NTNU
Kunnskap for en bedre verden

Department of Computer Science

TDT4173 - Assignment 3

# Forecasting COVID-19 With Machine Learning

*Group:*
Time Series Group 13

*Authors:*
Borger Christopher Melsom (borgercm)
An Thi Nguyen (antn)
Oskar Størmer (oskarsto)

November 26, 2020

## Abstract

The COVID-19 pandemic is an ongoing, global catastrophe that has caused more than 1 million deaths worldwide. From the virus' discovery late in 2019, experts all over the world have made efforts to predict the spread of the virus. Improving the accuracy of these predictions can assist governments in implementing preventive measures at the right time.

Although epidemiological models have been the most prominent in the public sphere, machine learning researchers have also attempted to build models to forecast the spread of COVID-19. In this project we explore the viability of machine learning for COVID-19 forecasting. This is done by creating two different machine learning models and comparing them to a statistical benchmark. The models created are Long short-term memory (LSTM), Random forest (RF), and Holt-Winters' Additive Dampened (H-W($A_d, A$)). The forecasting error of the two machine learning models exceeds the error of the benchmark model when measured in RMSE. Our results indicate potential areas for improvement of machine learning models applied to this task. Given a larger dataset, sophisticated clustering strategies for the time series, and incorporation of the right regressors, we believe that leveraging machine learning can help us better forecast the development of COVID-19 and other pandemics in the future.

The code implementation can be found on Github:
`https://github.com/ostormer/covid_ml`.

A podcast where we discuss our findings and the lessons learned can be found here:
`https://github.com/ostormer/covid_ml/blob/main/podcast_report/ML-Podcast.m4a`

# Table of Contents

# List of Figures

# List of Tables

# 1 Introduction

The COVID-19 pandemic is an ongoing, global catastrophe that has caused more than 1 million deaths worldwide.[1] The virus has spread quickly since it was first discovered in China in late 2019. Since then, experts all over the world have made efforts to predict the spread of the virus in order to provide a factual basis for the decision-makers in society. Improving the forecasting accuracy can help governments implement preventive measures at the right time, potentially saving lives and reducing the pandemic's impact on the economy.

Several researchers have attempted to leverage machine learning in order to forecast the spread of COVID-19. In Ribeiro et al. [2020], different multi-step-ahead forecasting models were employed for the task of forecasting the total confirmed cases in 10 Brazilian states with a horizon of 1, 3 and 6 days. The best models performed well with out-of-sample errors below 6.9%. However, the results may have been limited by the small dataset as the testing set consisted of only the six last observations. In Chimmula and Zhang [2020], Long short-term memory (LSTM) was employed to forecast the total cases of COVID-19 in Canada. The model predicted that the outbreak in Canada would end around June 2020, which turned out false as Canada experienced a second wave around October.[2] This shows the importance of continually assessing new data as the pandemic progresses, as models can quickly become outdated.

Other studies have attempted to predict the course of the pandemic in multiple countries. In Chakraborty and Ghosh [2020] Autoregressive Integrated Moving Average (ARIMA) and wavelet-based forecasting were combined to forecast COVID-19 cases with a ten-day horizon for India, United Kingdom, Canada, South Korea and France. In Zeroual et al. [2020] different deep learning methods were applied to predict new and recovered cases 17 days ahead in France, Italy, Spain, Australia, China, and the U.S.

There were some limitations with the studies mentioned above. Firstly, they had a limited amount of available data. Many of the studies were conducted in the early months of the pandemic, making the number of data points per country relatively small. The size of the datasets was further reduced as the research only looked at a few countries. The accuracy of machine learning models is dependent on the quality and the size of the dataset they use. Thus, for previous research the small amount of available data limited the quality of the models and the conclusions that could be drawn from them. Furthermore, there is a need to assess the suitability of machine learning models for forecasting at the current stage of the pandemic. The importance of different predictors may vary throughout the course of the pandemic, making it valuable to revisit this problem.

The purpose of this paper is to explore the viability of machine learning models for predicting the course of the COVID-19 pandemic in European countries. We apply different existing methods to an ongoing problem and investigate the behavior of the models we build. We utilize a dataset containing a variety of variables, which includes the number of daily confirmed cases of COVID-19. Data from 32 European countries, recorded between December 31st 2019 and November 7th 2020, is used to forecast new cases with a horizon of 7 days. We compare two machine learning models from two different classes of methods: Random forest (ensemble learning), and LSTM (deep learning). The statistical method Holt-Winters' Additive Dampened (H-W($A_d, A$)) is used as a baseline for comparison. Our approach differs from the previous research in the sense that: (i) we can use more data at this later stage of the pandemic as the number of observations has grown. (ii) the accuracy of the data is better as the testing volume has increased manyfold since the spring[3], and (iii) we have used 32 countries in our models, and thus trained them on a larger variety of data.

[1] WHO, "WHO Coronavirus Disease (COVID-19) Dashboard", *World Health Organization* https://covid19.who.int/, (accessed 20 November 2020)

[2] CNBC, "WHO warns Canada is facing a 'second wave' of coronavirus cases", *CNBC* https://www.cnbc.com/2020/10/14/who-warns-canada-is-facing-a-second-wave-of-coronavirus-cases-.html, (accessed 18 November 2020)

[3] ECDC, "Data on testing for COVID-19 by week and country", *European Centre for Disease Prevention and Control* https://www.ecdc.europa.eu/en/publications-data/covid-19-testing, (accessed 20 September 2020)

# 2 Data

This project utilizes the COVID-19 dataset by Our World in Data (OWiD)[4]. The dataset is a collection of all the COVID-19 related data OWiD has collected and is updated daily. OWiD's dataset is structured as a single table where each row is a report from one country for one date. The table has 50 columns, so a careful examination of the data was required when choosing what features to use for the methods.

## 2.1 Summary and Visualization

Before choosing which features to use for this project we first took into account which methods would be used. The objective of this project was to predict the number of new cases per one million population with a horizon of 7 days, so the column containing "daily new confirmed cases per one million population" was included. Our baseline statistical method, Holt-Winters' Additive Dampened, can only use one variable and thus only works on a univariate time series. Random forest and LSTM are machine learning models and are well suited to work with multivariate input data. Many of the features, such as population, median age, etc, were not updated daily and had the same values for all days for a given country. For time series forecasting, features that change from day to day are naturally more interesting than values that keep the same value for the entire series. Yet some of the features that keep the same value within a single country's data can still be useful for the model to learn relations between patterns of different countries. The group decided to use the features shown in Table 1 when tuning the models.

| Name | Description | Updated Daily | Source |
|---|---|---|---|
| `date` | Date of observations | Yes | Our World in Data |
| `iso_code` | ISO 3166 Alpha-3 code | No | International Organization for Standardization |
| `location` | Name of country | No | Our World in Data |
| `new_cases _per_million` | New confirmed cases of COVID-19 per million population | Yes | European Centre for Disease Prevention and Control |
| `new_cases_smoothed _per_million` | New confirmed cases of COVID-19 per million population (7-day smoothed) | Yes | European Centre for Disease Prevention and Control |
| `total_tests` | Total tests for COVID-19 | Yes | National government reports |
| `new_tests` | New tests for COVID-19 | Yes | National government reports |
| `new_tests_smoothed` | New tests for COVID-19 (7-day smoothed) | Yes | National government reports |
| `stringency_index` | Government Response Stringency Index: composite measure based on 18 response indicators | Yes | Oxford COVID-19 Government Response Tracker, Blavatnik School of Government |
| `latitude` | Country latitude | No | Our World in Data |
| `longitude` | Country longitude | No | Our World in Data |

Table 1: Variables kept in the dataset after preprocessing

To minimize the effect of outliers on the models, some European countries were removed from the dataset. These were all countries with population less than 600 000, countries which had negative values for `new_cases` in the time series, or countries missing all or a significant number of observations for some of the variables in Table 1. After removing countries with problematic data we were left with data on 32 European countries.

Figure 1 shows daily new cases of COVID-19 per 1 million inhabitants for four sample countries: Norway, Sweden, Finland and Belgium. This sample group was selected because these countries

[4] Our World in Data, "Coronavirus Pandemic (COVID-19)", *Our World in Data*, https://ourworldindata.org/coronavirus, (accessed on 2020-11-22)

represent a variety of challenges for our models as the countries have experienced different trajectories of the pandemic.

The number of cases per one million population shown in Figure 1 is a moving average of the 7 last days. We use this 7-days smoothed value instead of the actual daily reported cases because many countries have large fluctuations within each week due to the reporting policy depending on weekdays. As an example, many countries do not report any test results during weekends. This does not mean there are no new cases of COVID-19, just no new *reported* cases. As the objective of interest here is to know how many cases there will be a week into the future, correctly predicting these daily fluctuations is not a necessity. Figure 1 shows the first wave in March and April, and the second wave that is currently ongoing in most European countries as of November 2020.

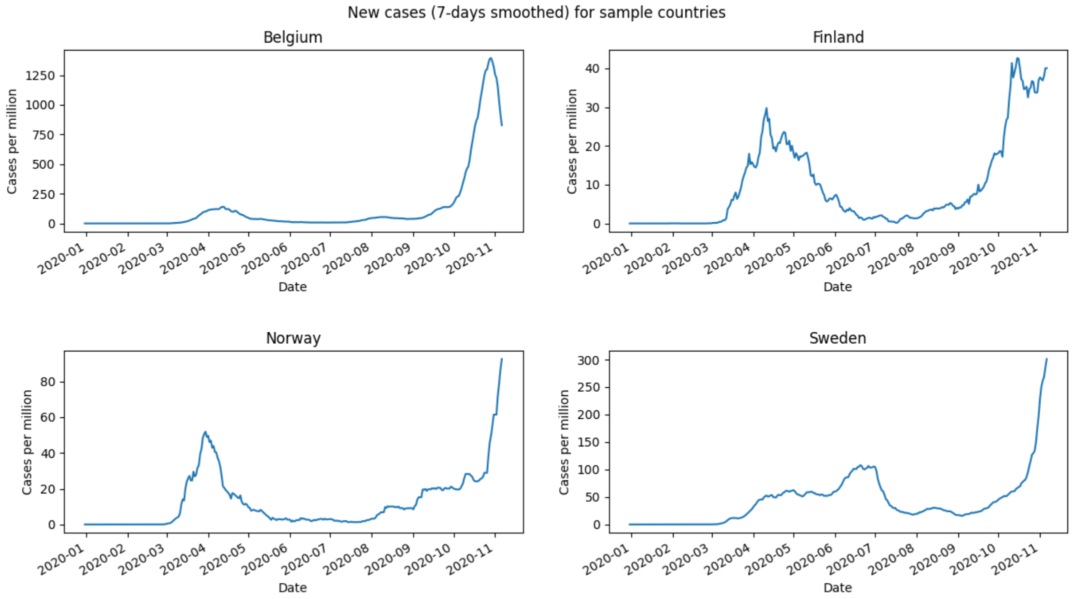

Figure 1: Data of new COVID-19 cases per one million population (7-days smoothed) in the sample countries Norway, Sweden, Finland and Belgium.

The disadvantage of using 7-days smoothed values is that it provides a delayed description of the situation. Assuming a steady trend, the value conveyed by the average value is effectively "lagged" by 3.5 days, as every value is the average of the past 7 days. It is beneficial for models to be able to handle raw data to avoid this lag.

## 2.2 Preprocessing

The dataset with the chosen countries and variables can be viewed as multiple time series, one for each country. Many countries have reported COVID-19-related data since December 31st 2019, while some only started reporting after the first confirmed case in the country. To simplify the training process, all these individual time series were "padded" so they started at the same date, December 31st 2019. For this padding it was assumed that no tests were done, no new cases confirmed and no countermeasures against the imminent pandemic were introduced.

Some countries were completely missing data for some dates. For these dates and countries new rows were inserted into the table, assuming no new tests were conducted or cases confirmed. Otherwise these new rows were a copy of the preceding date. Any additional missing values after this were filled by interpolation of near values.

### 2.3 Training, Test and Validation Data Split

As this is a time series forecasting problem with a forecast horizon of 7 days, splitting the data set into a training and test set by randomly picking observations does not work. This left the group with two options: (i) split the data by country, or (ii) do a chronological split. The first was not an option as our benchmark model, H-W($A_d, A$), fits a separate model for each country. H-W($A_d, A$) trains its parameters by optimizing loss on past data for each country separately. Therefore, every country needed to be represented in both the training and the test set. Thus, the dataset was split chronologically so that all data up to and including October 31st 2020 was used as the training set, while the first week of November 2020 was used as the test set.

One concern with this split is that the size of our test set is too small. The reasoning behind using only the first week of November as a test set is that recent parts of the time series have a high importance for both training and testing. Testing practices and other factors that influence the spread of COVID-19 have changed over the course of the pandemic. As a result of this it is important that the model may both learn from, and test its performance on recent data.

To facilitate tuning of the models, a validation set was split from the training set. This split was done by countries, with 75 % of countries being declared training countries and 25 % validation countries. This made it possible to tune a model by looking at the accuracy on the validation set, and not risking overfitting the test set in the tuning process.

# 3 Methods

## 3.1 Holt-Winters' Additive Seasonal, Additive Damped Trend

Holt-Winters' Additive Seasonal, Additive Damped Trend, hereafter H-W($A_d, A$), is an en exponential smoothing (ES) method for time series forecasting. It is based on weighted averages of previous observations. ES methods and the foundations of H-W($A_d, A$) were thoroughly explained in our Method Paper [Melsom et al., 2020], and the "Foundations" chapter is reproduced in Appendix A. We refer to this as the basis for the following discussion.

### 3.1.1 Foundation

H-W($A_d, A$) creates a forecast that is built up by three components; Level, Trend and Season. It calculates the forecast based on Equation 1:

$$\begin{aligned}
\text{Forecast Equation: } & \hat{y}_{t+h} = l_t + \phi_h b_t + s_{t+h-m} \\
\text{Level Equation: } & l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + \phi b_{t-1}) \\
\text{Trend Equation: } & b_t = \beta(l_t - l_{t-1}) + (1-\beta)\phi b_{t-1} \\
\text{Seasonal Equation: } & s_t = \gamma(y_t - l_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned} \tag{1}$$

where $\hat{y}_{t+h}$ is the forecasted number of new COVID-19 cases per million inhabitants at time $h$ days after the last data point. $m$ is the frequency of seasonality, $\alpha$ is a smoothing parameter with $0 \leq \alpha \leq 1$ and $\beta$ is a trend specific smoothing parameter with $0 \leq \alpha \leq 1$. The closer $\alpha$ and $\beta$ are to 1, the more weight is put on the most recent observations. $\phi$, with $0 \leq \phi \leq 1$, is a dampening factor to flatten the trend over time, as methods with constant trend have a tendency to overestimate the trend in the long run.

### 3.1.2 Relevance to the Problem

ES has been one of the leading time series forecasting models since the 1960s [De Gooijer and Hyndman, 2006], and the fourth Makridakis competition (M4) in 2018 showed that it is still highly relevant today [Makridakis et al., 2018b]. With a dataset consisting of few input variables and the relatively basic task of forecasting the continuation of an existing time series, ES is well suited as the baseline model for comparison.

The Holt-Winters' methods adjust for both seasonal- and trend variations when calculating the forecast. The seasonality in the COVID-19 dataset stems from different reporting policies in some countries, producing a seasonality frequency of 7 days. When seasonality fluctuates proportionately to the level of the time series, methods with multiplicative seasonality are preferred. However, as the dataset contains data points of value zero, only additive seasonality is applicable. Thus, of the more advanced exponential smoothing methods, Holt-Winters' additive seasonal, additive damped trend method was best suited for this dataset.

## 3.2 LSTM

Long Short-Term Memory (LSTM) models are a class of recurrent neural networks (RNN) that specializes on sequences of data. The method was proposed by Hochreiter and Schmidhuber [1997]. In recent years it has seen widespread successful use on problems relating to natural language processing such as translation, text classification and speech recognition [Wu et al., 2016]. LSTM networks have also seen much use on time series with success, as seen in Wang et al. [2018], Bao et al. [2017] and Hu and Chen [2018]. LSTMs have especially excelled on multivariate forecasting problems where univariate statistical methods are not applicable.

### 3.2.1 Foundation

An extensive explanation of LSTM is beyond the scope of this paper. Instead, a brief description is provided. The purpose of LSTMs is to solve the problem of RNNs only having short term memory. This problem arises when an RNN updates its weights during back propagation using gradients. The gradient value shrinks as it back propagates through time and as a result of this, RNNs often suffer from poor learning in the earlier layers where the gradient gets extremely small. In order to solve this problem, a way of handling long term memory must be introduced to the network.

The LSTM network has internal mechanism called gates that control the throughput of information and rate of learning. In addition to the hidden states in the network, an LSTM uses a cell state that acts as the "memory" for the network and carries relevant information throughout the processing of the sequence. By passing the cell state through the network, information from earlier time steps can still be intact when it reaches later time steps. This reduces the problem of short term memory. Information from each time step gets added to, or removed from, the cell state via gates.

The gates can learn what information is relevant to remember or forget during training. The *forget* gate decides what information is relevant from earlier time steps. The *input* gate decides what information should be added to the cell state in the current step. The *output* gate determines what the hidden state passed to the next time step should be. Through these mechanisms an LSTM can learn to discover data points important for the prediction task in the entire sequence.

### 3.2.2 Relevance to the Problem

As previously stated, LSTMs specialize on sequential data. With the goal of comparing a neural network model to ES and RF, LSTM was a natural choice for the NN model. An advantage of using LSTM is that it can handle extra regressors in addition to the time series of the variable to be predicted. It can also handle input where some variables are given as a time series while others are only provided once. In the case of COVID-19 prediction this is applied when using variables such as `date` and `iso_code`. Including all variables for each time step in the input provides no additional information, while making the input vector much larger than needed.

## 3.3 Random Forest

Random forest (RF) is an ensemble learning method which is comprised of multiple decision trees, where the final output is determined by the outputs of all the decision trees. For regression, the final output is the average of the individual trees' outputs. For classification, the final output class is the result of a majority vote of the classes output by the individual decision trees [Breiman, 2001].

### 3.3.1 Foundation

While individual decision trees often suffer from overfitting when they are deeply grown, RF mitigates overfitting by training each decision tree on different subsets of the training set [Hastie et al., 2009]. This is done using the bootstrap aggregating (bagging) method, where a random sample of the training set is selected. This is done several times, and individual decision trees are then trained on each of these samples. In addition to this bagging technique, RF also selects a random subset of the features in the dataset at each split. This contributes to reducing the correlation between individual trees as without this random feature selection, strong predictors would be selected in the top split in a large portion of the trees, often leading to overfitting [James et al., 2013].

RF has a number of different hyperparameters. These include the number of decision trees, the minimum observations that a terminal node should have (node size), and the number of candidate features in each split (mtry) [Probst et al., 2019]. In many practical applications, it is desirable to choose the smallest number of decision trees that still manages to stabilize the error rate [James et al., 2013]. The default value for node size, commonly 5 for regression and 1 for classification, is sufficient for most practical applications [Goldstein et al., 2011]. For mtry the optimal value is determined by the number of relevant predictor variables. Mtry should be small when there are many relevant variables and large when the portion of relevant variables is small [Bernard et al., 2009].

### 3.3.2 Relevance to the Problem

There are several advantages to RF which makes it suited for this task, as well as some disadvantages. Firstly, it is accurate for many practical applications. Secondly, it has fairly good performance using only the default hyperparameter values provided in packages [Probst et al., 2019]. Finally, it estimates the importance of each feature in the model, which can possibly help to determine relevant features for other machine learning models. However, RF is sometimes prone to overfitting for regression tasks with high noise. This should not be a big problem for our task when using the 7-day smoothed variables. Another disadvantage to the method is that the generated feature importances are not completely dependable when the dataset contains multiple categorical variables with different number of categories [Strobl et al., 2007]. However, this should not be a large problem for our dataset as the only categorical variable used for prediction is the ISO code.

# 4 Results

## 4.1 Experimental Setup

### 4.1.1 Holt-Winters' Additive Seasonal, Additive Damped Trend

The H-W($A_d$, $A$) model used as a base model for the research was built using functions and classes from the *Statsmodels* Python module. The model was designed to take `new_cases_smoothed_per_million` as input and forecast the development for a given time period and country. The *Statsmodels* functions applied in the model contained settings for automatic optimization of the hyperparameters.

Thus, the model calculates the optimal set of $\alpha$, $\beta$ and $\gamma$ parameters for each country. The hyperparameters used in the model for the sample countries are displayed in Table 2.

| Parameters | Norway | Sweden | Finland | Belgium |
|---|---|---|---|---|
| $\alpha$ | 0.83 | 1.00 | 0.84 | 0.80 |
| $\beta$ | 0.59 | 0.99 | 0.43 | 0.80 |
| $\phi$ | 0.87 | 0.88 | 0.80 | 0.97 |
| $\gamma$ | 0.00 | 0.00 | 0.00 | 0.20 |

Table 2: Optimized parameters for the H-W$(A_d, A)$ model per country when predicting `new_cases_smoothed_per_million` in the time period 01.11.20 - 07.11.20

### 4.1.2 LSTM

As a large variety of data was available in the dataset, the LSTM model was able to use additional regressors as input. `date`, `iso_code` and `stringency_index` were chosen as regressors to use in addition to `new_cases_smoothed_per_million`. One issue was how to encode the categorical feature `iso_code`. Some possible solutions for this problem are using integer encoding, one-hot encoding or learned embedding [Cerda and Varoquaux, 2020]. The issue was circumvented by replacing `iso_code` with `latitude` and `longitude`. The combination of latitude and longitude convey a country's geographical location and serves as a two dimensional embedding of `iso_code`. This embedding, conveying geographical proximity of countries, is relevant as virus transmission across borders largely happens with close countries [Skums P, 2020].

The dataset was further preprocessed to create input-output pairs for training the model. In each pair the output was $\texttt{new\_cases\_smoothed\_per\_million}_t$, and the input consisted of $\texttt{new\_cases\_smoothed\_per\_million}_{t-7...t-1}$ concatenated to the aforementioned additional regressors for time step $t-1$.

The LSTM network was built using the python library *Keras*[5], built on *Tensorflow* made by Google Inc. To enable tuning of the model, the training/validation set split by countries mentioned in section 2.3 was used. The tuning was done by examining the 1-step ahead MSE of the validation set and adjusting hyperparameters. After tuning, the final network had an LSTM layer with 32 nodes and a dense output layer with a single node. When more than 32 nodes were used, the validation error would increase as the model overfitted the training set. The model was fitted using the Adam optimization algorithm and mean squared error as its loss function. The batch size was set to 245, equal to the number of data points for each country, so the model would update its weights after processing all the data from one country. The model was trained in 400 epochs, which means the model passed through the entire training set 400 times. As with the number of nodes, 400 epochs were chosen because with any more, the NN would overfit the training data.

When evaluating the model, the 7-day forecast was done recursively. Each prediction was made one day ahead at a time, and the predicted values were used as input for further forecasts. If a longer forecast horizon than seven days would be used a direct multiple output forecasting model would be preferred to the recursive approach. This is because as the forecast horizon grows, the error of the recursive model accumulates and can lead to large prediction errors that the model has not been properly trained to avoid.

### 4.1.3 Random Forest

The 1-day difference between the smoothed new cases per million was used as the target variable for the RF model, as unstructured models tend to struggle when the data exhibits trend [Barker, 2020]. The inverse difference function `diff_inv` provided in the *Pmdarima* package was applied to obtain the final predictions. The models were built using the default hyperparameters provided in the *Scikit-learn RandomForestRegressor* package, except for the number of trees which was set to 1000.

[5] Keras, "Keras: the Python deep learning API", https://keras.io/, *Keras* (accessed on 21 November 2020)

We performed additional feature engineering and experimented with various combinations of features. The day and month were extracted as date features. In addition, we experimented with different numbers of lags for `new_cases`, `new_cases_smoothed`, `new_tests_smoothed`, and `new_cases_smoothed_per_million` and its 1-day difference. We also included the `stringency_index`. In addition to representing countries by latitude and longitude, target encoding [Micci-Barreca, 2001] was performed on the ISO code for each country, replacing each category with the category's mean of the target variable. The library *category_encoders*[6] was used for target encoding.

We also compared the results of employing two different methods for forecasting multiple time series: Training one model on all the time series, or training individual models on each series.

The experiments were evaluated on a validation set obtained by splitting up the training set. The validation set consisted of the last 7 observations of the training set in each country. This was done to make the validation set as representative as possible for the current stage of the pandemic. The RF models could not use the training/validation split by country made during preprocessing, since we trained separate models on each country. Analogous to LSTM, the 7-day forecast was performed recursively. The results from the iterative experimentation can be seen in Appendix B. We obtained the best results when training individual models on each series. The final models incorporated 14 lags, and included lagged features for `new_cases`, `new_cases_smoothed`, `new_cases_per_million` `new_cases_smoothed_per_million` and its 1-day difference, and `stringency_index`.

In the following section, we compare the results of LSTM, RF and H-W($A_d, A$) on the test set.

## 4.2 Comparison

When forecasting a pandemic, the consequences of big forecasting errors can be enormous. Minimizing the risk of big errors was therefore the most important criteria when selecting the best model for the project. Root mean squared error (RMSE) is a metric that heavily penalizes large forecasting errors, making it a well suited metric for comparing the models in this project.

| | H-W($A_d, A$) | LSTM | RF |
|---|---|---|---|
| Belgium | 201.36 | **82.35** | 277.84 |
| Finland | **1.04** | 2.35 | 1.10 |
| Norway | 8.50 | **3.46** | 8.26 |
| Sweden | 15.08 | 36.54 | **10.86** |
| Test set RMSE | **78.41** | 94.15 | 92.02 |

Table 3: Each of the models' RMSE of the sample countries Belgium, Finland, Norway and Sweden, and total RMSE of the entire test set.

The RMSE of each of the four sample countries, as well as the entire test set, is shown in Table 3. When looking at RMSE for the entire test set we see that none of the two machine learning models, LSTM and RF, beat the benchmark exponential smoothing model H-W($A_d, A$). Both machine learning methods use multiple input features and employ advanced strategies for their forecasts, yet the relatively simpler statistical model H-W($A_d, A$) performs better.

## 4.3 Discussion

Scientists have seen that several pandemics throughout history have occurred in waves and this is also the case with COVID-19.[7] Most of Europe is currently experiencing the second big wave of COVID-19 as the rate of new cases is growing rapidly across the continent. As most countries in the

[6] Will McGinnis, "Category Encoders — Category Encoders 2.2.2 documentation", https://contrib.scikit-learn.org/category_encoders/, *scikit-learn-contrib*, (accessed on 24 November 2020)

[7] John Hopkins Medicine, "Coronavirus Second Wave? Why Cases Increase", *John Hopkins Medicine*, https://www.hopkinsmedicine.org /health/conditions-and-diseases/coronavirus/first-and-second-waves-of-coronavirus, (accessed on 24 November 2020)

test set exhibit an upwards trend in the number of new COVID-19 cases, models like H-W($A_d, A$) that are able to accurately model the trend have an advantage.

Studies comparing machine learning models to statistical ones on univariate time series forecasting have shown that machine learning models are dominated by statistical models [Makridakis et al., 2018a]. When faced with a time series where multiple input features are available, the additional features used must make up for the accuracy lost by using a machine learning model instead of a statistical one. With the features used for training the models for this project, the results indicate that the additional regressors may not have been sufficient.

The small dataset may have been a limiting factor for the machine learning models' performance. Compared to statistical models, machine learning models often require a higher amount of data to define the manifold space [Barker, 2020]. For problems where the datasets are small, statistical models such as H-W($A_d, A$) may therefore be more suitable.

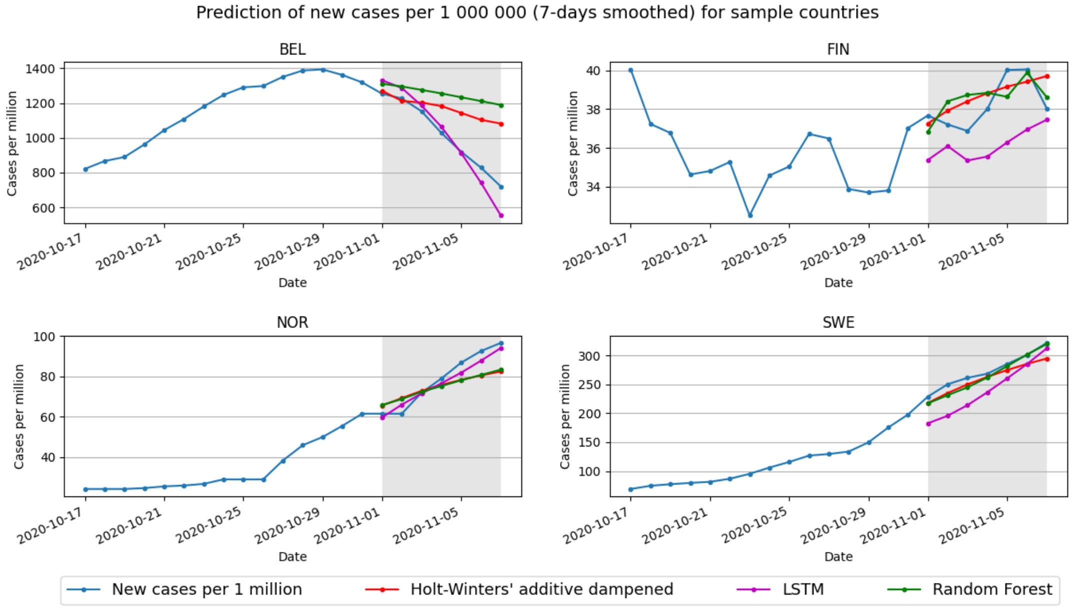

Figure 2: Test set predictions of new COVID-19 cases per one million population (7-days smoothed) in the sample countries compared to actual data.

**LSTM** As seen in Table 3, LSTM was slightly outperformed by H-W($A_d, A$). The following discussion aims to uncover the reasons for this underperformance. A recurring weakness of using Neural Network (NN) models, even convolutional ones like LSTM, for forecasting is that they tend to average their output too much [Makridakis et al., 2018a]. When we examine the graphs in Figure 2, it looks like the LSTM predicts that in countries where the number of cases is low cases will go up, while in countries with a high number of cases the number will go down. Presumably the predictions converge towards some average number of new cases per million when the forecast horizon is increased.

**Random Forest** Similarly to LSTM, RF did not manage to outperform the benchmark model, as seen in Table 3. The features with the biggest assigned importance were those related to the 1-day difference between new cases per million, as well as the actual new cases. Although the model utilized multiple variables to generate the forecasts, they were not sufficient to outperform the statistical benchmark. The Stringency Index was not as important for the model as we initially believed, as it did not contribute significantly to the model's performance. Furthermore, RF has a hard time forecasting at turning points, as illustrated by the model's predictions for Belgium in Figure 2. Here, the error in the first prediction quickly accumulates to the next recursively generated predictions, leading to a large RMSE as the true number of cases quickly drop.

# 5 Conclusion

The aim of the project was to explore the viability of machine learning models for predicting the course of the COVID-19 pandemic. Although our forecasting methods did not perform as well as we had hoped, we learned much about the strengths and weaknesses of the different models when applied to COVID-19 prediction.

In this project, we have trained machine learning models on data from European countries with the assumption that these countries are sufficiently similar to form a well-defined manifold. Future work may incorporate all countries with sufficient data and group series with similar properties, in order to create more densely populated manifolds [Barker, 2020]. One way of doing that, is clustering together series that follow the same rises and falls throughout the pandemic and train different models on each cluster. This applies in particular to RF, where we observed slightly better results by training separate models on each series, rather than training one model on all series (see Appendix B). That difference in performance indicate that the manifold where all European countries lie is not sufficiently well-defined for RF to be able to model it using the data.

Our project utilizes a bigger amount of data to train the machine learning models than the studies mentioned in Section 1. However, we still found that the size of the dataset limits the performance of our machine learning models compared to the statistical benchmark. This means that an important lesson for future work is developing more comprehensive datasets. A proactive way of improving the data would be to develop and incorporate new measures that are able to better explain the course of the pandemic.

For further work and improvement, we suggest incorporating new measures that quantify governmental response to the pandemic. Our models have used a Stringency Index based on The Oxford COVID-19 Government Response Tracker (OxCGRT). This is a combined score of 18 different factors. Clearly, the combined stringency index of OxCGRT was not sufficient to make the machine learning methods better than the statistical base model H-W$(A_d, A)$. We recommend implementing each of the 18 factors individually for possible improvement of the machine learning models. All factors could for instance be fed into a model such as RF that is able to rank feature importances.

Finally, as our results indicate, the additional regressors used in our machine learning models may not have been sufficient to outperform the statistical benchmark. Another improvement could therefore be to incorporate epidemiological domain knowledge in the models. This could for instance be done by including various epidemiological measures such as the basic reproduction number. In this way, the strengths of both machine learning and epidemiological models can be combined.

# Bibliography

Wei Bao, Jun Yue, and Yulei Rao. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PloS one*, 12(7):e0180944, 2017.

Jocelyn Barker. Machine learning in m4: What makes a good unstructured model? *International Journal of Forecasting*, 36(1):150–155, 2020.

Simon Bernard, Laurent Heutte, and Sébastien Adam. Influence of hyperparameters on random forest accuracy. In Jón Atli Benediktsson, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, pages 171–180, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-02326-2.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

P. Cerda and G. Varoquaux. Encoding high-cardinality string categorical variables. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1, 2020. doi: 10.1109/TKDE.2020.2992529.

Tanujit Chakraborty and Indrajit Ghosh. Real-time forecasts and risk assessment of novel coronavirus (covid-19) cases: A data-driven analysis. *Chaos, Solitons & Fractals*, page 109850, 2020.

Vinay Kumar Reddy Chimmula and Lei Zhang. Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos, Solitons & Fractals*, page 109864, 2020.

Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

Everette S. Gardner and Ed McKenzie. Forecasting trends in time series. 1985. doi: 10.1287/mnsc.31.10.1237.

Benjamin A Goldstein, Eric C Polley, and Farren BS Briggs. Random forests for genetic association studies. *Statistical applications in genetics and molecular biology*, 10(1), 2011.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Science & Business Media, 2009.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9: 1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5 – 10, 2004. ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2003.09.015. URL `http://www.sciencedirect.com/science/article/pii/S0169207003001134`.

Ya-Lan Hu and Liang Chen. A nonlinear hybrid wind speed forecasting model using lstm network, hysteretic elm and differential evolution algorithm. *Energy conversion and management*, 173: 123–142, 2018.

R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 3rd edition.* OTexts, 2019. URL `OTexts.com/fpp3`.

Rob Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach.* Springer Science & Business Media, 2008.

Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, 2013.

Prajakta S. Kalekar. Time series forecasting using holt-winters exponential smoothing. 2004.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, 13(3):e0194889, 2018a.

Spyros Makridakis, Evangelos Spilitos, and Vassilios Assimakopoulos. The m4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34:802–808, 2018b.

Borger Christopher Melsom, An Thi Nguyen, and Oskar Størmer. An introduction to exponential smoothing. 2020.

Daniele Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32, 2001.

Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 9(3):e1301, 2019.

Matheus Henrique Dal Molin Ribeiro, Ramon Gomes da Silva, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Short-term forecasting covid-19 cumulative confirmed cases: Perspectives for brazil. *Chaos, Solitons & Fractals*, page 109853, 2020.

Icer Baykal P Zelikovsky A Chowell G Skums P, Kirpich A. Global transmission network of sars-cov-2: from outbreak to pandemic. *Preprint. medRxiv*, 2020. doi: 10.1101/2020.03.22.20041145.

Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC bioinformatics*, 8(1): 25, 2007.

Fei Wang, Yili Yu, Zhanyao Zhang, Jie Li, Zhao Zhen, and Kangping Li. Wavelet decomposition and convolutional lstm networks based improved deep learning model for solar irradiance forecasting. *Applied Sciences*, 8(8):1286, 2018.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016.

Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, and Ying Sun. Deep learning methods for forecasting covid-19 time-series data: A comparative study. *Chaos, Solitons Fractals*, 140: 110121, 2020. ISSN 0960-0779. doi: https://doi.org/10.1016/j.chaos.2020.110121. URL `http://www.sciencedirect.com/science/article/pii/S096007792030518X`.

# Appendix

## A Exponential Smoothing from Method Paper

*The following section is the "Foundations" chapter from our Method Paper [Melsom et al., 2020]*

Exponential Smoothing is a class of forecasting methods based on weighted averages of previous observations. It is a reliable and uncomplicated method that is particularly suitable for time series data [Hyndman and Athanasopoulos, 2019].

ES methods use a linear combination of previous observations to forecast a future value. The value of each observation is multiplied by a weight. These weights are decreasing exponentially as the observations get older, making the most recent observations more impactful for the forecast. It is this exponential property of the weights that are the basis for the name Exponential Smoothing [Hyndman et al., 2008]. By using the weighted average of all historic observations, random fluctuations among them are smoothed out [Holt, 2004]. ES methods are the most used class of forecasting methods, and this section examines the most commonly used adaptions of them.

**Decomposing the Time Series** A time series can consist of several different components or pattern types. In order to accurately forecast the time series as a whole, it is often helpful to decompose it as seen in Figure 3. [Hyndman and Athanasopoulos, 2019]. Each pattern type can then be assessed separately while forecasting the future values. We will focus on the following types of time series patterns:

- **Level:** The height of the pattern, or the average value.
- **Trend:** The progression of the time series. Which direction it moves in the long run.
- **Seasonality:** Patterns that are repeated with a fixed time period.
- **Noise:** Random variations that cannot be explained by either the Trend or the Seasonality.

### .1 Simple Exponential Smoothing - No Trend or Seasonality

The most simple ES method is called Simple Exponential Smoothing (SES). SES is applicable when the observed data fluctuates randomly around a mean, and no trend or seasonality are assumed to have an impact [Kalekar, 2004]. It is given by the equation:

$$\hat{y_t} = \alpha y_t + (1-\alpha)\hat{y}_{t-1} = \hat{y}_{t-1} + \alpha(y_t - \hat{y}_{t-1}) \tag{2}$$

Where $\alpha$ is the smoothing parameter with $0 \leq \alpha \leq 1$, $y_t$ is the actual data at time $t$, and $\hat{y_t}$ is the forecast value at time $t$. The size of the smoothing parameter determines the rate of decrease for the weights [Hyndman and Athanasopoulos, 2019]. If the $\alpha$ value is close to 1, it means that the recent observations are weighted more heavily. Conversely, a lower $\alpha$ means that the more distant observations have higher impact on the forecast. By substituting directly and utilizing the recursiveness of (2), we can illustrate the exponential property of ES:

$$\hat{y_t} = \alpha y_t + (1-\alpha)\hat{y}_{t-1} + (1-\alpha)^2\hat{y}_{t-2} + (1-\alpha)^3\hat{y}_{t-3} + ... \tag{3}$$

ES methods can also be represented in component form [Hyndman and Athanasopoulos, 2019]. For SES this representation does not provide much of a different perspective, but it is useful for comparison with the extension methods. We include the level component $l_t$ and a time step variable $h$ to be able to forecast beyond the last data point:

$$\begin{aligned} \text{Forecast Equation: } & \hat{y}_{t+h} = l_t \\ \text{Level Equation: } & l_t = \alpha y_t + (1-\alpha)l_{t-1} \end{aligned} \tag{4}$$

All future forecasts will have the same value, regardless of the value of $h$. This makes SES suitable for time series without any trend or seasonality [Hyndman and Athanasopoulos, 2019].

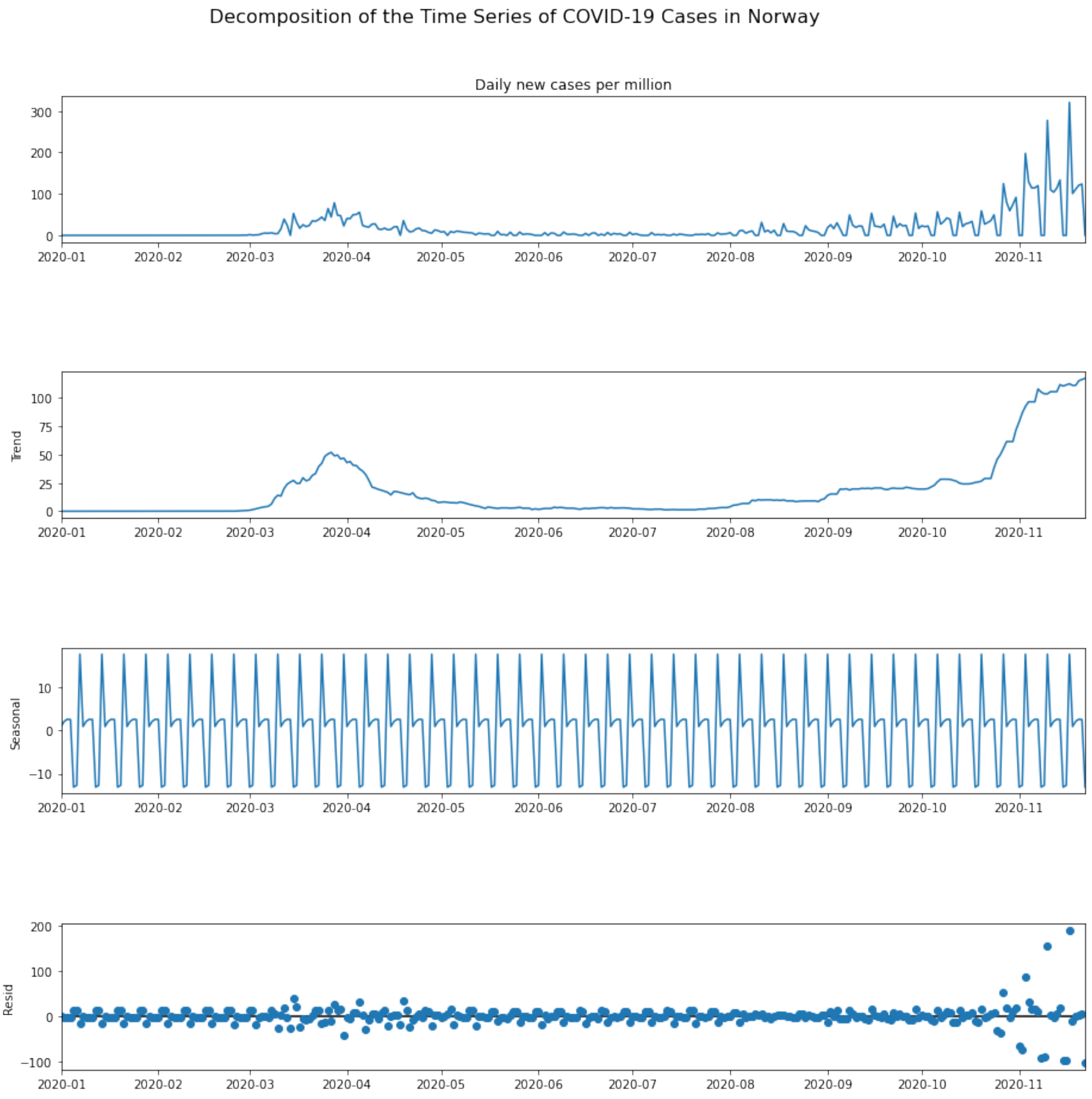

Figure 3: Data of new COVID-19 cases per million in Norway decomposed into a trend pattern, a seasonal pattern, and with the remaining random pattern at the bottom.

## .2 Methods With Trend

If there is a trend among the observations, meaning that the observations fluctuate randomly around a non-horizontal trend line, we can use an extended version of the SES method to incorporate that trend into the forecast. The trend can be modeled as either linear, dampened or exponential. We will elaborate on linear and dampened trend in this section.

**Holt's Linear Method** Charles Holt extended the SES method by adding another smoothing equation to the one in (4). In addition to the Level Smoothing equation $l_t$, he introduced an equation for Trend Smoothing [Hyndman and Athanasopoulos, 2019]. This is denoted by $b_t$:

$$\begin{aligned} \text{Forecast Equation: } & \hat{y}_{t+h} = l_t + hb_t \\ \text{Level Equation: } & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation: } & b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \end{aligned} \tag{5}$$

Where $\beta$ is the trend smoothing parameter with $0 \leq \beta \leq 1$, and $h$ is introduced in order to forecast values further ahead at time $t+h$. As per equation (5), the trend element of the forecast is a linear function of $h$. This means that the trend will either increase or decrease linearly into the future, as a smoothed estimate of the average growth rate [Kalekar, 2004].

**Dampened Trend** Empirical evidence have shown that methods with a constant trend, like Holt's Linear, have the tendency to overestimate the trend in the long run [Hyndman and Athanasopoulos, 2019]. A dampening parameter was introduced by Gardner and McKenzie [1985] to compensate for that error by flattening the trend over time. Using $0 < \Phi < 1$ as the dampening parameter, we can adjust equation (5), and get:

$$\begin{aligned}\text{Dampened Forecast Equation: } & \hat{y}_{t+h} = l_t + (\Phi + \Phi^2 + ... + \Phi^h)b_t \\ \text{Dampened Level Equation: } & l_t = \alpha y_t + (1-\alpha)(l_{t-1} + \Phi b_{t-1}) \\ \text{Dampened Trend Equation: } & b_t = \beta(l_t - l_{t-1}) + (1-\beta)\Phi b_{t-1}\end{aligned} \tag{6}$$

Thus, Damped Trend Methods will converge towards a constant as $h \longrightarrow \infty$.

### .3 Methods With Seasonality

To take into account the seasonality of the time series, Holt [2004] together with Winters introduced a third smoothing variable as a further extension of Equation (5) [Hyndman and Athanasopoulos, 2019].

**Holt Winters' Additive Method** The seasonality equation is denoted $s_t$, and we introduce a seasonality smoothing parameter, $0 \leq \gamma \leq 1$, while $m$ is the frequency of the seasonality [Hyndman and Athanasopoulos, 2019], e.g. 12 months in a year.

$$\begin{aligned}\text{Forecast Equation: } & \hat{y}_{t+h} = l_t + hb_t + s_{t+h-m} \\ \text{Level Equation: } & l_t = \alpha(y_t - p_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation: } & b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ \text{Seasonal Equation: } & s_t = \gamma(y_t - l_{t-1} - bt - 1) + (1-\gamma)s_{t-m}\end{aligned} \tag{7}$$

This method is called additive because the forecast $\hat{y}_{t+h}$ is the sum of the level, trend and seasonality elements. The seasonal variations over m consecutive periods will sum to zero. It is the preferred method when the seasonal variations are fairly constant through the entire time series [Hyndman and Athanasopoulos, 2019].

**Holt Winters' Multiplicative Method** When the variations in seasonality are changing proportionally to the level through the time series, the multiplicative method is preferred [Hyndman and Athanasopoulos, 2019]. With this method, the seasonal variations over m periods will sum to $m$.

$$\begin{aligned}\text{Forecast Equation: } & \hat{y}_{t+h} = (l_t + hb_t)s_{t+h-m} \\ \text{Level Equation: } & l_t = \alpha\frac{y_t}{p_{t-m}} + (1-\alpha)(l_{t-1} + b_{t-1}) \\ \text{Trend Equation: } & b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1} \\ \text{Seasonal Equation: } & s_t = \gamma\frac{y_t}{l_{t-1} + bt - 1} + (1-\gamma)s_{t-m}\end{aligned} \tag{8}$$

# B Random Forest Results from Iterative Experimentation

Table 4 shows a selected sample of validation results on the iterative experimentation we conducted for RF. As indicated in the *Models* column, we experimented with (i) training one model on all the time series, and (ii) training multiple models (one for each time series). The target encoding of each country's iso code is only relevant for the first case.

| Models | Lags | Stringency index | Target encoding | Validation RMSE |
|---|---|---|---|---|
| 1 | 3 | No | Yes | 150.20 |
| 1 | 7 | No | Yes | 145.50 |
| 1 | 7 | Yes | Yes | 144.49 |
| 1 | 14 | Yes | Yes | 145.26 |
| 1 | 7 | No | Yes | 144.62 |
| 1 | 7 | No | No | 147.06 |
| Multiple | 7 | Yes | - | 137.4 |
| Multiple | 14 | Yes | - | 136.09 |
| Multiple | 14 | No | - | 136.63 |
| Multiple | 21 | Yes | - | 419.56 |

Table 4: A selected sample of validation results on the iterative experimentation conducted for RF.