4	N layers = 5 N filters = 60	N layers = 1 N filters = 60
6	C=0.1	N estimators = 75 Criterion = MAE Min sample split = 5 Max depth = 6	N estimators = 125 Learning rate = 0.2 Min sample split = 5 Max depth = 8	N layers = 5 N filters = 20	N layers = 1 N filters = 50
7	C=0.1	N estimators = 100 Criterion = MAE Min sample split = 5 Max depth = 8	N estimators = 200 Learning rate = 0.4 Min sample split = 25 Max depth = 4	N layers = 6 N filters = 60	N layers = 3 N filters = 10
8	C=0.001	N estimators = 150 Criterion = MAE Min sample split = 5 Max depth = 8	N estimators = 200 Learning rate = 0.2 Min sample split = 15 Max depth = 8	N layers = 5 N filters = 40	N layers = 3 N filters = 20
9	C=0.1	N estimators = 125 Criterion = MSE Min sample split = 5 Max depth = 8	N estimators = 175 Learning rate = 0.5 Min sample split = 10 Max depth = 8	N layers = 4 N filters = 10	N layers = 2 N filters = 30
10	C=0.1	N estimators = 125 Criterion = MSE Min sample split = 5 Max depth = 8	N estimators = 175 Learning rate = 0.5 Min sample split = 10 Max depth = 8	N layers = 5 N filters = 60	N layers = 2 N filters = 60

Table 4. Hyperparameters setting by model and prediction window combination.

Evaluation

After training the models, we assess the generalizability of the models. In this step, we use the metrics MSE eq 1, RMSE eq 2, MAE eq 3, MAPE eq 4, and R^2 eq 5 to measuring the algorithm's performance. For the best interpretation of the results, we apply the Min-Max normalization to the expected and predicted values. Thus, for the metrics MSE, RMSE, MAE and MAPE, the values closer to zero indicate the best performance of the evaluated model. For the metric R2, the best performance is observed when the values are close to one. Tables 5 show the performance of each model by the prediction window.

$$MSE = \frac{1}{n}\sum_{i=i}^{N}(y_{true} - y_{pred})^2 \tag{1}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=i}^{N}(y_{true} - y_{pred})^2} \tag{2}$$

$$MAE = \frac{1}{n}\sum_{i=i}^{N}|y_{true} - y_{pred}| \tag{3}$$

$$MAPE = \frac{1}{n}\sum_{i=i}^{N}|\frac{y_{true} - y_{pred}}{y_{true}}| \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{true} - y_{pred})^2}{\sum_{i=1}^{n}(y_{true} - \bar{y})^2} \tag{5}$$

From all the models analyzed, we can see that the Random Forest and Gradient Boosting models did not reach generalization in any prediction window. This behavior can be explained due to the way these algorithms work. Random Forest and Gradient Boosting are tree-based algorithms that operate by if-then rules that recursively split the input space. These algorithms consider the observations to be independent and identically distributed and, therefore, they're unable to predict values that fall outside the range of values of the target in the training set, i.e., they're unable to predict a trend. To try to solve this problem, we use data transformation techniques that are widely used in time series[17]. In the pre-processing step, we applied the Box-Cox transformation to fix the variance and the differentiation between the cases reported to make it stationary in the mean. However, the Random Forest and Gradient Boosting models still failed to generalize. The performance of the Random Forest and Gradient Boosting models proves our hypothesis that despite the ensemble models are the best techniques to solve most problems where machine learning can be employed, there are exceptions.

About the other models, we can see that there was a variation in performance due to the prediction window. To make easing the visualization of these variations, we group the performance of the models and plot the graphs presented in Figures 5 and 6. In Figure 5, we have the radar charts comparing the performance of the models for each prediction window. We plotted the radar chart, taking into account the 5 evaluation metrics. However, for better visualization, we inverted the MAE, MAPE, MSE, and RMSE metrics, i.e., we apply $1 - \text{metric}$. With this transformation, the models with the best performance should present a performance closer to 1 for all metrics. In visualization, the best model is the one with the largest pentagon. In Figure 6, we are plotting each model's prediction according to the used prediction window. This plot refers to the test data that was not used in the training and validation phase. The black line corresponds to the confirmed true cases and the dashed lines to the model's predictions. Predictions were made for the 12th to the 21st of May 2020. In window 1, to forecast the 12th we use the data from the 11th, to predict the data from the 13th we use the data from the 12th and so on. Generalizing, considering a window w, the prediction of the day p_d is performed using the data of the day p_{d-w}.

Looking at the results, we can see that the Linear Regression model was losing performance as the prediction window grew. In Figure 5, we can see that the pentagon of Linear Regression is similar to the best models for the first 5 windows. However, for windows 6 to 10, the low performance of this model is notable. This behavior can also be seen in the predictions in Figure 6. As the prediction window increases, the (blue) curve of the Linear Regression model moves away from the true (black) cases. The results of the Linear Regression show that this model is only efficient for forecasting small horizons. In summary, to predict horizons > 5, we do not recommend using the Linear Regression model.

The Convolutional Neural Network models showed performance variation between the prediction windows. In Figure 5, in the prediction windows 5, 7, 8, and 10, we can see that the CNN pentagon (purple) is significantly smaller than the pentagon of

Windows	Metrics	LR	SVM	RF	GB	CNN	MLP
1	*MAE*	0.0321	**0.0219**	0.6024	0.5682	0.0372	0.0381
	MAPE	0.1143	**0.0704**	1.2866	1.1943	0.0867	0.0870
	MSE	0.0013	**0.0007**	0.4575	0.4170	0.0018	0.0021
	RMSE	0.0366	**0.0261**	0.6764	0.6458	0.0429	0.0456
	R^2	0.9858	**0.9928**	-3.8389	-3.4114	0.9806	0.9780
2	*MAE*	0.0754	**0.0454**	0.6527	0.5655	0.1017	0.0533
	MAPE	0.2487	**0.1600**	1.4220	1.1883	0.3004	0.1780
	MSE	0.0077	**0.0027**	0.5215	0.4143	0.0110	0.0035
	RMSE	0.0879	**0.0516**	0.7221	0.6436	0.1050	0.0591
	R^2	0.9183	**0.9719**	-4.5160	-3.3822	0.8833	0.9631
3	*MAE*	0.1372	0.0916	0.5711	0.6706	0.0739	**0.0658**
	MAPE	0.3878	0.2971	1.1580	1.4744	0.2546	**0.2317**
	MSE	0.0215	0.0113	0.4176	0.5446	0.0071	**0.0062**
	RMSE	0.1466	0.1062	0.6462	0.7380	0.0842	**0.0787**
	R^2	0.7727	0.8808	-3.4176	-4.7611	0.9250	**0.9344**
4	*MAE*	**0.0489**	0.1110	0.7273	0.5563	0.0924	0.0523
	MAPE	**0.1734**	0.3244	1.6323	1.1994	0.2863	0.1943
	MSE	**0.0037**	0.0148	0.6228	0.4015	0.0106	0.0039
	RMSE	**0.0612**	0.1217	0.7892	0.6337	0.1030	0.0626
	R^2	**0.9604**	0.8435	-5.5881	-3.2475	0.8878	0.9585
5	*MAE*	0.0506	0.0724	0.5770	0.5954	0.1491	**0.0477**
	MAPE	0.1763	**0.1440**	1.2308	1.2703	0.3790	0.1621
	MSE	0.0049	0.0072	0.4249	0.4489	0.0256	**0.0038**
	RMSE	0.0698	0.0851	0.6519	0.6700	0.1601	**0.0616**
	R^2	0.9484	0.9233	-3.4950	-3.7479	0.7290	**0.9599**
6	*MAE*	0.1285	0.1036	0.6218	0.5763	**0.0485**	0.0496
	MAPE	0.3048	0.2581	1.3476	1.2105	**0.1131**	0.1259
	MSE	0.0182	0.0123	0.4790	0.4245	**0.0034**	0.0037
	RMSE	0.1349	0.1109	0.6921	0.6516	**0.0580**	0.0612
	R^2	0.8076	0.8698	-4.0673	-3.4906	**0.9644**	0.9604
7	*MAE*	0.1559	**0.0371**	0.5945	0.5818	0.1271	0.0459
	MAPE	0.3498	**0.0475**	1.2652	1.2071	0.2988	0.1032
	MSE	0.0314	**0.0026**	0.4489	0.4315	0.0181	0.0029
	RMSE	0.1772	**0.0512**	0.6700	0.6569	0.1345	0.0537
	R^2	0.6677	**0.9722**	-3.7480	-3.5643	0.8086	0.9695
8	*MAE*	0.1479	**0.0376**	0.7502	0.5863	0.1138	0.0716
	MAPE	0.3068	**0.0844**	1.7319	1.2482	0.3160	0.1928
	MSE	0.0300	**0.0020**	0.6551	0.4385	0.0150	0.0058
	RMSE	0.1732	**0.0446**	0.8094	0.6622	0.1223	0.0761
	R^2	0.6826	**0.9790**	-5.9292	-3.6384	0.8418	0.9387
9	*MAE*	0.1644	0.0647	0.6617	0.5707	**0.0233**	0.0537
	MAPE	0.2913	0.1498	1.4597	1.2314	**0.0582**	0.1116
	MSE	0.0449	0.0059	0.5290	0.4105	**0.0007**	0.0044
	RMSE	0.2119	0.0769	0.7273	0.6407	**0.0271**	0.0664
	R^2	0.5249	0.9375	-4.5952	-3.3425	**0.9922**	0.9533
10	*MAE*	0.1814	0.0613	0.5936	0.6113	0.1492	**0.0442**
	MAPE	0.3773	0.1551	1.3012	1.3165	0.3439	**0.1107**
	MSE	0.0515	0.0047	0.4451	0.4679	0.0286	**0.0028**
	RMSE	0.2269	0.0685	0.6672	0.6841	0.1690	**0.0531**
	R^2	0.4553	0.9504	-3.7082	-3.9497	0.6978	**0.9702**

Table 5. Error Metrics. LR= Linear Regression; SVM= Linear SVM; RF = Random Forest; GB = Gradient Boosting; CNN = Convolutional Neural Network; MLP= Multilayer Perceptron.

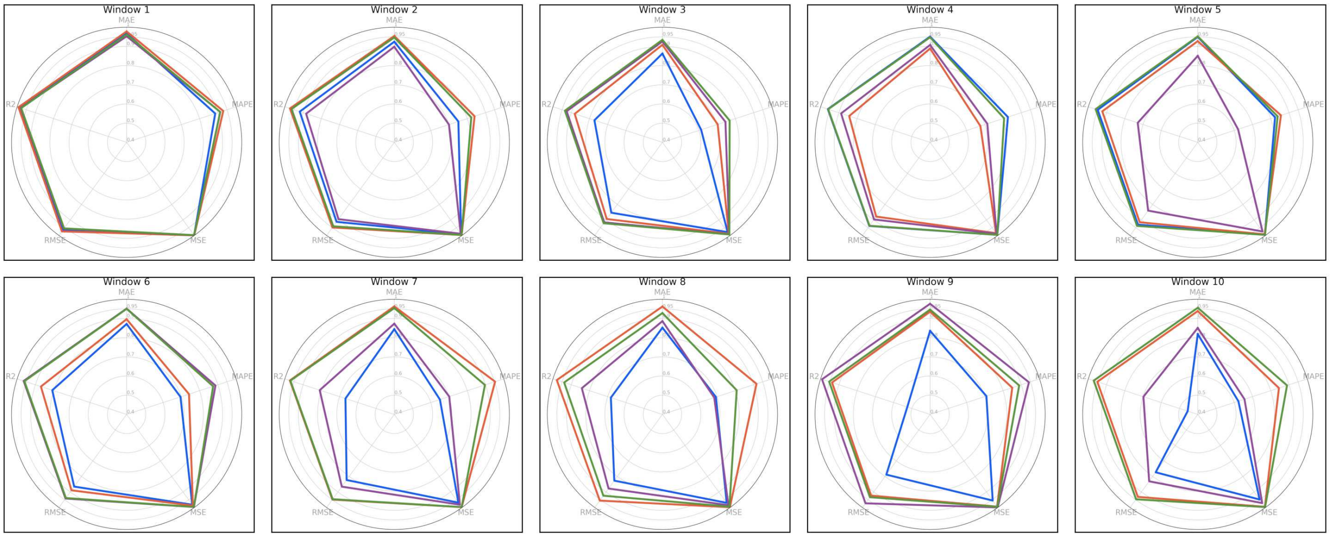

Figure 5. Radar Chart. Blue: Linear Regression; red: Linear SVM; purple: Convolutional Neural Network; green: Multilayer Perceptron.

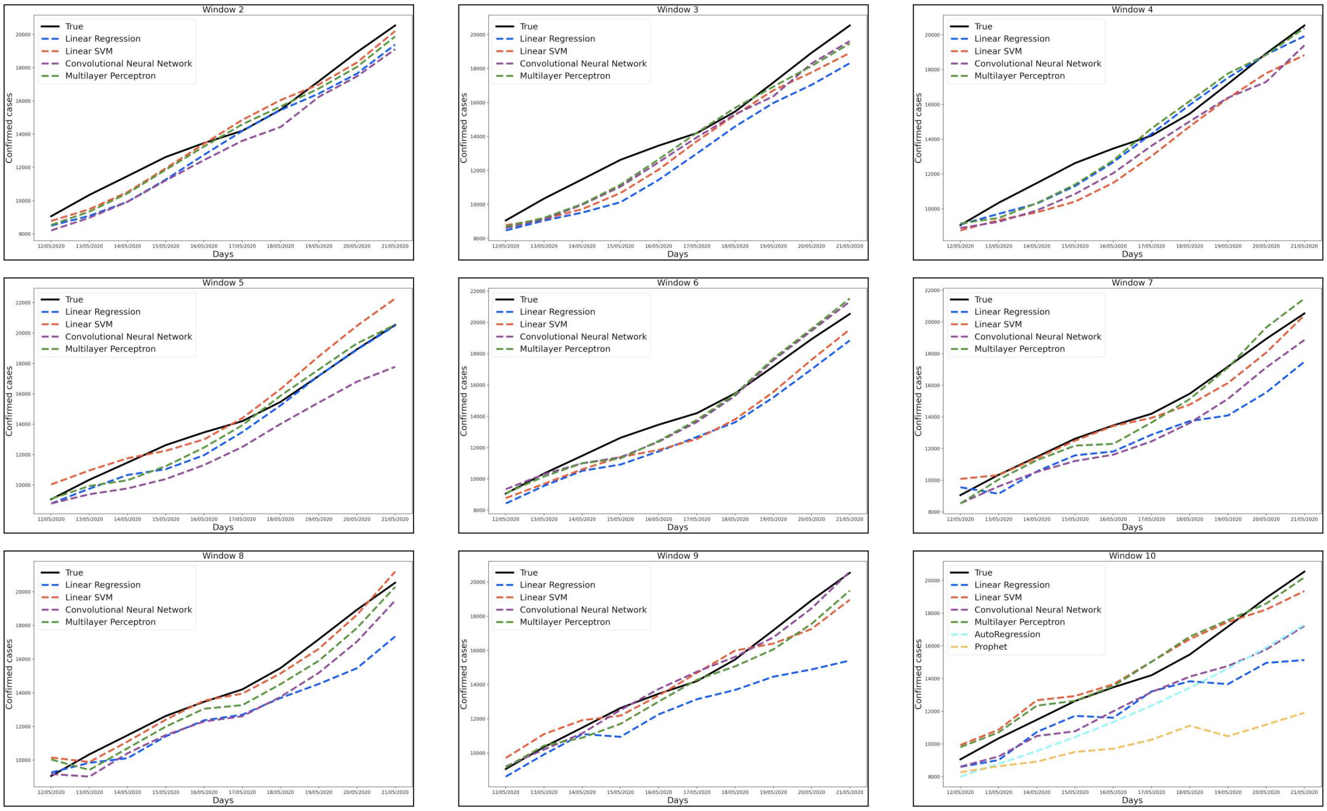

Figure 6. Predicted Values. Blue: Linear Regression; red: Linear SVM; purple: Convolutional Neural Network; green: Multilayer Perceptron; cyan: AutoRegression; orange: Prophet.

the best models. This behavior indicates a certain inconsistency in the predictability of the CNN model. When looking at the windows 5, 7, 8, and 10 in Figure 6, we can also observe this inconsistency when we notice that the CNN prediction curve deviates from the true cases throughout the time series. Although the CNN model shows good results for other prediction windows, in general, it showed inferior performance than the Linear SVM and Multilayer Perceptron models. This reinforces our assumption that depending on the problem and the data type, other machine learning models may perform better than the ensemble and deep learning models.

In this experiment, the Linear SVM and Multilayer Perceptron presented the best overall performances. In Figure 5, we can see that the Linear SVM (red) model showed some performance variations in the smaller prediction windows. However, in the larger prediction windows, this model was similar to the performance of the Multilayer Perceptron (green). In Figure 6, when we analyze the fit of the predictions about the true cases, we find that the Linear SVM and Multilayer Perceptron predictions are well adjusted in almost all prediction windows. Besides, for windows ≥ 7, these models have the best predictions for the longest horizon (21/05/200). The only exception is window 8, where the CNN model has the best prediction for the longest horizon. In the real scenario, obtaining the best prediction for the longest horizon is ideal because this is the information used for decision making by the authorities. This assumption also reinforces our analysis that points out the Linear SVM and Multilayer Perceptron as the best models evaluated in our experiment.

In window 10 of Figure 6, we also compare the performance of the AutoRegressive and Prophet[15] statistical models. As can be seen in this figure and Table 6, these models did not achieve the desired generalization in our experiment. This result suggests that statistical models are not always sufficient to solve prediction problems like the one presented in this work. Other statistical models, such as Autoregressive Integrated Moving Average (ARIMA), Seasonal Autoregressive Integrated Moving-Average (SARIMA), Seasonal Autoregressive Integrated Moving-Average with Exogenous Regressors (SARIMAX), Simple Exponential Smoothing (SES), and Holt Winter's Exponential Smoothing (HWES) could be tested to try to resolve the prediction proposed in this work, however, testing all these models is beyond the scope proposed in this research.

Model	MAE	MAPE	MSE	RMSE	R^2
AutoRegressive	0.1881	0.4685	0.0383	0.1957	0.5948
Prophet	0.3771	0.7911	0.1880	0.4336	-0.9887

Table 6. Error metrics for the AutoRegressive and Prophet models.

Checking the models' equivalence

Following the methodology presented in Figure 4, the last step of our experiment is to conduct a hypothesis test to verify the equivalence between the models. In this case, we have verified the equivalence between the two best evaluated models. The null hypothesis (H0) of our test considers that the Linear SVM and Multilayer Perceptron models are statistically equivalent and the alternative hypothesis (H1) that the models are different. The Student's t-test was employed to determine the significance of these differences. If we get a p-value ≤ 0.05, it means that we can reject the null hypothesis and the machine-learning models are significantly different with 95% confidence. To perform this test, we use window 10 to predict the cases from May 22, 2020, to May 31, 2020. Until the submission of this article, this prediction period had not been reported in the official panel yet. So, we want to check the equivalence of the best models for future cases. Figure 7 and 8 show the estimated prediction density of these two models and the prediction curve for the ten-day horizon. Looking at these two plots, we can see the similarity between these two models. When applying the Student's t-test we get a p-value $= 0.0.438$ indicating that we must accept the H0 hypothesis because the models are statistically equivalent. This result confirms the analysis carried out in the previous one in which we selected these two models as the best.

Conclusion

COVID-19's problems are growing every day, especially in the poorest countries. In Brazil, the number of infected people is growing rapidly and health, economic and social problems are becoming more difficult in almost all states. In the state of Pará, where we choose to conduct this research, the government has declared a lockdown since May 7, 2020. In an attempt to slow Covid-19's progress, the lockdown suspends all non-essential activities and prohibits the movement of people through the streets in the capital Belém and other 9 cities in the state. To contain the progress of this epidemic in the state, it is important to have access to information from a predictability perspective. In this sense, considering the official data reported by the Pará Department of Health, we have built a data set to analyze the performance of various models of machine learning in the task of predicting the number of infected due to COVID-19. Based on the prediction of the number of infected, we believe that the authorities can use this information to make the best decisions.

In our experiment, we analyzed the performance of six machine learning models and two statistical models. The models analyzed were: Linear Regression, Linear SVM, Random Forest, Gradient Boosting, Convolutional Neural Network, Multilayer Perceptron, AutoRegressive, and Prophet. We chose these models because we had the hypothesis that although ensemble and deep learning models are used to solve almost any problem where machine learning models can be applied, there are exceptions, depending on the nature of the problem and the data type. We analyze the prediction performance of these models taking into

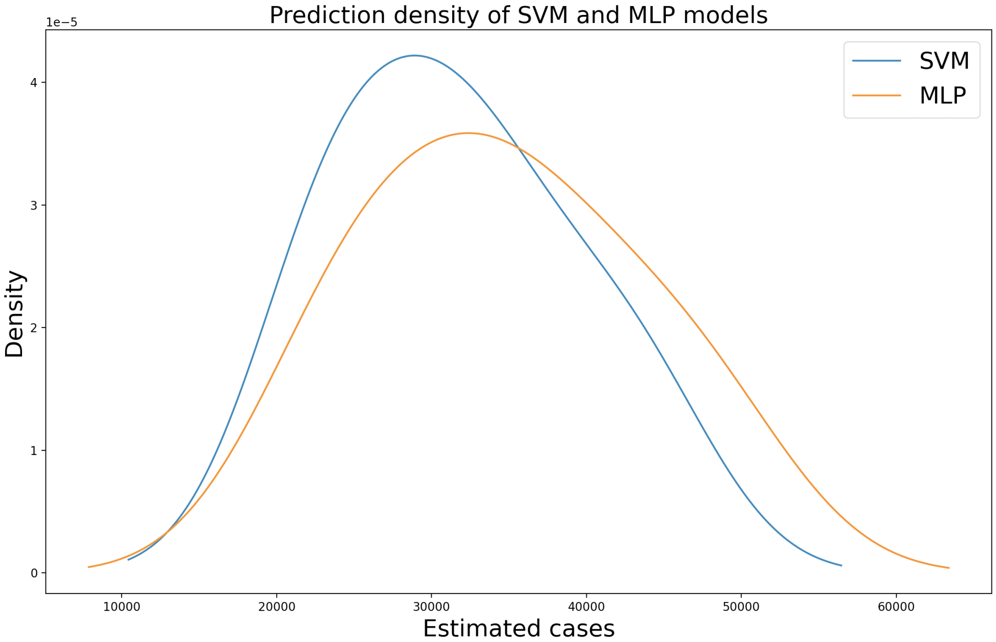

Figure 7. Kernel density of Linear SVM and Multilayer Perceptron models.

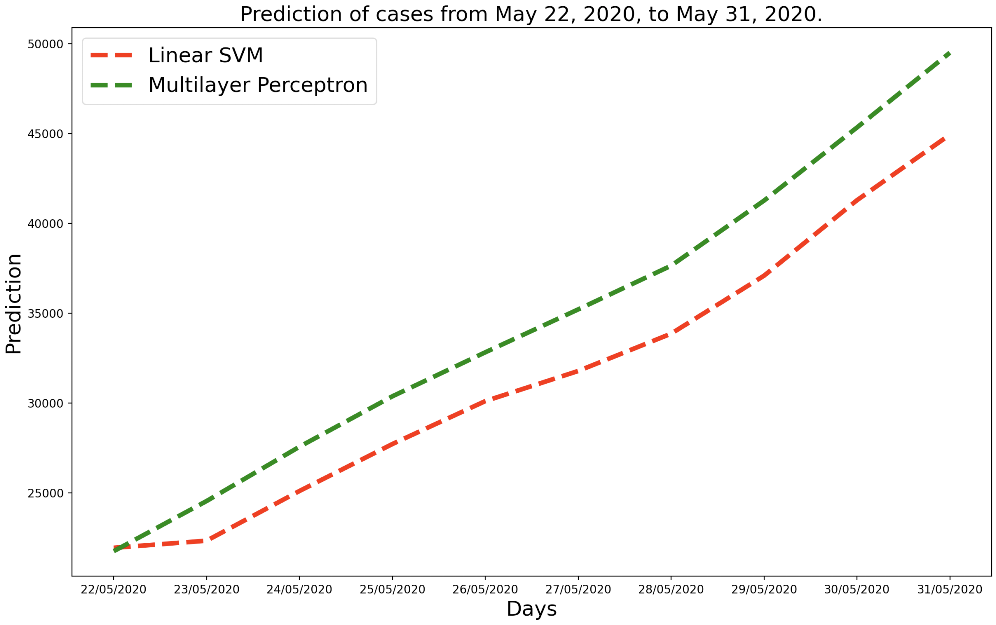

Figure 8. Prediction of cases from May 22, 2020, to May 31, 2020.

account 10 prediction windows. This analysis was performed to verify if there was a change in the models' performance due to the prediction horizon. At the end of our analysis, we applied the Occam's razor principle which, in the context of machine learning, says that when faced with two classified algorithms with the same training performance and testing capacity, the simplest model will probably be the best choice.

Returning to our problem formulation, we evaluate different prediction windows $\Omega = \{1, 2, ..., 10\}$ and the generalization capacity of machine learning models $\Psi = \{$Linear Regression, Linear SVM, Rnadom Foresr, Gradient Boosting, CNN, Multilayer Perceptron$\}$, in the task of estimating the number of infected due to COVID-19 in the state of Pará Φ, i.e., we evaluate how Φ can be solved by $\Psi(\Omega)$ and also which the influence of $\omega \in \Omega \; \forall \; \psi \in \Psi$. From this formulation, we can conclude that the Ω set influenced the performance of the Linear Regression and Convolutional Neural Network models. The performance of Linear Regression was inversely proportional to the values of the Ω set, i.e., the larger the prediction window, the lower the performance of Linear Regression. CNN, on the other hand, presented performance variation for some values of the Ω set. his performance fluctuation demonstrates the unpredictability of the model and, for this reason, CNN cannot be compared to the best prediction models analyzed in this work.

In our analysis, we also saw that Φ cannot be solved by $\Psi^*(\Omega)$ where $\Psi^* = \{$ Random Forest, Gradient Boosting$\}$. These two models failed to generalize the problem with all the prediction windows. As previously explained, the poor performance of these models can be explained due to the way the decision tree algorithms work. Finally, the two best models analyzed in this experiment were Linear SVM and Multilayer Perceptron. The MultiLayer Perceptron showed excellent performance for all the predicted windows analyzed. The SVM model, on the other hand, showed little variation in the intermediate windows but showed excellent performance in the largest windows, which are the most important. Considering the performance equivalence of these models for the largest prediction windows, we applied Student's t-test to verify the statistical significance between these two models. The result of the hypothesis test showed that the Linear SVM and Multilayer Perceptron models are equivalent. Given the equivalence between the models, we applied the principle of Occam's razor and concluded that the Linear SVM model is the most recommended to perform the prediction of infected due to COVID-19 in the state of Pará.

References

1. WHO. Coronavirus disease 2019 (covid-19) situation report - 122. Tech. Rep., World Health Organization (2020).
2. WHO. Novel coronavirus (2019-ncov) situation report - 1. Tech. Rep., World Health Organization (2020).
3. Ardabili, S. F. *et al.* Covid-19 outbreak prediction with machine learning. *medRxiv* DOI: 10.1101/2020.04.17.20070094 (2020).
4. Huang, C.-J., Chen, Y.-H., Ma, Y. & Kuo, P.-H. Multiple-input deep convolutional neural network model for covid-19 forecasting in china. *medRxiv* DOI: 10.1101/2020.03.23.20041608 (2020).
5. Al-qaness, M. A., Ewees, A. A., Fan, H. & Abd El Aziz, M. Optimization method for forecasting confirmed cases of covid-19 in china. *J. Clin. Medicine* **3**, DOI: 10.3390/jcm9030674 (2020).
6. Ceylan, Z. Estimation of covid-19 prevalence in italy, spain, and france. *J. Clin. Medicine* **729**, DOI: 10.1016/j.scitotenv.2020.138817 (2020).
7. Bandyopadhyay, S. K. & Dutta, S. Machine learning approach for confirmation of covid-19 cases: Positive, negative, death and release. *medRxiv* DOI: 10.1101/2020.03.25.20043505 (2020).
8. IBGE. Pesquisa nacional por amostra de domicílios: Síntese de indicadores. Tech. Rep., Instituto Brasileiro de Geografia e Estatística (2016).
9. Covid-19 cases are reported by the public health department of the state of pará. https://www.covid-19.pa.gov.br/. Accessed: 2020-05-21.
10. Rasmussen, C. E. & Ghahramani, Z. Occam's razor. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, 276–282 (MIT Press, Cambridge, MA, USA, 2000).
11. Shalev-Shwartz, S. & Ben-David, S. *Understanding Machine Learning: From Theory to Algorithms* (Cambridge University Press, USA, 2014).
12. Alimadadi, A. *et al.* Artificial intelligence and machine learning to fight covid-19. *Physiol. Genomics* **52**, 200–202, DOI: 10.1152/physiolgenomics.00029.2020 (2020). PMID: 32216577.
13. Bullock, J., Luccioni, A., Pham, K. H., Lam, C. S. N. & Luengo-Oroz, M. Mapping the landscape of artificial intelligence applications against covid-19 (2020). 2003.11336.
14. Ming, W.-K., Huang, J. & Zhang, C. J. P. Breaking down of the healthcare system: Mathematical modelling for controlling the novel coronavirus (2019-ncov) outbreak in wuhan, china. *bioRxiv* DOI: 10.1101/2020.01.27.922443 (2020).
15. Taylor, S. J. & Letham, B. Forecasting at scale. *PeerJ Prepr. 5:e3190v2* DOI: 10.7287/peerj.preprints.3190v2 (2017).
16. Hutter, F., Kotthoff, L. & Vanschoren, J. *Automated Machine Learning: Methods, Systems, Challenges* (Springer International Publishing, 2019), 1 edn.
17. Hyndman, R. & Athanasopoulos, G. *Forecasting: Principles and Practice* (OTexts, Australia, 2018), 2nd edn.
18. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. Rethinking the inception architecture for computer vision. *Proc. IEEE Conf. on Comput. Vis. Pattern Recognit.* (2016).

Declarations:

Competing interests: The authors declare no competing interests.

Figures

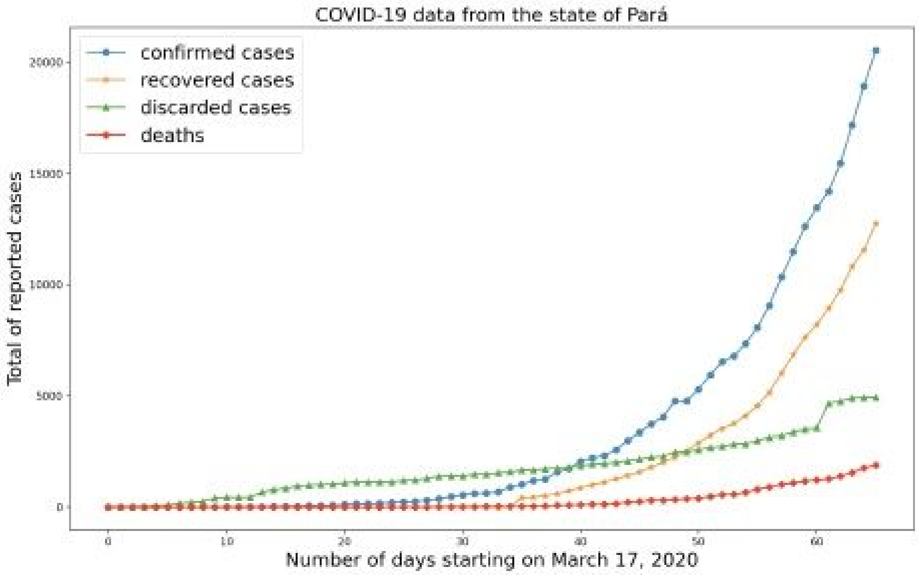

Figure 1

Official numbers of cases reported by the public health department of the state of Pará.

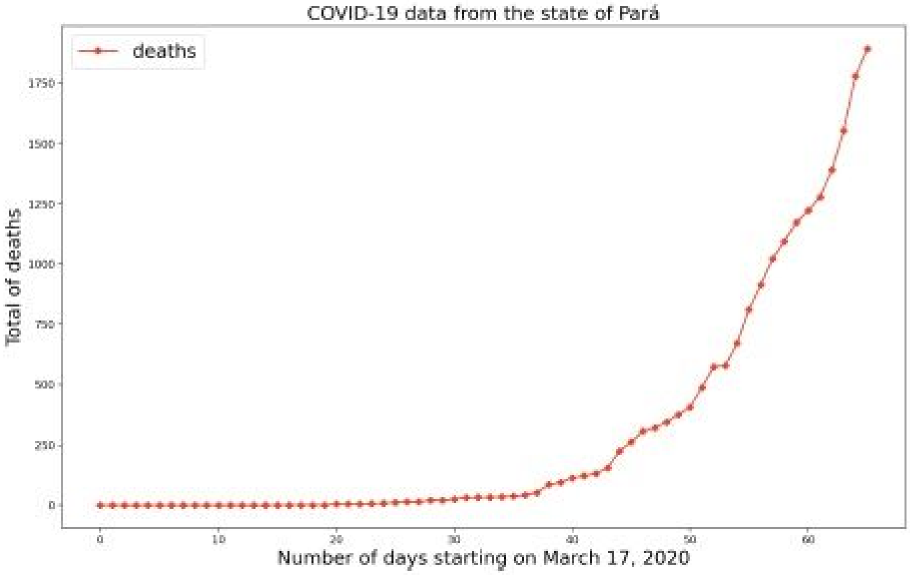

Figure 2

Official numbers of deaths reported by the public health department of the state of Pará.

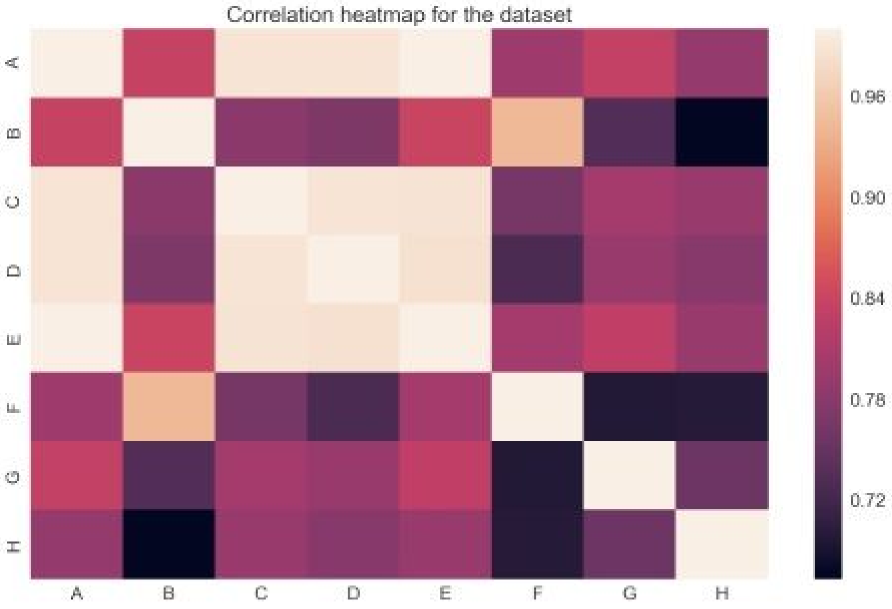

Figure 3

Correlation Heatmap. A=Confirmed cases; B=Discarded cases; C=Deaths; D=Recovered cases; E=Confirmed/100k inhabitants; F= Deaths/confirmed; G= New cases in one day; H= New deaths in one day.

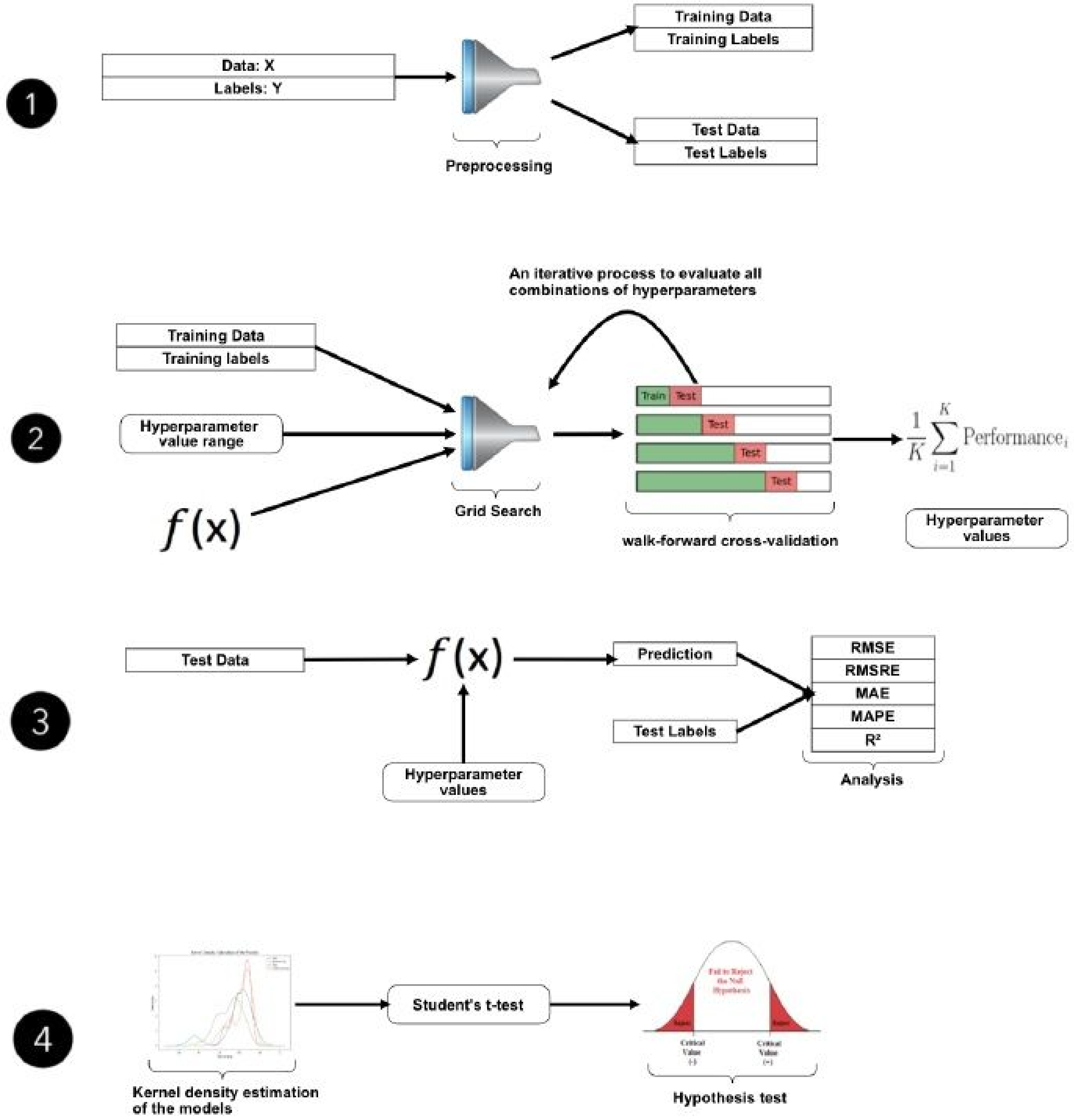

Figure 4

Methodology to assess the performance of the models. Step 1: split the data set into training and testing subsets. Step 2: application of grid search and cross-validation to train and select the hyperparameters of the models. Step 3: evaluation of the models with the test subset. Step 4: hypothesis test to verify the statistical significance of the best models evaluated.

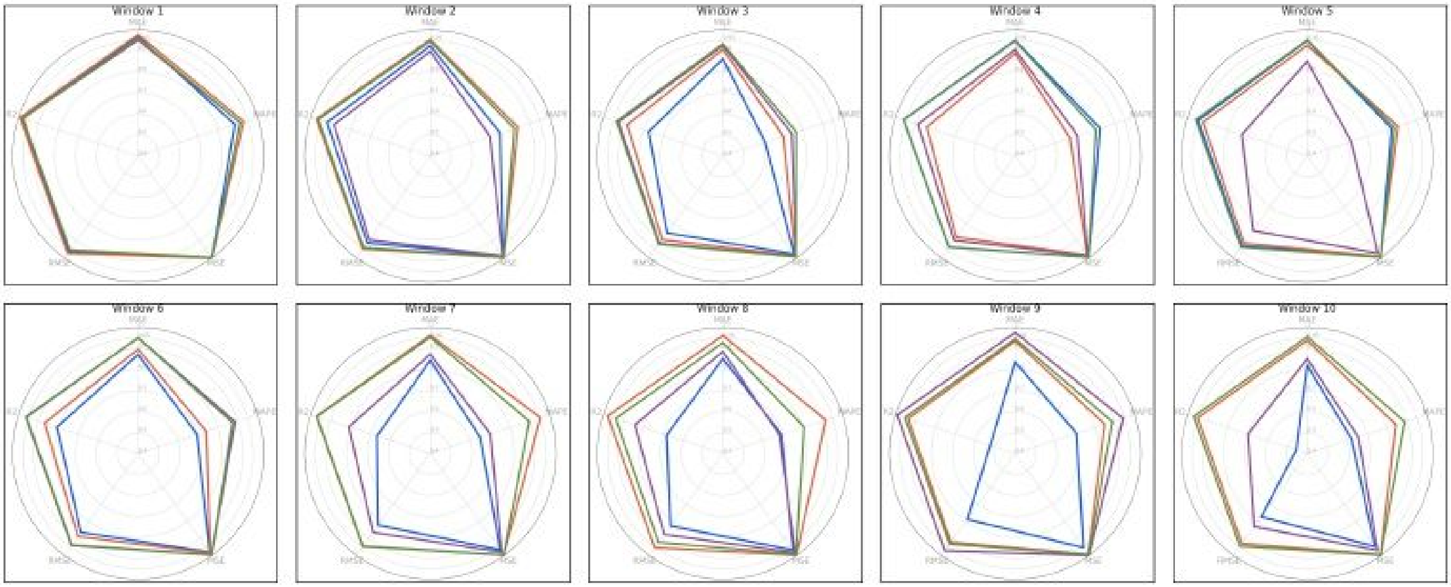

Figure 5

Radar Chart. Blue: Linear Regression; red: Linear SVM; purple: Convolutional Neural Network; green: Multilayer Perceptron.

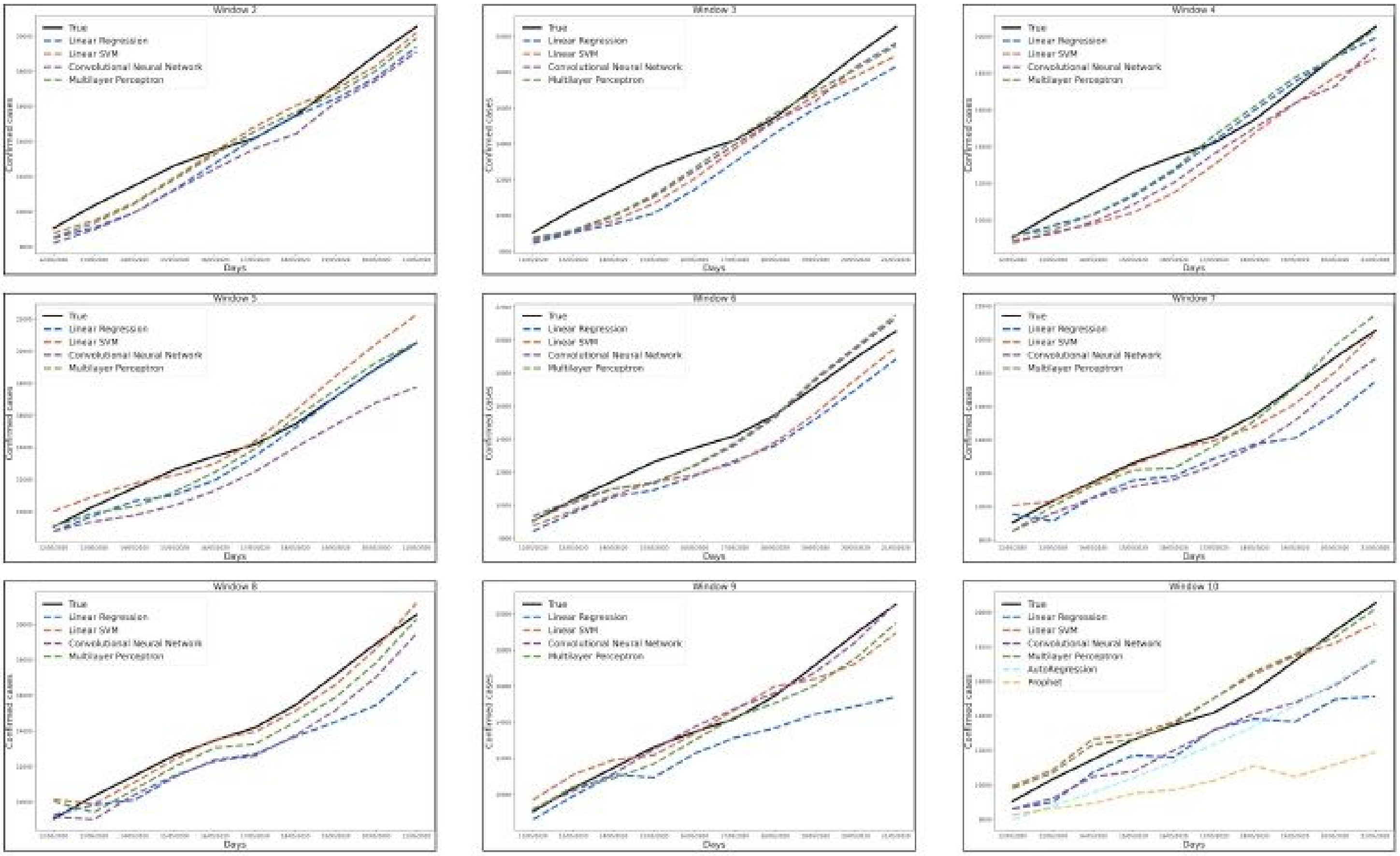

Figure 6

Predicted Values. Blue: Linear Regression; red: Linear SVM; purple: Convolutional Neural Network; green: Multilayer Perceptron; cyan: AutoRegression; orange: Prophet.

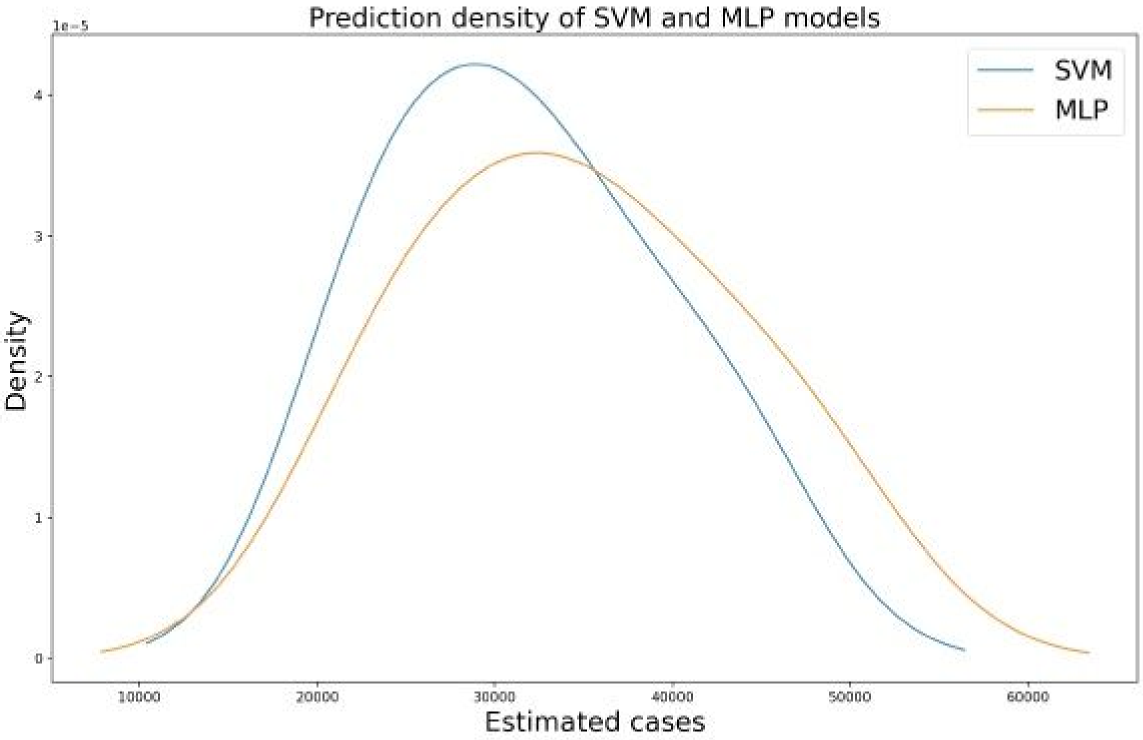

Figure 7

Kernel density of Linear SVM and Multilayer Perceptron models.

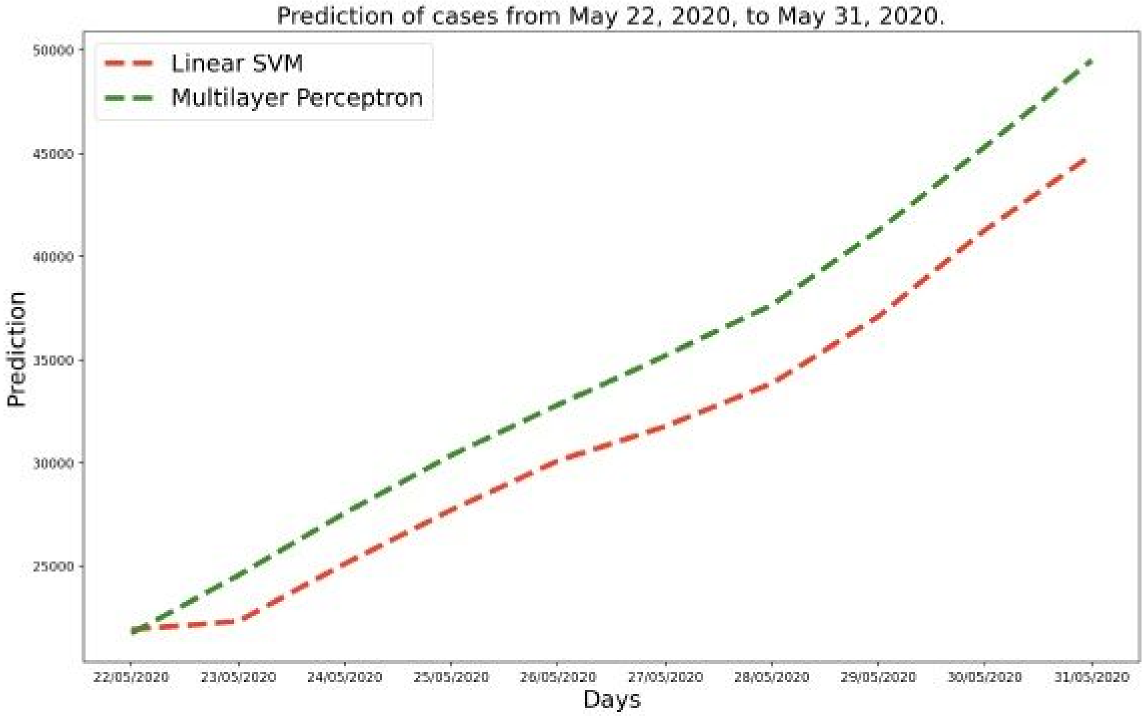

Figure 8

Prediction of cases from May 22, 2020, to May 31, 2020.