



**Universiteit  
Leiden**  
The Netherlands

## **Automated machine learning for COVID-19 forecasting**

Tetteroo, J.; Baratchi, M.; Hoos, H.H.

### **Citation**

Tetteroo, J., Baratchi, M., & Hoos, H. H. (2022). Automated machine learning for COVID-19 forecasting. *Ieee Access*, 10, 94718-94737. doi:10.1109/ACCESS.2022.3202220

Version: Publisher's Version

License: [Creative Commons CC BY 4.0 license](https://creativecommons.org/licenses/by/4.0/)

Downloaded from: <https://hdl.handle.net/1887/3483916>

**Note:** To cite this publication please use the final published version (if applicable).

## APPLIED RESEARCH

# Automated Machine Learning for COVID-19 Forecasting

JACO TETTEROO<sup>1</sup>, MITRA BARATCHI<sup>1</sup>, AND HOLGER H. HOOS<sup>2</sup><sup>1</sup>Leiden Institute of Advanced Computer Science (LIACS), Leiden University, 2333 CA Leiden, The Netherlands<sup>2</sup>Chair for AI Methodology (AIM), RWTH Aachen University, 52062 Aachen, Germany

Corresponding author: Mitra Baratchi (m.baratchi@liacs.leidenuniv.nl)

This work was supported by TAILOR, a Project through the European Union (EU) Horizon 2020 Research and Innovation Program under Grant 952215.

**ABSTRACT** In the context of the current COVID-19 pandemic, various sophisticated epidemic and machine learning models have been used for forecasting. These models, however, rely on carefully selected architectures and detailed data that is often only available for specific regions. Automated machine learning (AutoML) addresses these challenges by allowing to automatically create forecasting pipelines in a data-driven manner, resulting in high-quality predictions. In this paper, we study the role of open data along with AutoML systems in acquiring high-performance forecasting models for COVID-19. Here, we adapted the AutoML framework auto-sklearn to the time series forecasting task and introduced two variants for multi-step ahead COVID-19 forecasting, which we refer to as (a) multi-output and (b) repeated single output forecasting. We studied the usefulness of anonymised open mobility datasets (place visits and the use of different transportation modes) in addition to open mortality data. We evaluated three drift adaptation strategies to deal with concept drifts in data by (i) refitting our models on part of the data, (ii) the full data, or (iii) retraining the models completely. We compared the performance of our AutoML methods in terms of RMSE with five baselines on two testing periods (over 2020 and 2021). Our results show that combining mobility features and mortality data improves forecasting accuracy. Furthermore, we show that when faced with concept drifts, our method refitted on recent data using place visits mobility features outperforms all other approaches for 22 of the 26 countries considered in our study.

**INDEX TERMS** Automated machine learning, time series forecasting, concept drift, COVID-19, mobility data.

## I. INTRODUCTION

In December 2019, a coronavirus disease (COVID-19), caused by the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), emerged in the city of Wuhan, China. By January 2020, the World Health Organisation advised governments to prepare for active surveillance and case management [1]. For policymakers to respond adequately, the ability to accurately forecast the spread of the disease is essential. This has inspired many researchers to work on forecasting methods in response to the COVID-19 pandemic based on available data. Such data may be in the form of the number of confirmed cases, deaths, hospitalisations or vaccinations. Agencies such as the European Centre

for Disease Prevention and Control [2] have invested substantial effort into consolidating such data sources. Furthermore, several technology companies – including Apple, Facebook, Foursquare and Google – have published data reflecting the movement of people within a population. These data sources are interesting with respect to COVID-19 forecasting, as the movement of people is directly related to the spread of the contagious disease.

Despite such efforts, Ioannidis *et al.* [3] claim that forecasting for COVID-19 has majorly failed. They argue that draconian countermeasures have been taken on the basis of incorrect modelling assumptions, poor data quality and high sensitivity of estimates due to exponentiated variables. Early models have built upon speculations while predicting for entire seasons. As a result, many forecasting models would only work well for isolated homogeneous populations

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

but not for richer real-world scenarios. In this paper, we study to understand why forecasting COVID-19 is difficult and how available open data and recent advancements in machine learning can be used to create accurate forecasting models.

Constructing high-performance machine learning pipelines is challenging. A forecasting task pipeline typically consists of many steps, such as data pre-processing, feature extraction and selection, and model fitting. For each of these steps, choices need to be made, and corresponding hyperparameters need to be tuned that may impact the accuracy of predictions. Manual tuning often relies on simplifying assumptions, which may not fully capture the underlying characteristics of the data. Automated machine learning (AutoML) is a recently growing field that enables users to construct high-performance classification or regression pipelines in a fully automated manner. AutoML has been demonstrated to extract competitive and high-quality models automatically in various applications, often outperforming manually tuned models. Successful AutoML frameworks include *auto-sklearn* [4], *Auto-Keras* [5] and *Auto-WEKA* [6].

In this work, we approach the forecasting of COVID-19 mortality as a time series regression task and explore how AutoML can be deployed for this goal. Specifically, we look into the role of different datasets and how to account for data drift when the underlying concept generating the data changes. We are forecasting COVID-19 substantially later than early work, such as [7], [8], [9]. This gives us the opportunity to use methods that need more training data (e.g., deep learning models) and would not be feasible in early forecasting scenarios. We adapt the well-known and freely available AutoML framework *auto-sklearn* [4] to the task of COVID-19 forecasting. Extending this framework, in this paper, we aim to investigate the usefulness of various disease and mobility datasets, analysing how they can best be used for COVID-19 forecasting. There are two main challenges to achieve this goal. The first challenge lies in the fact that *auto-sklearn* was not explicitly designed for the task of time series forecasting. Time series forecasting introduces extra parameters to set for instance to allow for suitable input window sizes and forecasting horizons. The second challenge is that available AutoML frameworks such as *auto-sklearn* are designed for data with a stable data generation process. The pandemic, however, has a complex data generation process. As lock-down policies are applied, the virus mutates, or vaccination campaigns modulate its spread, the underlying data generation process undergoes substantial changes. This type of concept drift necessitates adaptations to this framework in order to ensure high-quality forecasting results. Our contributions in this paper are as follows:

- We adapt the *auto-sklearn* AutoML framework to the task of forecasting COVID-19 mortality data and introduce two AutoML forecasting variants for multi-step ahead time series forecasting.
- We study how we can incorporate anonymised mobility data representing place visits and the use of different

transportation modes. We also study to which extent doing so permits more accurate forecasting.

- We extend this framework to take into account non-stationarity and concept drift in the data by comparing the performance of three different drift adaption strategies.
- We evaluate our methods on real-world datasets from 58 countries worldwide and against five baselines.

## II. RELATED WORK

### A. COMPARTMENTAL MODELS

Traditionally, epidemics are charted using compartmental models, like the SIR model [10]. This model splits the population of individuals in different compartments based on their health status. At each time step, the flow of individuals transitioning from one compartment to the other is described by differential equations, representing contact ratios and recovery time. Given more knowledge about a given disease, more complex compartmental models may be created by adding more compartments that reflect that knowledge. The SEIR model [11], for instance, extends the SIR model by injecting the exposed compartment, holding people infected by the disease but not yet capable of infecting others.

### B. COMPARTMENTAL MODELS WITH CONTACT NETWORKS

Basic compartmental models require the unrealistic assumption that the population is homogeneous and every individual has an equal amount of contact with every other individual. To become more realistic, compartmental models may be extended with contact networks. Liu *et al.* [12] used a multi-layered contact network – where each layer entails a mode of contact – and an SIR model to simulate the propagation of flu. They show that this approach gives more insights about the underlying dynamics of the spread of diseases. Balcan *et al.* [13], similarly used a multi-scale network to simulate an influenza-like disease. Instead of individuals, they used sub-populations as nodes and gravitational flow derived from commuting and flight data as weights for the edges introducing a form of spatial awareness to the compartmental models.

In order to create realistic contact networks, detailed mobility datasets are needed. Ideally, these datasets encompass the entire population of a region, detailing where and how people have come in contact with each other. In reality, datasets often summarise interactions and often present samples of a population. Also, recorded interactions in these datasets are not enriched with duration, or intensity [14]. Contact networks where individuals are simulated as a basis for the spread of diseases are called agent-based networks. To create agent-based networks, one needs datasets containing the movement patterns of individuals. For instance, Aleta *et al.* [15] created an agent-based network using a dataset containing place visits published by Foursquare to simulate the spread of COVID-19 through a synthetic

population in the Boston metropolitan area. While for some countries, the mobility data is detailed enough to create realistic contact networks, for most, this is not the case. In many countries, mobility data is considered as personal data and should not be collected. In our work, we make predictions for a large number of countries. Instead of detailed mobility data on the individual level, we use aggregated mobility data on a national level. This is not detailed enough to construct contact networks. However, such datasets can easily be shared and used as features for regressors and extracting general knowledge from global data that can be useful for forecasting.

### C. AUTOREGRESSIVE MODELS

Another classic approach is to use autoregressive methods. An autoregressive model is a regression model where the input variables are observations from previous time steps. ARIMA models were successfully deployed to forecast COVID-19. Kumar *et al.* [16] used the ARIMA model to analyse the trend of 15 countries during the first three months of the pandemic. Alzahrani *et al.* [17] compared the ARIMA model with the simpler AR, MA and ARMA models making forecasts for four weeks for Saudi Arabia and found that ARIMA outperformed the others. Chakraborty and Gosh [9] extended an ARIMA model by adding a wavelet transformation on the residuals of the model. This improved the forecasts and was tested for Canada, France, India, South Korea and the UK on a forecasting range of ten days.

### D. DEEP LEARNING MODELS

Recently, deep learning methods got applied to forecast epidemics. One of such approaches is the work by Wu *et al.* [18] who predicted flu in the United States using a combination of CNN, RNN and residual links. They achieved a robust improvement over autoregressive models using multiple real-world datasets. Aiken *et al.* [19] compared autoregressive models with a GRU RNN to predict flu prevalence. They found that on larger prediction horizons, the RNN achieved significantly lower RMSE. Fu *et al.* [20] predicted influenza using an attention-based LSTM. One of the observations they made was that the sequence length of their training data highly influenced the performance of their model. Applied to COVID-19, many other work has been performed using LSTMs [8], [21], [22], [23]. Shahid *et al.* [24] perform a comparative study using a GRU, LSTM and Bi-directional LSTM. To train deep neural networks, one needs a lot of training instances. As for early epidemics, the number of instances is limited, and it may be challenging to create sufficiently detailed models. Typically, the architecture used has a great influence on the performance of the model and should be carefully constructed. In our work, this is not necessary as we use the underlying characteristics of the pandemic to automatically create our models. It is possible to automatically construct deep learning architectures via Neural Architecture Search, using for example, Auto-Keras [5] or

Auto-Pytorch [25], but as these need large quantities of data, the amount of data available in epidemics may be insufficient.

### E. AUTOMATED MACHINE LEARNING METHODS

The creation of regression pipelines encompasses many steps; data pre-processing, feature pre-processing, hyperparameter optimisation and algorithm selection. The best choice of the algorithm, pre-processing step and further how to set their hyperparameters, typically depends on the data at hand. Therefore, it is difficult to select a single algorithm to ensure that the best model is configured for a forecasting problem. Different choices of these components may vastly influence the predictive performance of the pipeline, which is why we can benefit from making these choices automatically. AutoML systems have recently addressed this issue through developing techniques to automatically configure high-performing machine learning pipelines. Sequential Model Based Optimisation (SMBO) is a black box optimisation framework that has been used for the purpose of hyperparameter optimisation. Hutter *et al.* [26] used (SMBO) to automatically optimise hyperparameters of machine learning algorithms. Sequential Model-based Algorithm Configuration (SMAC) [27] is a system that implements SMBO and can be used for hyperparameter optimisation. This is a general-purpose algorithm configurator, which makes it possible to both select algorithms and tune their hyperparameters efficiently. Auto-WEKA [6] is an AutoML framework around the WEKA software package using SMAC for its configuration. This framework fully automated the creation and tuning of classification and regression pipelines. Auto-sklearn [4] is an AutoML framework by Fuerer *et al.* around the scikit-learn [28] Python package. This framework includes meta-learning to warm start the configuration search and creates ensembles of pipelines. In more recent updates, this framework is updated with multi-output regression. This option makes it suitable for forecasting with a range of multiple days. TPOT [29] is a tree-based pipeline optimisation tool for AutoML. Similar to auto-sklearn, it is built upon scikit-learn. Instead of using SMBO, TPOT uses genetic programming for hyperparameter optimisation. H2O [30] is another AutoML framework that uses the random search for its hyperparameter optimisation and combines models in stacked ensembles. Unlike auto-sklearn, H2O does not optimise data and feature pre-processors, but only optimises models. It is also possible to automatically construct deep neural networks. Frameworks that support this are Auto-Keras [5] and Auto-PyTorch [25], build Python packages. These frameworks find solutions to neural architecture search (NAS), where they aim to find the optimal neural network, minimising a loss function. Han *et al.* [31] used TPOT and H2O to forecast COVID-19 mortality data from Ceará. Their study found that TPOT outperforms regression models not automatically tuned, achieving a higher  $R^2$  score. Marques *et al.* [32] compared models produced with H2O with an LSTM network using data from the countries of Brazil, China, the United States of America, Italy, and Singapore and found

that H2O outperformed the LSTM in terms of MAE, MSE and  $R^2$ .

Among these frameworks in this work, we have selected to adapt auto-sklearn to the task of COVID-19 forecasting. As data is limited when forecasting the pandemic, using AutoML systems generating deep neural networks is unfeasible. Among frameworks based on classic machine learning algorithms, H2O does not support the automation of data and feature pre-processors which are both key components for time series forecasting to configure the auto-regressive model and its window size automatically. TPOT and auto-sklearn are comparable to each other in creating full pipelines. However, since TPOT relies on cross-validation to validate its pipelines, it is less suitable for time series forecasting tasks. The cross-validation scheme splits the data in  $k$  folds, training the models on  $k - 1$  folds and evaluating on the one that was left out. When the evaluation fold is earlier in time than the train folds, the model trains to predict past observations instead of the future one. Auto-sklearn supports holdout sets as a validation scheme, ensuring we can train our models without relying on future data. We further compare our work with deep learning and auto-regressive methods.

### III. PROBLEM STATEMENT

We view the forecasting of COVID-19 as a time series forecasting task. A time series holds discrete observations indexed over time. In our case, the rate over which the time series is sampled is constant due to the availability of daily case and mortality data. Considering a time series containing COVID-19 mortality numbers of length  $n$  as  $\mathbf{x} = [x_1, \dots, x_n]$  with  $x_t \in \mathbb{R}^n$ , a time series segmentation window of size  $w$ , a time step  $t$  and a forecasting horizon of size  $h$ , we want to use a segment of historical observations  $\mathbf{x}_{t,w} = [x_{t-w}, \dots, x_t]$  from the time series up to observation  $x_t$  to forecast future data points  $\mathbf{x}_{t,h} = [x_{t+1}, \dots, x_{t+h}]$ . For the task of COVID-19 forecasting, the time series we consider are the mortality rate of a country, where  $x_t$  denotes the number of new deaths at time step  $t$ . When we consider using mobility time series  $\mathbf{m} = [\mathbf{m}_1, \dots, \mathbf{m}_n]$  alongside mortality data  $\mathbf{x}$ , we extend the notation to use a bivariate time series  $\mathbf{x}\mathbf{m}_{t,w} = [x_{t-w}, \mathbf{m}_{t-w}, \dots, x_t, \mathbf{m}_t]$  for the forecasting of  $[x_{t+1}, \dots, x_{t+h}]$ . In our approach,  $\mathbf{m}_t$  is a vector holding a number of features in terms of the percentual increase of mobility for a country in a given form, at timestamp  $t$ , such as the increase of time spent driving or the increase of time spent visiting recreational areas. This format is dictated by the mobility data provided by Apple [33] and Google [34], which we study in this work. To make comparisons between different countries, areas or cities possible, we normalise the mortality data by the size of its population  $N$ .

Since the infectiousness of COVID-19 may change over time, for instance, due to mutations or vaccinations, the underlying concept generating the data may change. Vast changes to the concept are detrimental to the performance of machine learning algorithms. This change is known as

concept drift. In Equation 1, we show a formal definition of concept drift between two time steps  $t_0$  and  $t_1$  [35].

$$\exists X : p_{t_0}(X) \neq p_{t_1}(X) \quad (1)$$

In this definition,  $p_{t_0}$  is the joint distribution between the set of input sequences  $X$  where  $\{\mathbf{x}, \mathbf{m} \in X\}$ .

We aim to address the forecasting task by formulating the Combined Algorithm Selection and Hyperparameter (CASH) Optimisation problem [6]. Given a set of machine learning algorithms  $\mathcal{A} = A^{(1)}, \dots, A^{(k)}$  with hyperparameter spaces  $\Lambda^{(1)}, \dots, \Lambda^{(k)}$ , we search the optimal algorithm with optimal hyperparameter settings  $A_{\lambda^*}$  following Equation 2.

$$A_{\lambda^*} \in \operatorname{argmin}_{A^{(i)} \in \mathcal{A}, \lambda \in \Lambda^{(i)}} \frac{1}{k} \cdot \sum_{i=1}^k \mathcal{L}(A_{\lambda}^{(i)}, \mathcal{D}_{\text{train}}^{(i)}, \mathcal{D}_{\text{valid}}^{(i)}) \quad (2)$$

Here  $\mathcal{L}$  is the loss generated by algorithm  $A$  when trained using set  $\mathcal{D}_{\text{train}} \in X$  and validated using set  $\mathcal{D}_{\text{valid}} \in X$ . This loss is the mean squared error between the forecast made by algorithm  $A$  using  $\mathbf{x}\mathbf{m}_{t,w}$  and with hyperparameter settings  $\lambda$  (i.e.,  $\hat{\mathbf{x}}_{t,h}$ ) and the true observations in the validation set (i.e.,  $\mathbf{x}_{t,h}$ ), unseen by algorithm  $A$ . We are optimising a full pipeline. Therefore, optimising  $A$  means that we are optimising the hyperparameters of a combination of pre-processors  $P$ , features  $F$  and regressors  $R$ , or  $A = \{P, F, M\}$ . Part of this process is internally optimising the input window size  $w$ , which is a newly added feature pre-processing step for time series forecasting.

### IV. METHODS

As discussed in Section II, in this work, we extend auto-sklearn to address the problem mentioned in Section III, as it supports multi-output regression and holdout validation. Furthermore, it supports automation of data and feature pre-processing steps, which are both important for time series forecasting to configure the auto-regressive model and set its window size.

Still, as this system was not necessarily created to perform time series forecasting, we add an additional variable input window size as feature pre-processor and introduce a new way to perform multi-step ahead forecasting. In this section, we provide the details on the data used in this work and how we adapted auto-sklearn to perform the forecasting task. Finally, we specify how we adapt the auto-sklearn ensembles when faced with concept drifts.

#### A. DATA

The data used for our predictions comes from three sources: mortality data and mobility data representing two types of mobility modalities: (i) the mode of transport and (ii) place visits. Table 1 presents the meta-data of these sources.

##### 1) THE MORTALITY DATA

This is collected by the European Centre for Disease Prevention and Control (ECDC) [36], [37]. The data is split into two

**TABLE 1. Meta-data of data sources. The end dates marked with an asterisk (\*) are not actual end dates, as these datasets are at the date of writing still updated regularly.**

Data source	Originality	Category	Countries	Start date	End date
ECDC 1 [36]	Original	Mortality	214	2019-12-31	2020-12-14
ECDC 2 [37]	Original	Mortality	30	2021-02-28	2021-07-10*
Apple [33]	Original	Mode of transport mobility	63	2020-01-13	2021-07-10*
Google [34]	Original	Place visits mobility	135	2020-02-15	2021-07-10*
2020	Merged	Combined	58	2020-02-15	2020-12-14
2021	Merged	Combined	26	2020-03-01	2021-07-10

sets, with the main difference being the period over which time series are collected and the number of countries. Both datasets hold the daily number of new cases and new deaths. Additionally, they provide the country population size of the previous year. For the first dataset, this is the population size of 2019, and for the second dataset, this is the population size of 2020. The ECDC 1 dataset has data from December 31st, 2019 until December 14th, 2020. Not all countries have values at the start of the dataset, as COVID-19 was not first encountered in all countries at the same time. The data is provided for 214 countries from all around the world. The ECDC 2 dataset contains more recent data starting on the first of March 2021 and is still being updated daily. The data in this set is collected for 30 countries in the European Union. Both datasets are maintained and adjusted by ECDC when numbers are deemed inaccurate due to delays in reporting. We use the daily new deaths as part of our input and as truth value to evaluate our estimations. We do so because the reported deaths are likely to be more reliable than reported cases, as mentioned by [7]. To make sure the data is comparable between countries, we normalise the daily new deaths to depict the number of daily new deaths per 1,000,000 people within the population.

## 2) THE APPLE MOBILITY TREND REPORTS [33] (MODE OF TRANSPORTATION)

This data contains the percentual increase or decrease of the use of modes of transportation as compared with a baseline volume on January 13th, 2020. The modes of transportation specified are *walking*, *driving* and use of *transit*. However, this latter mode is not available for all countries. Therefore, in our features, we only use the increase or decrease in the use of walking and driving as means of transportation. The dataset includes data starting from January 13th, 2020. It holds data for 63 countries, excluding many African countries.

## 3) THE GOOGLE COMMUNITY MOBILITY REPORTS [34] (PLACE VISITS)

This data contains the percentual increase or decrease of place visits as compared with a baseline period from January 3th to February 6th, 2020. The places are categorised in the following six categories: retail and recreation, grocery and pharmacy, parks, transit stations, workplaces and finally, residential. The dataset starts on February 15th, 2020. It holds data for 135 countries.

## 4) COMBINED DATA

We merged the mortality data and the mobility data into two combined datasets. The first combined dataset captures the first year of the pandemic. We used the intersection of dates and countries of the first ECDC dataset and both mobility datasets. There were some missing values, which we imputed by taking the average of the values 7 days before the missing data point and 7 days after the missing data point. This way, the imputed value fits well between the previous and next week and daily trends are preserved. For the country of Serbia, the number of missing values exceeded 10%, which is why we omitted it from the dataset. The resulting first combined dataset contains data from February 15th, 2020 until December 14th, 2020. The second combined dataset contains data from March 1st, 2021 until July 10th, 2021. When combining the mortality data with the mobility data for these periods, there were no missing values to account for. The first dataset includes 58 countries from all over the world. The second dataset contains 26 countries from the European Union.

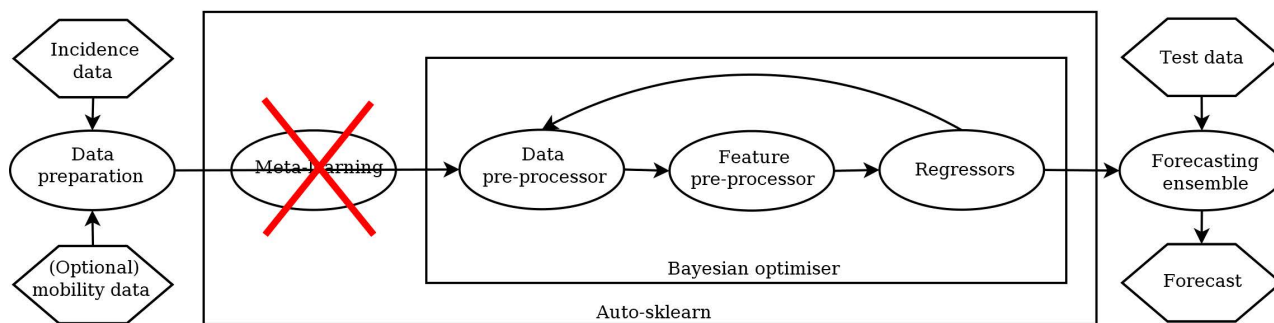
## B. FORECASTING STRATEGIES

Auto-regressive modelling is a common approach taken for forecasting tasks. An auto-regressive model performs regression using past measurements in a time series to predict its future timestamps. Many regression algorithms can be used to create an auto-regressive model. Furthermore, the data can be pre-processed in different ways within a machine learning pipeline before being fed into the regression algorithm. In this paper, we extend the auto-sklearn [4] AutoML framework to achieve this goal. Auto-sklearn is a wrapper around the popular Python module scikit-learn [28]. Scikit-learn is a machine learning library including a large set of algorithms that can be used for regression and classification tasks, providing various ways to pre-process data, select features, fit models and evaluate the results.

### 1) VANILLA AUTO-SKLEARN

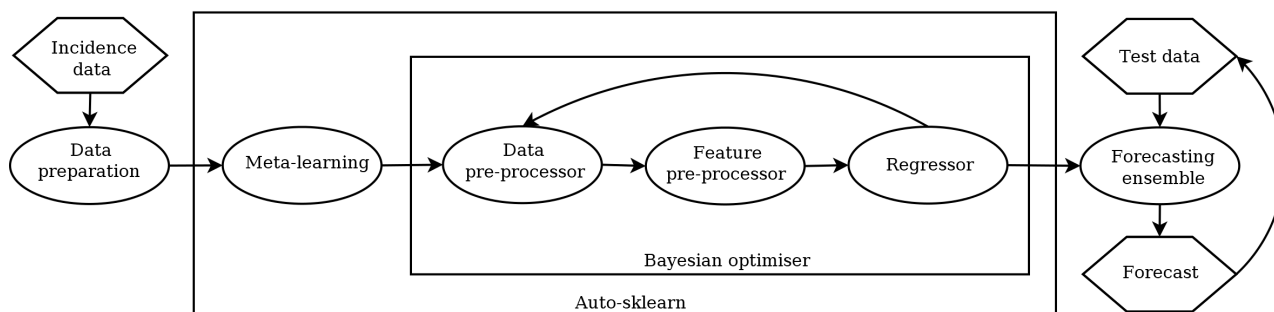
Auto-sklearn automates the process of creating good pipelines. Internally, it uses [27] SMAC, an SMBO framework. SMAC constructs a surrogate model capable of predicting the performance of an algorithm on the corresponding hyperparameter space (in this case, the space of all possible pipelines and hyperparameter settings). This model selects a list of promising configurations, evaluated on a validation set, based on their expected improvement

### Multi-output ensemble



**FIGURE 1.** The multi-output ensemble. This ensemble creates multiple predictions at once but has no access to meta-learning. Within the framework, pipelines are constructed to form a forecasting ensemble. By feeding this ensemble test data predictions can be made.

### Repeated single-output ensemble



**FIGURE 2.** The repeated single-output ensemble. This ensemble creates one prediction at each time step. To create predictions for a longer time period, for each new time step the predictions of previous steps are used. It has access to meta-learning, but cannot use additional data sources as input.

over the incumbent, the best-seen configuration. A local search is performed near these promising configurations to find configurations with higher expected improvement. In each iteration, the incumbent is updated to store the best found configuration. The best configurations are grouped together in an ensemble using ensemble selection [38]. This ensemble method iteratively adds the configuration with the highest ensemble performance gain on the validation set. Configurations grouped in the ensemble, each create their own forecast, which is averaged to create the ensemble forecast. The process of constructing an optimal pipeline can be warm-started using a meta-learning module. Before the search for good pipelines starts, the input dataset is compared with 140 datasets from the OpenML [39] repository. Then, configurations are selected which are known to perform well on similar datasets. In the rest of this section, we will explain different adaptations made to auto-sklearn. Auto-sklearn is not developed for modelling time series data, which is why we add a variable window size: a feature pre-processor that changes the number of days used as the input sequence. Additionally, there are two ways we approach the multi-step ahead forecasting: via (i) multi-output regression and (ii) repeated single-output regression. We detail these additions in the following parts.

#### 2) DEFINING A FORECASTING FRAMEWORK BASED ON AUTO-SKLEARN WITH VARIABLE INPUT WINDOW SIZE

To predict the value of  $[x_{t+1}, \dots, x_{t+h}]$ , we train the models with sequences of the time series in the form of  $[x_{t-w}, \dots, x_t]$ . In vanilla auto-sklearn this window size  $w$  has to be determined by the user. This would mean that when we use lags of the time series as features, the number of lags is predetermined. When making predictions with different regressors, not all parts of the time series may be relevant and depending on the configuration, it can be good to use a longer or shorter input sequence. This is why we implement the *variable window size* feature pre-processor as proposed in our earlier work [40]. This pre-processor has the hyperparameter  $w$  that is optimised within auto-sklearn. The pre-processor takes the input sequence with predetermined static length and cuts off the first values, resulting in an input sequence in the form of  $[x_{t-w}, \dots, x_t]$ . The work presented in [40] experiments on a large set of time series tasks and showed that the variable window size had major impact on the accuracy of the framework. We still need to set a maximum value for the window size. As larger windows limit the number of data instances we can use, we limit our window size to a maximum of 30 days. By incorporating the variable window size optimisation in auto-sklearn, it

possible to define a forecasting task in the following two ways:

- **Multi-output.** Since version 0.8, auto-sklearn supports multi-output regression, such that forecasts with forecast horizon  $h > 1$  may be performed without the need for training multiple models. We use this feature to define multi step-ahead forecasting in our method and refer to it *multi-output*. We show a schematic overview of this method in Figure 1. To make a multi-output prediction, separate regressors are fitted for each value of output. This means that each model consists of  $h$  regressors. As this output format was implemented much later than others, there is no meta-learning available for multi-output regression.
- **Repeated single-output.** The *repeated single-output* forecasting scheme with a horizon of  $h > 1$  is a model trained for single-output regression, but once it starts forecasting, its output is appended to the input sequence. For instance, when we want to predict the value of  $x_{t+2}$ , we use the sequence  $[x_{t-w+1}, \dots, x_t, x'_{t+1}]$  as input. In this sequence,  $x'_{t+1}$  denotes the prediction of value  $x_{t+1}$ . Note that when we append values to the input sequence, we remove values at the start of the sequence. Each model uses one regressor. We show a schematic overview of how this approach is implemented in Figure 2. The advantage of this method over the multi-output regression method is that it benefits from meta-learning. However, as it is not trained specifically for forecasting multiple days in future, predictions further away may suffer from errors made earlier. Another disadvantage is that this method can not use external changing variables as input, as only one time series is predicted.

### 3) AUTO-SKLEARN PARAMETER SETTINGS

As tuning many hyperparameters requires lots of data instances to prevent overfitting, we put together the time series of all countries in the training dataset, as opposed to training separate models for separate countries. This way, we create a joint model capable of forecasting for many countries. We normalised the mortality data by the size of the population of each country. The mobility data depicts percentual changes in mobility, which does not require further normalisation to make comparisons between countries possible. To make sure it handles individual countries well, we pass the country name as a categorical feature to each instance. For testing, we separate time series per country again. This way, we can compare the forecasting quality between countries. The default setting for resampling strategies is the use of cross-validation. The resampling strategy dictates what parts of the training data is used to validate the models. Applied to time series, cross-validation would mean that for most folds, future data is used to predict previous values. To negate this problem, we use a holdout set for validation. This set is situated at the end of the training set, just previous to the start

of the test set, to be sure that the ensemble model generated by auto-sklearn can't learn future information. This is why we also disable shuffling. This keeps the temporal integrity of the data intact and ensures that the holdout validation set consists of the last dates in the train set. As an optimisation metric, we use the mean squared error to evaluate the performance of the pipelines. This ensures that the regressor line tries to fit the set of data points as close as possible. To ensure our ensembles are fully trained on the data, we refit the ensembles on the full train and validation set after validation is finished. This means that while the pipeline stays the same, the models are updated with both the train and validation set. This way, we make sure that there is no gap in knowledge just before the forecasting starts.

### C. DRIFT ADAPTATION

For the pandemic problem, it is important to consider the changes in the data generation process that lead to concept drifts in data. On the one hand, there may be a concept drift caused by the fact that in 2021, many countries in Europe started their vaccination programs. Furthermore, lock-downs, mutations in the disease and changes in healthcare can lead to additional concept drifts in the data. On the other hand, we use two mortality datasets, separated in time, each normalised with a different population size (the country population numbers have slightly changed from 2020 to 2021). Currently, auto-sklearn has no drift detection mechanism.

Celik and Vanschoren [41] created several concept drift adaptation mechanisms for automated machine learning frameworks. It is not trivial to use any drift detection methods during training models with autosklearn. This requires dynamically training multiple models to monitor the drift. However, autosklearn works with a predefined number of training instances to create a single model and cannot dynamically detect drift in consecutive windows of training data. As training a single autosklearn ensemble with sufficient complexity takes multiple hours, creating many ensembles for drift detection can quickly increase the time needed beyond feasibility. While in the problem of COVID-19 forecasting, we can safely assume that drift exists in data, further research can study how automatic drift detection techniques can be incorporated directly in autosklearn. We implement three methods based on the work of Celik and Vanschoren [41] that do not use drift detection to cope with concept drift. For each of the methods, we first construct ensembles using the old dataset. The drift adaptation strategies can be viewed as a forget mechanism, discarding old information in varying degrees. Depending on the magnitude of the concept drift there can be merit for each method. In our experiments, we study the performance of these approaches in forecasting. The methods are explained below:

- **Full refit.** The full refit method keeps the models trained on old data and after drift occurs uses the full combination of both datasets to refit the ensembles. This method places most emphasis on older data in comparison with



the others, as it trains the original models on the older data and uses it for refitting.

- **Partial refit.** The partial refit method also keeps the models trained on older data, but after drift occurs, it uses only the new dataset to refit the ensembles. This method still uses the older data in the form of ensembles, but the models are only updated with new data, placing more emphasis on the newer data.
- **Retrain.** The retrain method discards the ensembles and constructs new ones with the new dataset. This method forgets the old data altogether and only uses new data for its predictions.

## V. EXPERIMENTS

Our goal is to answer the following questions with our experiments (all resources for reproducing this research and results are available online<sup>1</sup>):

- **Q1:** How does the use of mobility data as features improve COVID-19 forecasting accuracy using our proposed AutoML approach?
- **Q2:** How does this framework perform in COVID-19 forecasting compared to baselines?
- **Q3:** Does adapting for concept drift help to improve COVID-19 forecasting accuracy using this AutoML approach?

Based on the data sources available and the change in the population numbers used for normalisation, we use two scenarios to address these questions.

### A. FIRST SCENARIO: 2020

The first scenario uses 58 countries from all over the world. We use an evaluation period of 30 days starting on 15 November 2020. The training data comprises time series data between 15 February 2020 and 14 November 2020, of which the last 30 days are used as a holdout validation set for our models.

### B. SECOND SCENARIO: 2021

The second scenario uses 26 countries from the European Union. This scenario has an evaluation period of 30 days, starting July the 11th in 2021. Depending on the drift adaptation technique, the way the data is used changes. In case of no adaptations, we use the data between 15 February 2020 and 14 December 2020, as well as between 1 March 2021 and 10 June 2021 as train data, of which the last 30 days are used for holdout validation. When refitting on the full dataset, we train between 15 February 2020 and 14 December 2020, of which we use the last 30 days for validation. Then, the data between 15 February 2020 and 14 December 2020, as well as between 1 March 2021 and 10 June 2021, is used to update the ensemble weights. When refitting on the partial dataset, we again train between 15 February 2020 and 14 December 2020, of which we use the last 30 days for validation. Then, we update the weights using only the

data between 1 March 2021 and 10 June 2021. Finally, when retraining the ensembles fully, we disregard the 2020 data, training only on data between 1 March 2021 and 10 June 2021, using the last 30 days for validation.

### C. BASELINES

We selected the following baselines based on earlier research in COVID-19 forecasting that use machine learning models and can train models based on the dataset we have collected. Compartmental methods (e.g., [15]) need specific data that is not available for all regions. Therefore, we cannot compare our methods with these:

- **Persistence.** The persistence baseline can give an idea of the minimum performance expected. When forecasting the window  $[x_{t+1}, \dots, x_{t+h}]$  each predicted value will be  $x_t$ , disregarding all previous values  $x_i$  with  $i < t$ .
- **ARIMA wavelet.** The ARIMA wavelet model [9] is the combination of an ARIMA model and a wavelet-based forecasting model. It fits an ARIMA model on the mortality data and then models the residuals via the wavelet model. We use the model as implemented by Chakraborty and Ghosh [9] for COVID-19 forecasting, but increase the number of forecasting days to align with the scenarios. The parameters of the ARIMA model controlling the order of autoregression, the order of differencing and the moving average are automatically configured using a grid search and the Akaike Information Criterion [42].
- **GRU, LSTM and Bi-LSTM.** To compare our framework with recurrent neural networks, we reproduce the GRU, LSTM and Bi-LSTM as studied by Shahid et al. [24] for COVID-19 forecasting. In our comparison, all three architectures share the same architectures, as chosen by [24] and shown in Table 2. We did, however, enlarge the batch size from 10 to 58 for the first scenario or 26 for the second scenario, which are the number of countries in the dataset. This allows the models to train for each country simultaneously without them being able to see future time steps. We also increased the number of time steps used as input to 30 to match the other ensembles and baselines in our comparison.

### D. EXPERIMENTAL SET-UP

Our framework is built on version 0.12.1 of auto-sklearn. Auto-sklearn requires users to define a maximum runtime. All of our ensembles, multi-output or repeated single output, were ran for 3 hours. For the training of every single pipeline, we limit the runtime of auto-sklearn to a maximum of 10% of the total runtime, which comes down to 18 minutes. The majority of iterations, however, finish much faster. This amount of time ensures that hundreds of models are compared to create the resulting ensembles. We run auto-sklearn in parallel on 8 cores, of an Intel(R) Xeon(R) CPU of 2.1 GHz with 10 GB of RAM. As mentioned before in Section IV, we use a holdout set as a validation strategy,

<sup>1</sup><https://github.com/AutoML4covid19/Forecasting>

**TABLE 2.** Hyperparameter settings for the GRU, LSTM and Bi-LSTM.

Hyperparameters	Values
No. of neurons	{16, 32, 64, 128}
Learning rate	0.001
Optimiser	Adam
Batch size	58 or 26, depending on the number of countries in the dataset
Epochs	300
Time steps	30

and make sure not to shuffle the data. As the internal performance metric, we use the mean squared error for evaluating the performance of models. We use the full default regressor and pre-processor search space and extend the feature pre-processor search space by adding the variable window size pre-processor. We limit the window pre-processor to a minimum of three days and a maximum of 30 days.

As the Bayesian optimisation used by auto-sklearn is stochastic, one run of the framework may optimise towards a locally optimal configuration, thus not yielding the actual optimal configuration. We perform bootstrapping to gain confidence in our predictions by creating a distribution over results. For each estimator we make, we run our framework 25 times. Repeating 1,000 times, we sample with replacement five ensembles from the 25 runs, of which we select the one with the lowest validation error. These 1,000 selected models form our bootstrap distribution used to evaluate the models on the test set. For each day within the forecasting horizon, we report on the mean forecast and the 95% confidence interval. We use our bootstrapping approach not only for our methods but also for the deep learning baselines.

To evaluate our methods we use the root mean squared error as defined  $RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (Y_t - \hat{Y}_t)^2}$ . Here  $Y_t$  denotes the true observation of our time series at time  $t$  and  $\hat{Y}_t$  the prediction of the model. As our ensembles create multiple predictions for each day, the daily average is used for  $\hat{Y}_t$ .

## VI. RESULTS

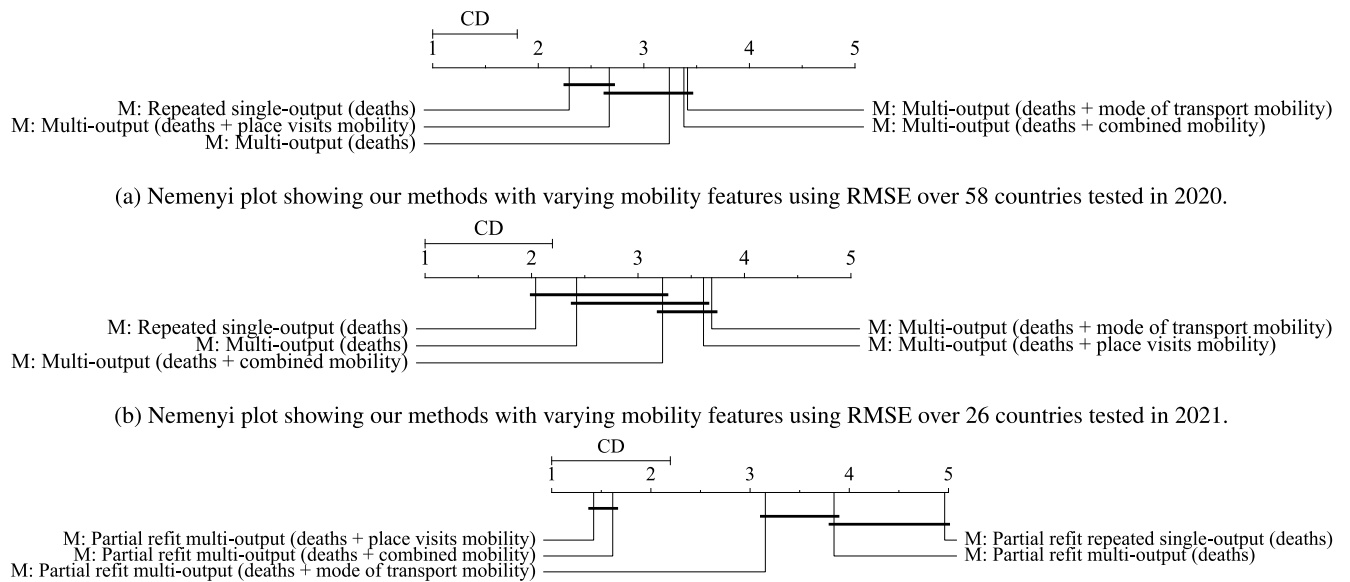
In this section, we answer the questions stated earlier in Section V. In the figures and tables, we denote our methods with the prefix  $M$  and the baselines with the prefix  $B$ .

### A. Q1: MOBILITY FEATURES

We initially aim to find answers to the question how does the use of mobility data as features improve COVID-19 forecasting accuracy using our proposed AutoML approach? This will allow to identify the most informative source of data for forecasting. To do so, we study the role of incorporating different types of mobility datasets (i) mode of transport, (ii) place visits and (iii) their combination on the quality of our automatically configured models. We perform this analysis for both scenarios using 2020 and 2021 datasets and when partial refit concept drift adaptation is performed. Because we want to compare the predictive performance of our methods for many different countries, we rank their performance based

on RMSE over all countries. These rankings come from a bootstrap distribution of 1,000 resamples, based on 25 runs per ensemble. A method that is consistently better than other methods in most countries will be assigned a lower average rank. These average ranks will give an insight on how well these methods perform compared to each other. For comparing the ranks, we use Nemenyi test [43], a standard test for inspecting the significant difference between average ranks. This test defines a critical distance between average ranks. Any method within a critical distance to another one is not significantly different. A critical distance diagram or a Nemenyi plot, such as those provided in Figures 3a and 3b can be used to visualize these rankings and their significance.

In Figure 3a, we compare our multi-output ensembles using different sources of mobility data and our repeated-single output ensemble for the 2020 scenario. In this scenario, the repeated single-output ensemble and the multi-output ensemble using place visits mobility have the best performance. The repeated single-output ensemble outperforms all multi-output ensembles not using place visits data. When we compare the same methods for the 2021 scenario in Figure 3b, we see that there is a drop in predictive power when using place visits mobility features. In this scenario, the repeated single-output and the multi-output ensemble using only mortality features are better than the ensembles using mobility features. The best mobility ensembles now use the combination of place visits and mode of transport, with place visits ranking slightly higher than the mode of transport. The drop in predictive power of the ensemble using place visits mobility can be explained by the concept drift and changes in data distribution in the second scenario. In this case, complex models with more features will lose to simpler models. In Figure 3c, we show the comparative performance of our methods in 2021 with the partial refit adaptation strategy. We found this strategy to be the best approach, as we will detail when discussing the answer to Q3. The figure indicates that mobility datasets can also show their power with proper drift adaptation in the second scenario. This experiment has shown that using mobility features can improve forecasts but does not guarantee improvement. Of the mobility datasets studied, the best results can be found using the place visits data. This dataset holds more predictive power than the mode of transport dataset. This may be due to their level of abstraction. The place visits data holds six categories, whereas the mode of transport has only two. Moreover, the place visits categories specify groups of locations instead



**FIGURE 3.** The comparative performance of our methods with varying mobility features using RMSE. A lower rank depicts a better performance. When methods are linked with a horizontal bar, they are within critical distance, meaning there is no significant difference between average ranks. Our methods are denoted with the prefix *M*.

of just an increase in activity. If more contagion happens at specific location groups, this can be picked up easier from the place visits data.

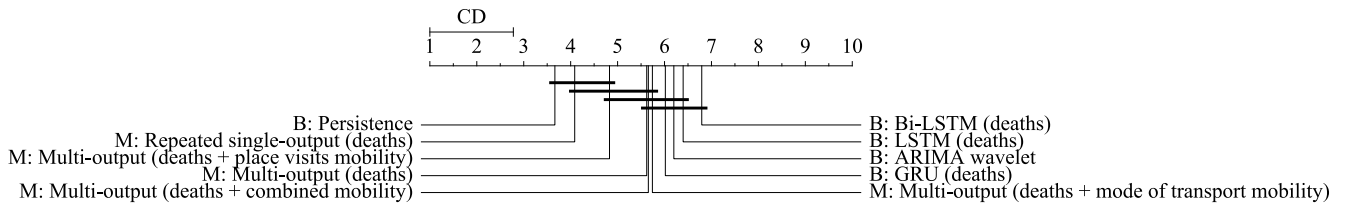
**B. Q2: COMPARISON TO BASELINES**

To place the previous results into perspective, we compare them to the baselines, answering **Q2**: How does this framework perform in COVID-19 forecasting compared to baselines? We compare the performance of the methods over 58 countries for 2020 and 26 countries for 2021.

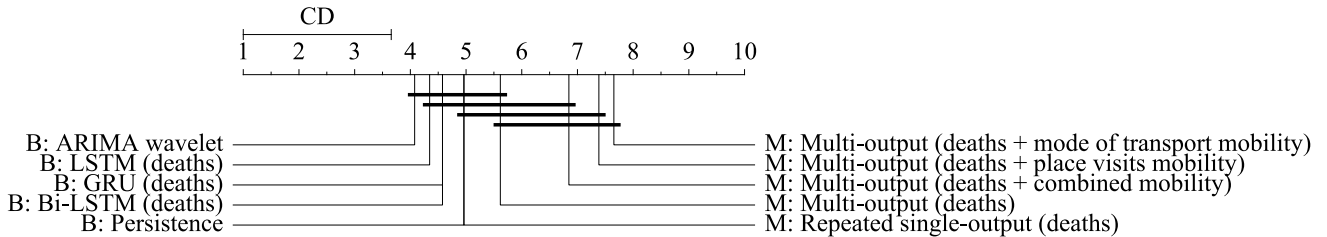
In Figure 4a, we show the results for the 2020 scenario. Here, the performance of the methods and baselines are close. The best baseline is the persistence baseline. Our best two ensembles perform slightly worse than the persistence baseline but outperform all other baselines significantly. Our other methods fall between our best methods and the other baselines. A lower rank for more complex deep learning baselines compared to simpler models, such as a persistence baseline, in Figure 4a is explained by the lack of enough training data in the first year that is necessary for training these models. The predictive power of these models, however, improves as more data becomes available in the second year, as shown in Figure 4b. In Figure 4b, we show the comparison with baselines for the 2021 scenario. While in the previous scenario, the persistence baseline was stronger than the deep learning baselines, it performs the worst here. Our methods, however, are all performing worse than the baselines. In the next section, we show how we can regain the power of our automatically configured models and mobility features using the concept drift adaptation techniques.

**C. FINE-GRAINED ANALYSIS OF RESULTS**

For deeper inspection of results per country, we show the RMSE of all methods and baselines for the 2020 scenario in the Appendix in Table 3. The table shows that for 20 out of 58 countries, the persistence baseline has the best forecast. However, as the first five of these have no new deaths in the test period, the persistence baseline wins in these by default as there are no fluctuations in the time series. Our best method for this scenario, the repeated single-output ensemble, scores best for 21 of the 58 countries. Using this table, we would further investigate if the performance of models depends on the properties of the time series acquired from different countries. Notably, we look at the existence of (i) periodic patterns and (ii) trends that point to the complexity of the time series. In this table, we grouped countries based on the trend and periodicity importance of the true values acquired using the procedure explained in [44]. To compute this importance, we split the true value time series  $Y_t$  into its trend  $T_t$ , periodicity  $P_t$  and remainder series  $E_t$ . Then, the trend importance can be computed as  $1 - \frac{Var(E_t)}{Var(T_t+E_t)}$  and the periodicity importance as  $1 - \frac{Var(E_t)}{Var(P_t+E_t)}$ . These measures range from 0 to 1, allowing us to group the countries into 4 quadrants. We indicate values lower than 0.5 as low and higher than 0.5 as high. When writing about quadrants, we mention trend importance first and periodicity importance second. The low-high quadrant, thus, has low trend importance and high periodicity importance. The table shows that for the low-low quadrant, the persistence baseline often has the lowest error. When there is high periodicity importance in both the low-high and the high-high quadrants, our repeated single-output ensemble proves



(a) Nemenyi plot showing our methods compared to the baseline methods using RMSE over 58 countries tested in 2020.



(b) Nemenyi plot showing our methods compared to the baseline methods using RMSE over 26 countries tested in 2021.

**FIGURE 4.** Nemenyi plot showing the comparative performance of our methods compared to the baseline methods using RMSE. None of the methods use drift adaptation techniques. A lower rank depicts better performance. When methods are linked with a horizontal bar, they are within critical distance, meaning there is no significant difference between average ranks. *M* and *B* prefixes denote our methods and baselines.

to be quite strong. In the high-low quadrant, there is no clear winner.

Zooming in on countries within these quadrants, we can see what patterns our methods are capable of capturing. We show that the countries of Estonia, Sweden, Switzerland and the United States as respectively an example of irregular patterns in the low-low quadrant, an important trend in the high-low quadrant, clear cycles in the low-high quadrant and a combination of an important trend and clear cycles in the high-high quadrant (similar graphs of all countries are available online<sup>2</sup>). For Estonia (Figure 5), we see that none of the baselines is able to capture the unsteady pattern of the true data. Our multi-output ensembles using mobility data also have difficulty here, but they get closer than the baselines. The repeated single-output ensemble predicts a rising trend with cycles, with rising uncertainty as time progresses. For Sweden (Figure 6), an example with high trend importance, all methods are performing worse than persistence. Our multi-output ensembles predict the rising trend too weakly and the downward trend too late. The deep learning baselines estimate an upward trend where in reality, it drops later. The repeated single-output ensemble predicts the first few days closely but gets eluded most when true observations drop. The persistence baseline takes an average position. To review a case with high cycles, we show the forecasts for Switzerland (Figure 7). This country is grouped in the high-low quadrant, with high importance of periodicity and low importance of trend. We see that most of our methods only slightly capture the periodic pattern of the data, except for the repeated single-output method, where the prediction is much better. The deep learning methods are able to predict some periodicity but

do so too low. Finally, to show a combination of trend and cycles, we show the results for the United States of America in Figure 8. This shows a similar situation as the low-high quadrant, where periodic patterns are somewhat captured by most methods and baselines but not as strong as the repeated single-output ensemble. In cases like these, we see that the persistence baseline can be difficult to beat if the observations on the day before the test period are close to the average of the true observations later.

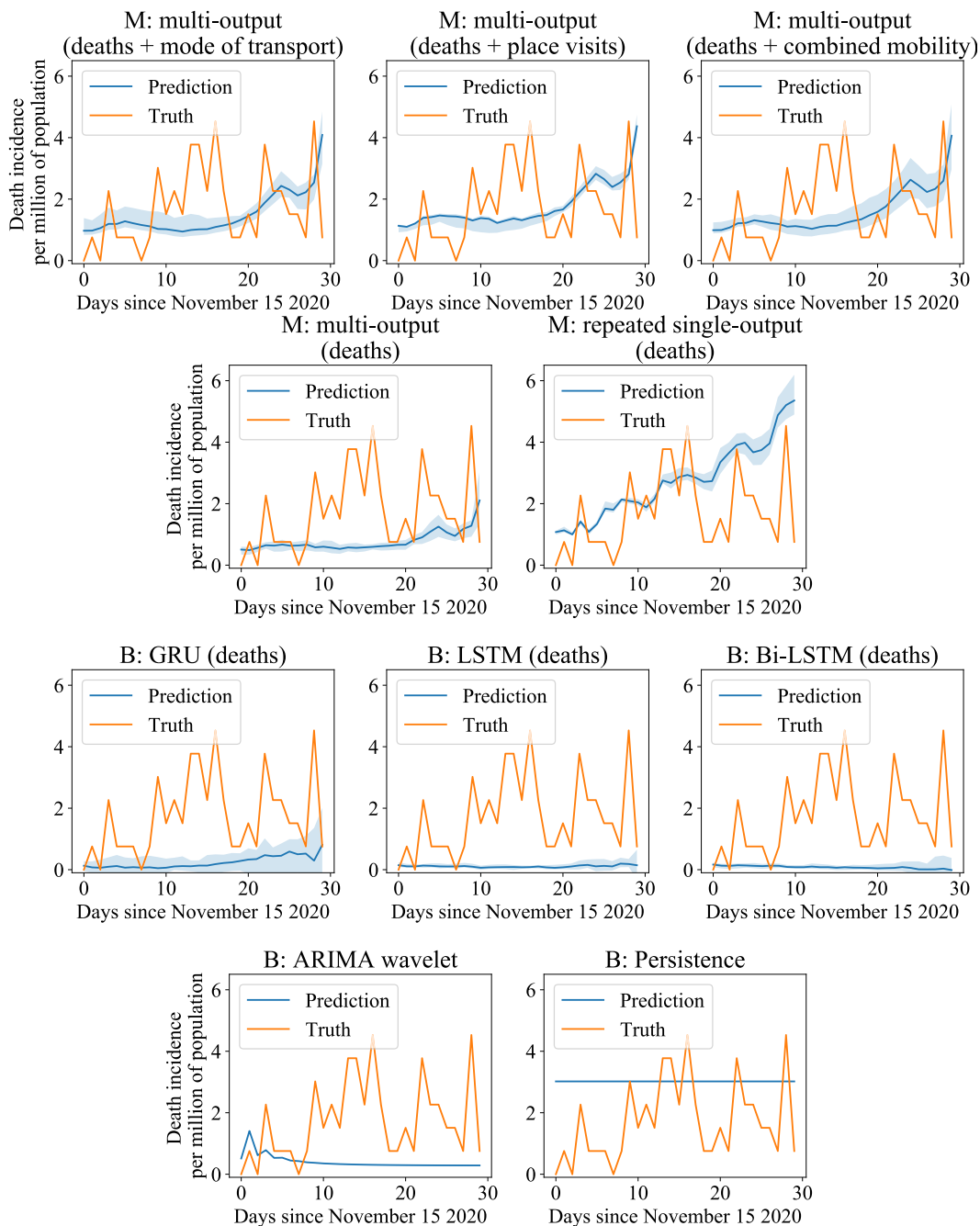
This shows that compared to other baselines, our repeated single-output ensemble and the multi-output ensembles using place visit mobility data are quite strong in the 2020 scenario. While the persistence baseline outperforms for 20 of the 58 countries, it fails with time series data that exhibits strong patterns of periodicity or trends. The other baselines perform worse than our methods. Our repeated single-output ensemble is strong when cycles are apparent but fails when the true observations suddenly change. In the 2021 scenario, all baselines are performing better than our methods. Our methods are not adapted to the concept drift in this scenario. Due to the change in the normalising factor, old patterns learned may obfuscate the new ones. We demonstrate how to address this using the concept drift adaptation techniques mentioned in Section IV-C.

#### D. Q3: DRIFT ADAPTATION

We aim to understand if adapting for concept drift helps in improving COVID-19 forecasting accuracy using this AutoML approach. The answer of Q2 showed that our methods performed worse than the baselines in 2021, while they were better than most in 2020. This may be a result of concept drift. This section shows the results of our experiments adapting our methods to this drift.

<sup>2</sup><https://github.com/AutoML4covid19/Forecasting>

### 2020 pandemic forecast for Estonia



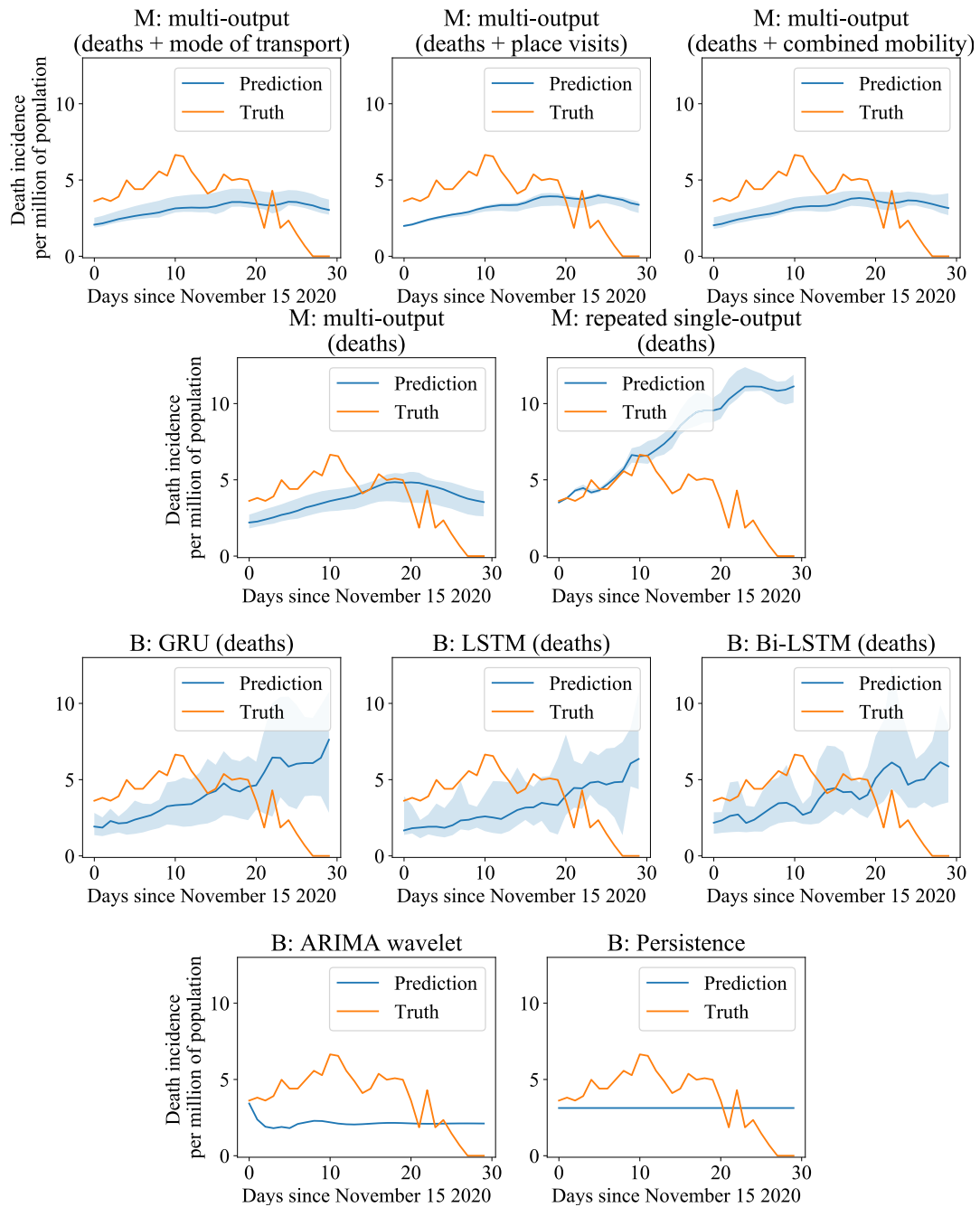
**FIGURE 5.** Forecasts of the different methods for Estonia in the 2020 scenario. *M* and *B* prefixes denote our methods and baselines.

In our experiments, the simpler baselines – persistence and the ARIMA wavelet – use only the new data for their predictions. The persistence baseline uses just the last observation seen before the start of the test period, and the ARIMA wavelet baseline relies on an assumption of no missing values. As there is a gap between datasets, this is not the case for the second scenario. To be fair, using the deep learning baselines compared non-adapted models with models retrained

on the new data. As Figure 10a shows, the retraining was detrimental to their performance. Therefore, in the subsequent comparisons, we thus only consider the deep learning baselines using the full dataset.

As we can only effectively adapt for drift in the 2021 scenario due to a lack of a drift detection mechanism, we show only results for 2021 in this section. We compare all drift adaptation strategies previously introduced in

### 2020 pandemic forecast for Sweden

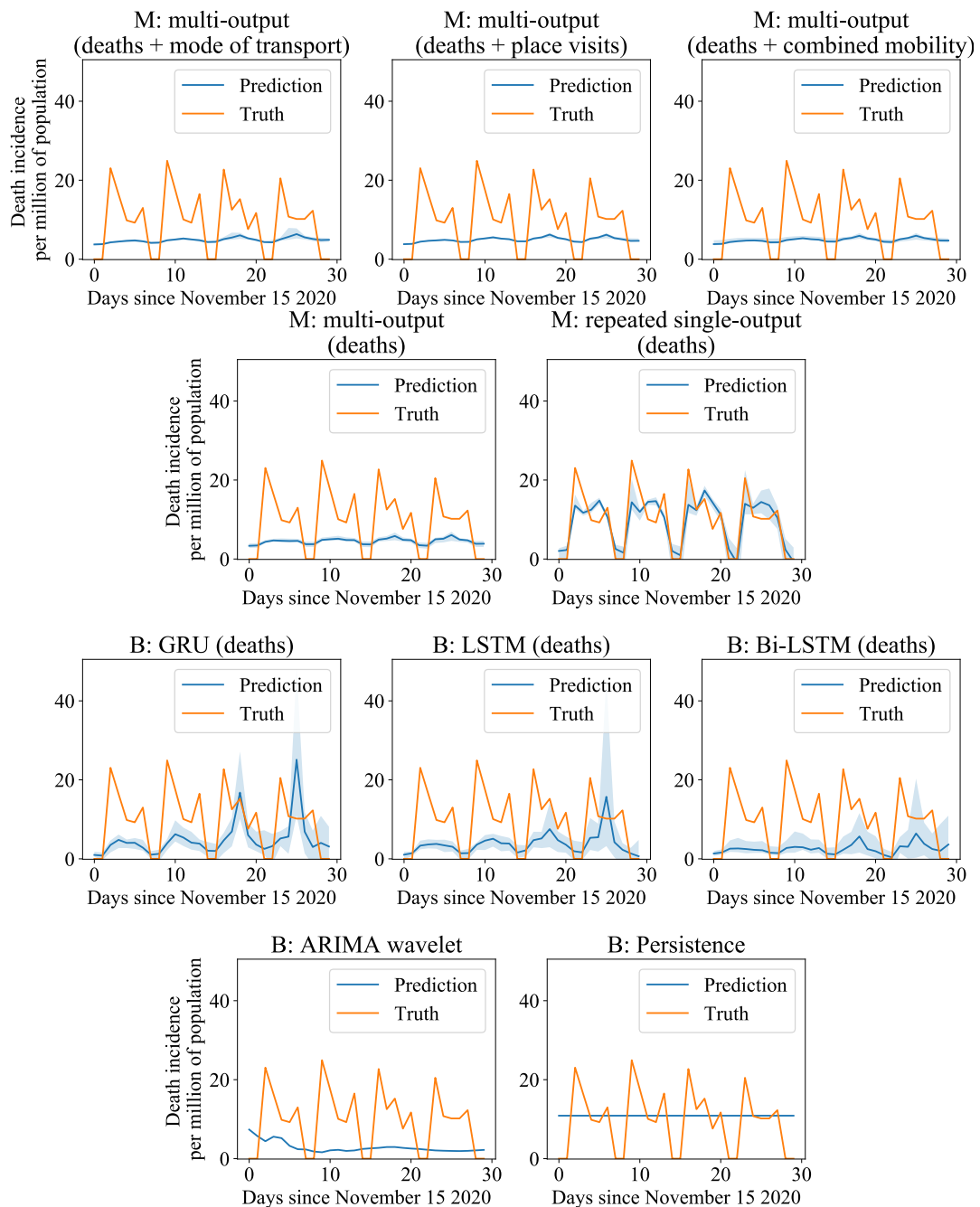


**FIGURE 6.** Forecasts of the different methods for Sweden in the 2020 scenario. *M* and *B* prefixes denote our methods and baselines.

Section IV-C for all of our ensembles and use different mobility data sources in Figure 10b. This leads to 20 combinations considering different mobility sources, adaptation strategies and forecasting approaches. This figure shows some distinguishable groups of methods. The best methods are all multi-output ensembles adapted using the partial refit strategy. The best two of this group – the ensemble using place visits mobility features and the ensemble using combined mobility features – outperform all methods using different

adaptation strategies on a significant level. The next group of methods consist mainly of the multi-output ensemble using only mortality features. For this ensemble, changes in performance with different drift adaptation strategies are smaller than for the ensemble using mobility features, but a partial refit still yields the best performance. The last group consists of multi-output methods using mobility features and drift adaptation strategies other than the partial refit strategy. These strategies do not go well together. The repeated single-output

### 2020 pandemic forecast for Switzerland



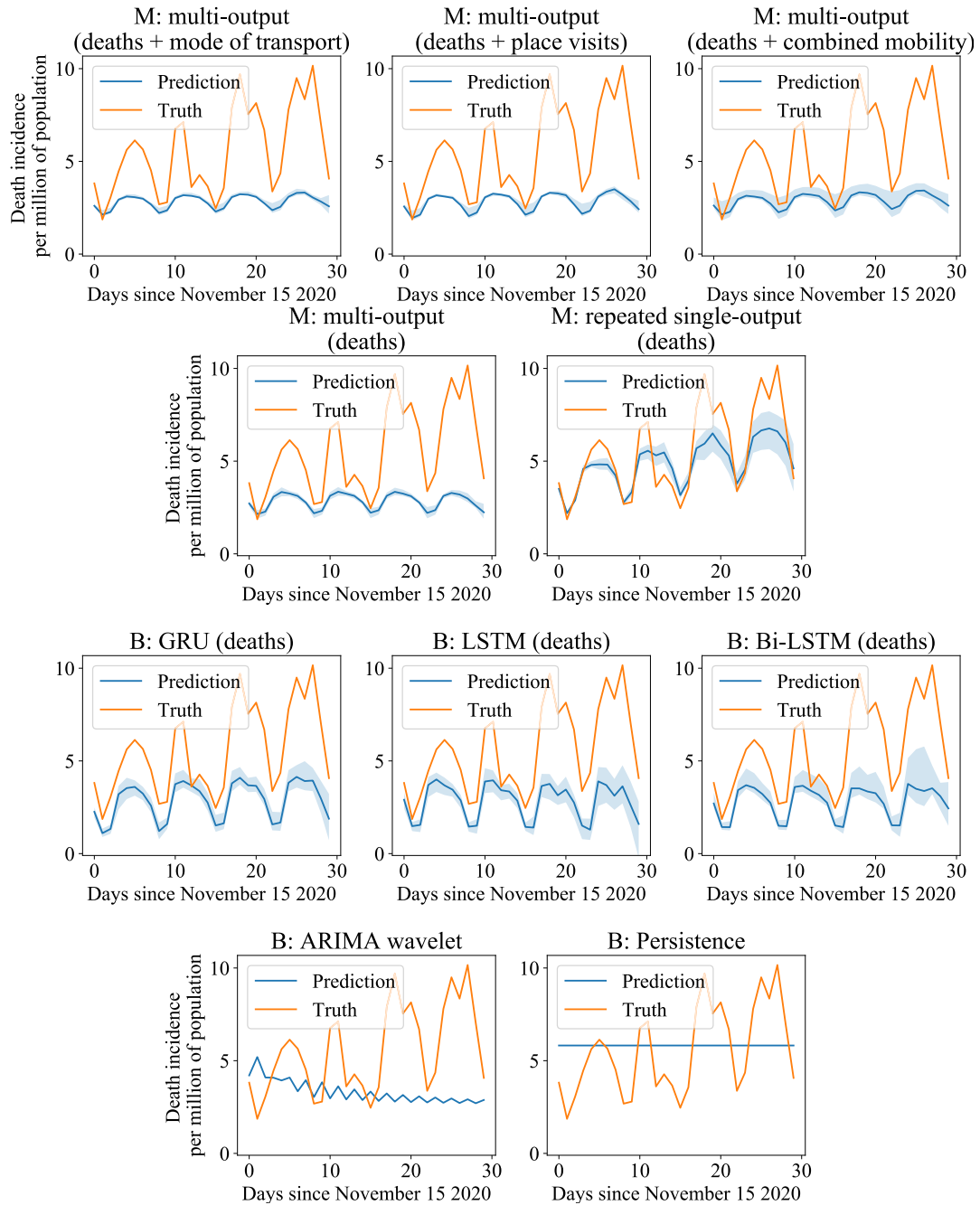
**FIGURE 7.** Forecasts of the different methods for Switzerland in the 2020 scenario. *M* and *B* prefixes denote our methods and baselines.

ensemble is the only method that does not improve by adapting to drift. The non-adapted version of this approach is significantly better than all its adapted counterparts. Still, its performance is ranked worse than all other partial refit methods.

We also compare the ensembles using the partial refit drift adaptation strategy with the baselines in Figure 10c. This figure shows that all multi-output ensembles using the partial

refit strategy outperform all baselines. In this scenario, the ARIMA wavelet baseline is the strongest but performs significantly worse than the multi-output ensembles using place visits mobility data or combined mobility data. The deep learning methods are in the same group as the persistence and ARIMA wavelet baseline and are within a critical distance of the partial refit multi-output ensemble using mortality data. However, they are all significantly outperformed by all

### 2020 pandemic forecast for United States



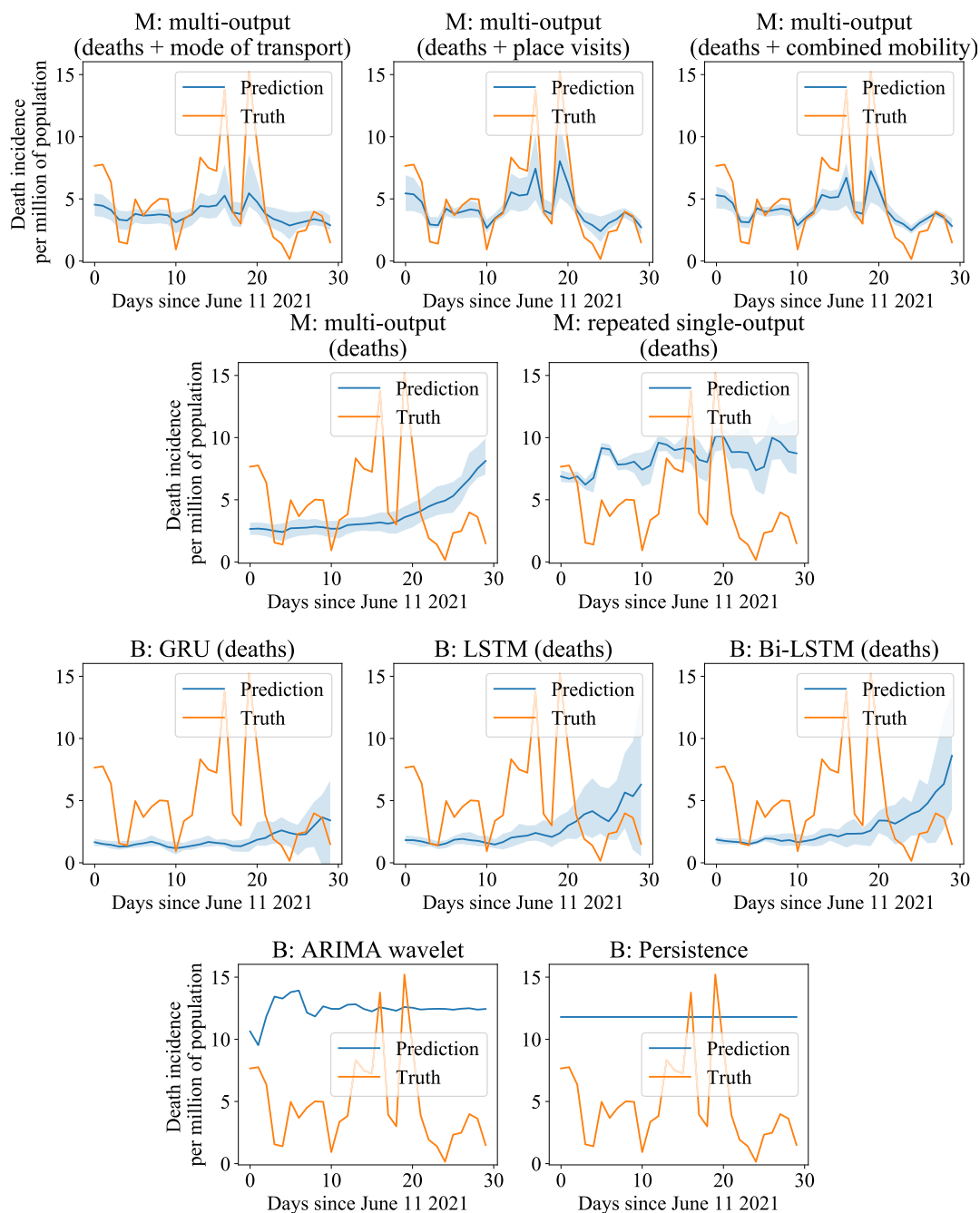
**FIGURE 8.** Forecasts of the different methods for the united states of America in the 2020 scenario. *M* and *B* prefixes denote our methods and baselines.

multi-output ensembles using mobility features. We show the RMSE of the baselines and our methods using the partial refit drift adaptation strategy for all countries separately in Table 4. This table shows that the multi-output ensemble using place visits features has a lower RMSE than all other methods and baselines for 22 of the 26 countries and lower RMSE than the strongest baseline for the other four countries. To give a notion of the quality of forecasts of the adapted methods,

we show that the country of Romania in Figure 9, grouped in the low-low quadrant with irregular true observations in the test set but some indication of trend and periodicity. The sudden drops and spikes are quite difficult to anticipate for all baselines, as well as for our methods not using mobility features. The ensembles using these features, however, while not exactly predicting the magnitude of the extreme values, can predict where spikes and drops will occur.



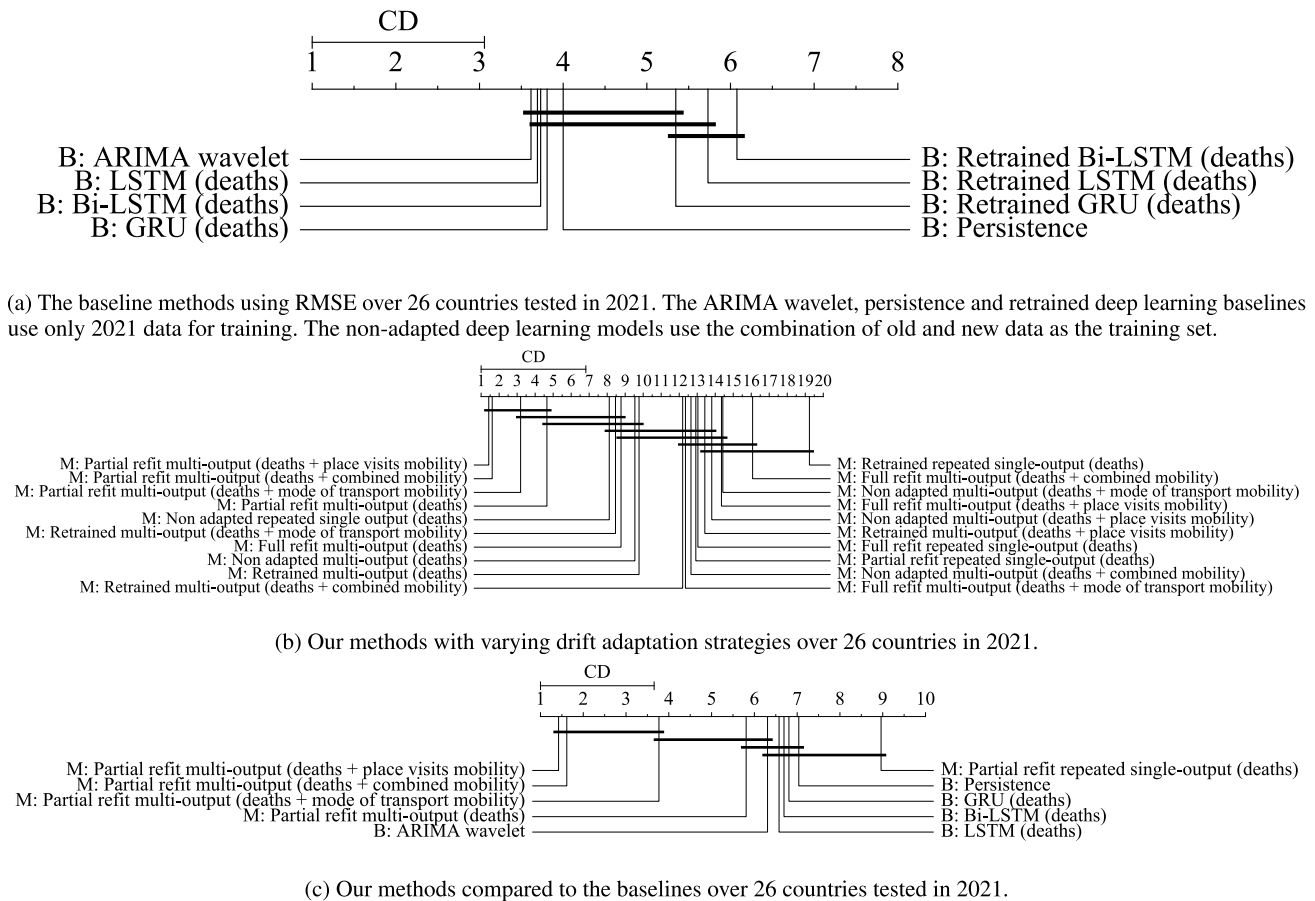
### 2021 pandemic forecast for Romania



**FIGURE 9.** Forecasts of the different methods for Romania in the 2021 scenario. *M* and *B* prefixes denote our methods and baselines.

Drift adaptation may seem a lot more impactful for our AutoML-based approaches than for baselines. We are performing hyperparameter optimisation to ensure the best models are configured on the provided training data. However, as the concept changes, this approach will lead to a model that over-fits the older part of the data. Consequently, this approach performs much worse on new data compared to baselines with average performance on all data.

This experiment has shown that adapting to concept drift can indeed help to improve the accuracy of COVID-19 forecasts using an AutoML approach. This is specifically the case for our multi-output ensembles using the partial refit strategy. This strategy entails keeping the ensembles trained using the old dataset but updating the model weights using the new data. This way, old knowledge is used, but the emphasis is placed on the newer data. This strategy



**FIGURE 10.** Nemenyi plots showing the comparative performance of baselines and our methods when exposed to drift based on RMSE. Methods on the left have a lower average rank and are thus comparatively better than methods on the right. When methods are linked with a horizontal bar, they are within critical distance, meaning there is no significant difference between average ranks. *M* and *B* prefixes denote our methods and baselines.

works especially well when combined with mobility data features.

**VII. LIMITATION OF THE STUDY**

We found that when the pandemic is still novel, our methods are outperformed by baselines as simple as persistence. However, when the pandemic has progressed for just shy of a year in many countries, our ensembles are on par with the best baselines. Even later, when concept drift occurs due to a shift in data normalisation and possibly mutation of the virus, our methods significantly outperform the baselines, especially when using mobility data along with mortality data. Our work has shown that our modified version of auto-sklearn does not perform as well as simple baselines within the first few months of the pandemic but gains importance as time progresses. After a little less than a year, we have gained enough data to be able to capture most cycles and trends occurring in the time series. Only when trends suddenly change are our predictions eluded. Additionally, we discovered that when concept drift occurs by a change in data normalisation or possibly a mutation of the virus, refitting the models trained on the older data enables a major performance boost, especially when (unchanged) mobility data is used alongside the mortality data.

Another limitation of our work is that the best moments to adapt the ensembles over time are not detected automatically. Current AutoML systems use large batches of data at the same time to train their models. If these batches are too large, however, chances are the concept drift slips in undetected. A proper trade-off should be made between how much data is used in order to learn the data patterns sufficiently and to be able to detect concept drift within the used data. Future work can address this issue further.

Finally, we want to note that due to a lack of availability of the COVID-19 mortality data, we were only able to use the countries in Europe for our scenario in 2021. For the countries outside of Europe that were used in the 2020 scenario, we were thus not able to test the drift adaptation strategies. It would be interesting to see whether or not the partial adaptation improves forecasts consistently for these countries as well.

**VIII. CONCLUSION**

In this work, we adapted the AutoML framework of auto-sklearn to COVID-19 forecasting. We used mortality data and mobility data collected from 26 European countries to construct automatically configured ensembles of regression models. We compared the performance of a multi-output

**TABLE 3. RMSE on 2020 forecasts. Countries are grouped by quadrant: Low trend - low periodicity, low trend - high periodicity, high trend - low periodicity and high trend - high periodicity. Our methods are abbreviated to MO for multi-output and RSO for repeated single-output. *d* denotes mortality data, *mtm* mode of transport mobility, *pvm* place visits mobility and *cm* combined mobility. Our methods and baselines are denoted with *M* and *B*, respectively. All Baselines only use mortality data.**

Country	Trend	Periodicity	M: MO (d+mtm)	M: MO (d+pvm)	M: MO (d+cm)	M: MO (d)	M: RSO (d)	B: GRU	B: LSTM	B: Bi-LSTM	B: Persistence	B: ARIMA wavelnet
Cambodia	0	0	0.32 ± 0.26	0.13 ± 0.11	0.11 ± 0.11	0.13 ± 0.5	1.081 ± 0.47	1.527 ± 0.69	1.834 ± 0.39	1.413 ± 0.93	0.0 ± 0.0	0.0 ± 0.0
New Zealand	0	0	0.214 ± 0.09	0.164 ± 0.03	0.24 ± 0.16	0.183 ± 0.58	0.312 ± 0.13	0.443 ± 0.2	0.532 ± 0.11	0.468 ± 0.27	0.0 ± 0.0	0.044 ± 0.0
Taiwan	0	0	1.334 ± 1.16	1.368 ± 0.21	1.952 ± 0.7	1.092 ± 1.59	1.566 ± 0.68	2.203 ± 1.0	2.644 ± 0.57	2.326 ± 1.34	0.0 ± 0.0	0.0 ± 0.0
Thailand	0	0	1.189 ± 2.79	0.789 ± 0.76	0.798 ± 1.8	0.858 ± 1.34	4.584 ± 1.98	9.203 ± 3.22	8.832 ± 1.61	7.284 ± 4.25	0.0 ± 0.0	0.069 ± 0.0
Vietnam	0	0	16.95 ± 5.45	21.704 ± 2.11	17.564 ± 4.07	0.929 ± 2.22	6.349 ± 2.77	8.937 ± 4.04	10.728 ± 2.3	9.438 ± 5.44	0.0 ± 0.0	0.027 ± 0.0
Philippines	0	0.065	31.8 ± 2.01	34.37 ± 1.48	37.733 ± 4.55	44.848 ± 3.75	25.579 ± 0.95	29.911 ± 5.3	36.293 ± 13.87	31.069 ± 15.76	<b>24.354 ± 0.0</b>	24.766 ± 0.0
Malaysia	0	0.094	3.094 ± 0.49	<b>3.003 ± 0.2</b>	3.29 ± 0.42	7.388 ± 2.23	5.212 ± 1.18	6.042 ± 0.79	5.501 ± 1.95	5.008 ± 1.65	1.507 ± 0.0	3.045 ± 0.0
Colombia	0	0.177	18.422 ± 5.74	18.461 ± 2.12	19.391 ± 3.42	28.735 ± 5.14	37.756 ± 9.76	36.469 ± 11.08	65.748 ± 27.6	72.034 ± 30.26	<b>13.733 ± 0.0</b>	20.206 ± 0.0
United Arab Emirates	0	0.192	<b>1.542 ± 0.03</b>	1.55 ± 0.04	<b>1.538 ± 0.04</b>	2.784 ± 0.43	2.202 ± 0.27	2.218 ± 0.65	2.482 ± 0.93	2.541 ± 0.6	2.601 ± 0.0	1.707 ± 0.0
Luxembourg	0	0.393	4.64 ± 0.09	4.578 ± 0.06	4.596 ± 0.11	4.575 ± 0.14	<b>3.139 ± 0.04</b>	5.798 ± 0.25	5.699 ± 0.42	5.776 ± 0.39	3.459 ± 0.9	5.942 ± 0.0
Ireland	0.01	0.396	4.12 ± 0.04	4.107 ± 0.06	4.106 ± 0.04	5.096 ± 0.44	<b>3.941 ± 0.04</b>	3.952 ± 0.28	4.206 ± 0.26	4.115 ± 0.2	4.767 ± 0.0	5.388 ± 0.0
Norway	0.012	0.24	5.268 ± 0.28	5.214 ± 0.03	5.229 ± 0.09	5.275 ± 0.06	5.083 ± 0.14	5.51 ± 0.16	5.547 ± 0.06	5.574 ± 0.07	<b>5.023 ± 0.0</b>	4.767 ± 0.0
Mexico	0.017	0.493	254.896 ± 6.76	261.954 ± 6.74	254.052 ± 7.02	271.527 ± 9.76	<b>220.226 ± 5.88</b>	269.197 ± 19.2	273.764 ± 16.14	257.575 ± 19.03	266.958 ± 0.0	280.753 ± 0.0
Slovakia	0.03	0.416	11.894 ± 0.33	11.693 ± 0.21	11.681 ± 0.21	11.008 ± 0.35	28.148 ± 1.18	15.386 ± 2.81	16.646 ± 1.31	18.468 ± 3.21	11.567 ± 0.0	14.947 ± 0.0
Ukraine	0.086	0.416	113.98 ± 16.53	108.103 ± 9.72	117.986 ± 9.88	104.972 ± 3.33	99.257 ± 8.93	89.113 ± 4.4	<b>77.413 ± 4.69</b>	77.417 ± 4.37	89.445 ± 0.0	115.374 ± 0.0
Uruguay	0.086	0.342	0.964 ± 0.06	0.901 ± 0.02	0.95 ± 0.07	1.738 ± 0.32	1.229 ± 0.15	0.861 ± 0.14	0.892 ± 0.05	0.911 ± 0.05	<b>0.816 ± 0.0</b>	0.987 ± 0.0
Australia	0.11	0.231	4.64 ± 1.64	3.726 ± 1.31	4.139 ± 1.59	1.745 ± 0.46	1.653 ± 0.69	1.386 ± 0.65	1.337 ± 0.32	0.768 ± 0.33	<b>0.183 ± 0.0</b>	1.169 ± 0.0
Singapore	0.15	0.206	0.189 ± 0.02	0.184 ± 0.0	0.213 ± 0.04	0.188 ± 0.05	0.449 ± 0.14	0.437 ± 0.09	0.404 ± 0.06	0.372 ± 0.22	<b>0.183 ± 0.0</b>	0.185 ± 0.0
Israel	0.202	0.399	10.061 ± 0.37	10.285 ± 0.2	9.887 ± 1.11	10.359 ± 0.48	<b>9.258 ± 0.17</b>	10.796 ± 0.73	10.798 ± 0.49	10.45 ± 0.47	9.265 ± 0.0	13.32 ± 0.0
Lithuania	0.241	0.447	10.761 ± 0.4	9.477 ± 0.36	9.986 ± 0.62	11.932 ± 0.57	<b>6.71 ± 0.26</b>	16.711 ± 1.09	14.668 ± 1.53	12.125 ± 1.12	12.863 ± 0.0	17.772 ± 0.0
Morocco	0.245	0.347	52.153 ± 7.68	62.771 ± 4.61	53.216 ± 6.98	45.164 ± 6.52	45.571 ± 8.02	25.649 ± 4.24	32.99 ± 6.96	29.208 ± 7.99	<b>22.927 ± 0.0</b>	23.58 ± 0.0
Portugal	0.37	0.192	38.94 ± 2.3	38.56 ± 0.33	39.235 ± 2.85	37.643 ± 2.97	27.444 ± 4.15	20.63 ± 2.33	23.273 ± 5.09	24.611 ± 2.47	<b>12.823 ± 0.0</b>	58.075 ± 0.0
Latvia	0.408	0.079	5.365 ± 0.15	5.079 ± 0.13	5.357 ± 0.17	6.036 ± 0.33	<b>4.334 ± 0.14</b>	7.384 ± 0.78	8.406 ± 0.23	8.402 ± 0.22	5.301 ± 0.0	8.057 ± 0.0
Estonia	0.445	0.212	<b>1.97 ± 0.06</b>	<b>1.878 ± 0.04</b>	1.971 ± 0.1	2.135 ± 0.07	2.206 ± 0.19	2.704 ± 0.16	2.734 ± 0.05	2.795 ± 0.04	2.415 ± 0.0	2.538 ± 0.0
South Korea	0.455	0.369	21.509 ± 1.4	30.694 ± 2.55	26.501 ± 6.29	2.682 ± 0.94	6.6 ± 1.77	2.582 ± 0.53	2.631 ± 0.32	2.62 ± 0.5	<b>2.456 ± 0.0</b>	2.666 ± 0.0
Chile	0.027	0.805	<b>27.169 ± 2.22</b>	<b>28.242 ± 2.13</b>	<b>30.626 ± 1.79</b>	<b>22.198 ± 0.82</b>	<b>21.631 ± 1.72</b>	<b>19.341 ± 4.56</b>	<b>23.722 ± 2.98</b>	<b>24.41 ± 3.59</b>	22.029 ± 0.0	28.134 ± 0.0
United Kingdom	0.07	0.853	251.505 ± 16.72	258.507 ± 12.04	251.626 ± 11.91	230.949 ± 13.49	<b>109.637 ± 14.68</b>	132.277 ± 8.29	140.575 ± 9.42	207.986 ± 123.94	171.946 ± 0.0	165.801 ± 0.0
Romania	0.079	0.692	69.74 ± 3.57	69.141 ± 2.29	70.845 ± 3.24	70.845 ± 3.24	50.042 ± 13.85	48.347 ± 3.75	48.273 ± 3.38	52.004 ± 3.65	<b>31.17 ± 0.0</b>	85.367 ± 0.0
Switzerland	0.124	0.961	76.182 ± 0.91	75.478 ± 0.96	76.424 ± 1.0	76.266 ± 1.14	<b>45.287 ± 1.61</b>	79.577 ± 6.78	79.954 ± 5.4	86.021 ± 5.29	68.277 ± 0.0	89.256 ± 0.0
Spain	0.18	0.763	160.769 ± 6.49	158.399 ± 5.52	165.438 ± 6.7	<b>152.177 ± 7.53</b>	195.637 ± 42.67	256.755 ± 23.31	202.185 ± 22.46	225.704 ± 26.23	297.479 ± 0.0	306.011 ± 0.0
Finland	0.253	0.655	3.705 ± 0.08	3.613 ± 0.08	3.67 ± 0.08	<b>3.588 ± 0.05</b>	4.126 ± 0.38	4.376 ± 0.18	4.125 ± 0.08	4.068 ± 0.13	3.907 ± 0.0	2.932 ± 0.0
Brazil	0.307	0.86	488.393 ± 119.58	413.655 ± 55.03	527.556 ± 74.95	225.257 ± 21.69	<b>159.886 ± 4.99</b>	256.533 ± 44.43	177.092 ± 48.35	210.016 ± 63.43	238.741 ± 0.0	226.523 ± 0.0
Austria	0.32	0.506	70.909 ± 1.57	70.909 ± 1.57	68.149 ± 3.19	56.061 ± 3.54	<b>31.384 ± 0.62</b>	61.559 ± 2.18	66.199 ± 2.27	67.219 ± 0.37	47.652 ± 0.0	88.662 ± 0.0
Slovenia	0.329	0.11	31.496 ± 0.27	31.462 ± 0.26	31.084 ± 0.39	31.473 ± 0.56	26.691 ± 0.78	31.76 ± 0.09	32.423 ± 1.5	32.423 ± 1.5	<b>25.371 ± 0.0</b>	36.206 ± 0.0
Italy	0.345	0.657	377.355 ± 13.76	377.614 ± 6.23	377.787 ± 12.51	368.746 ± 18.21	<b>120.112 ± 10.63</b>	337.985 ± 28.13	333.81 ± 26.82	335.793 ± 20.08	180.322 ± 0.0	357.585 ± 0.0
Argentina	0.43	0.522	124.01 ± 12.17	132.72 ± 6.13	113.227 ± 12.98	109.479 ± 20.22	<b>71.224 ± 12.63</b>	81.043 ± 10.77	106.665 ± 7.34	111.14 ± 10.95	104.163 ± 0.0	113.392 ± 0.0
Netherlands	0.482	0.821	19.903 ± 1.05	19.314 ± 1.23	20.167 ± 1.07	19.877 ± 0.97	<b>14.642 ± 1.88</b>	37.3 ± 37.66	33.042 ± 15.65	59.044 ± 33.14	20.926 ± 0.0	22.382 ± 0.0
Turkey	0.542	0.22	<b>47.783 ± 10.72</b>	70.413 ± 8.15	65.061 ± 10.08	85.431 ± 4.68	106.219 ± 6.99	120.463 ± 7.31	105.313 ± 5.26	99.52 ± 6.44	85.707 ± 0.0	98.375 ± 0.0
Denmark	0.584	0.331	1.15 ± 1.77	1.163 ± 0.79	4.928 ± 1.34	<b>2.611 ± 0.19</b>	5.614 ± 0.37	4.401 ± 0.51	3.695 ± 0.59	4.033 ± 0.49	5.908 ± 0.0	4.155 ± 0.0
Croatia	0.656	0.173	35.543 ± 0.76	33.915 ± 0.68	35.118 ± 1.28	35.618 ± 1.08	21.968 ± 1.29	37.474 ± 3.13	41.186 ± 3.02	39.38 ± 2.02	<b>18.214 ± 0.0</b>	44.293 ± 0.0
Indonesia	0.688	0.11	42.742 ± 3.47	39.192 ± 2.04	41.117 ± 4.13	<b>29.683 ± 2.3</b>	91.937 ± 22.59	33.238 ± 4.93	33.08 ± 13.98	64.922 ± 18.61	33.883 ± 0.0	33.883 ± 0.0
Greece	0.724	0.246	43.401 ± 2.23	33.947 ± 2.59	38.571 ± 4.3	49.063 ± 3.85	<b>26.908 ± 3.79</b>	56.176 ± 3.47	55.67 ± 3.92	56.811 ± 4.66	52.971 ± 0.0	78.198 ± 0.0
Saudi Arabia	0.768	0.046	4.823 ± 1.16	4.237 ± 1.16	<b>3.655 ± 1.6</b>	12.391 ± 1.52	<b>2.178 ± 2.59</b>	14.173 ± 5.02	16.871 ± 4.64	27.705 ± 10.69	6.483 ± 0.0	4.576 ± 0.0
Egypt	0.781	0.214	105.698 ± 23.23	156.058 ± 18.68	148.257 ± 62.07	7.951 ± 4.62	19.453 ± 5.5	6.364 ± 1.19	<b>3.692 ± 1.43</b>	4.528 ± 0.85	<b>6.199 ± 0.0</b>	4.832 ± 0.0
Hungary	0.847	0.401	84.779 ± 2.38	86.261 ± 1.97	85.435 ± 1.53	103.592 ± 3.04	64.936 ± 3.55	87.224 ± 3.33	97.269 ± 5.26	97.306 ± 3.68	<b>45.008 ± 0.0</b>	98.564 ± 0.0
Belgium	0.883	0.3	49.054 ± 2.04	47.665 ± 2.93	49.351 ± 4.44	48.943 ± 6.17	<b>18.248 ± 0.73</b>	53.395 ± 3.69	58.499 ± 5.11	69.618 ± 6.77	76.276 ± 0.0	46.515 ± 0.0
Sweden	0.93	0.97	22.359 ± 0.81	22.039 ± 0.57	21.747 ± 0.45	21.728 ± 1.04	58.301 ± 1.72	32.616 ± 4.85	32.821 ± 3.0	29.965 ± 4.84	<b>20.231 ± 0.0</b>	25.393 ± 0.0
Poland	0.544	0.964	295.172 ± 6.01	292.461 ± 5.9	296.392 ± 5.24	323.275 ± 7.28	<b>176.458 ± 10.86</b>	249.192 ± 12.89	293.259 ± 22.52	258.842 ± 20.09	177.655 ± 0.0	336.678 ± 0.0
France	0.614	0.939	318.124 ± 5.07	307.682 ± 1.77	317.085 ± 10.58	314.669 ± 7.31	<b>246.63 ± 10.1</b>	348.501 ± 23.55	322.253 ± 21.44	318.322 ± 15.58	549.614 ± 0.0	340.719 ± 0.0
Canada	0.641	0.619	34.69 ± 2.4	35.479 ± 2.64	39.937 ± 6.61	30.814 ± 9.16	<b>22.733 ± 5.28</b>	32.413 ± 5.61	25.238 ± 2.88	28.127 ± 6.68	34.042 ± 0.0	38.337 ± 0.0
Czechia	0.667	0.7	55.133 ± 4.78	51.139 ± 4.7	54.091 ± 3.25	74.659 ± 6.59	<b>28.585 ± 1.99</b>	63.597 ± 2.87	71.5 ± 3.33	76.238 ± 3.98	69.95 ± 0.0	118.404 ± 0.0
Bulgaria	0.692	0.961	82.844 ± 2.64	75.533 ± 2.66	75.813 ± 3.9	93.925 ± 2.1	66.24 ± 1.58	92.185 ± 2.88	99.703 ± 3.4	97.31 ± 3.2	<b>51.91 ± 0.0</b>	48.458 ± 0.0
South Africa	0.702	0.562	41.61 ± 1.38	<b>40.015 ± 2.25</b>	40.762 ± 1.02	54.057 ± 3.66	64.248 ± 4.18	83.199 ± 8.97	82.81 ± 6.24	76.28 ± 4.3	52.798 ± 0.0	51.02 ± 0.0
Japan	0.761	0.538	15.924 ± 1.66	15.123 ± 0.91	13.146 ± 1.41	14.559 ± 10.23	31.725 ± 6.67	<b>12.679 ± 2.21</b>	13.493 ± 1.53	16.485 ± 2.61	16.485 ± 2.61	19.981 ± 0.0
India	0.763	0.066	421.055 ± 101.89	473.874 ± 25.56	516.055 ± 53.29	116.945 ± 113.52	156.634 ± 40.38	250.148 ± 153.11	220.584 ± 150.57	346.538 ± 121.74	76.175 ± 0.0	<b>72.221 ± 0.0</b>
United States	0.776	0.856	1128.432 ± 16.48	1122.445 ± 16.34	1116.084 ±							

drift occurs, due to a shift in data normalisation and possibly virus mutations, it is necessary to incorporate concept drift adaptation techniques into our AutoML methods in order to obtain useful predictions. When adapted, our multi-output methods using mobility data significantly outperform the baselines we have considered in our study.

Our best-performing ensembles utilised the concept drift adaptation strategy of refitting the ensembles once drift has occurred. Automatically, finding the best moments to adapt the ensembles over time is an interesting direction for future research.

## APPENDIX

See Tables 3 and 4.

## REFERENCES

- [1] World Health Organization. (2020). *Statement on the First Meeting of the International Health Regulations (2005) Emergency Committee Regarding the Outbreak of Novel Coronavirus (2019-NCoV)*. [Online]. Available: <https://www.who.int/>
- [2] ECDC. (2021). *European Centre for Disease Prevention and Control*. [Online]. Available: <https://www.ecdc.europa.eu/en>
- [3] J. P. Ioannidis, S. Cripps, and M. A. Tanner, "Forecasting for COVID-19 has failed," *Int. J. Forecasting*, vol. 38, pp. 423–438, Aug. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169207020301199>
- [4] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2962–2970.
- [5] H. Jin, Q. Song, and X. Hu, "Auto-Keras: An efficient neural architecture search system," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, 2019, pp. 1946–1956, doi: [10.1145/3292500.3330648](https://doi.org/10.1145/3292500.3330648).
- [6] C. Thornton, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA, Aug. 2013, pp. 847–855, doi: [10.1145/2487575.2487629](https://doi.org/10.1145/2487575.2487629).
- [7] S. Flaxman, S. Mishra, A. Gandy, H. J. T. Unwin, T. A. Mellan, H. Coupland, C. Whittaker, H. Zhu, T. Berah, J. W. Eaton, and M. Monod, "Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe," *Nature*, vol. 584, no. 7820, pp. 257–261, 2020, doi: [10.1038/s41586-020-2405-7](https://doi.org/10.1038/s41586-020-2405-7).
- [8] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons Fractals*, vol. 135, Jun. 2020, Art. no. 109864. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077920302642>
- [9] T. Chakraborty and I. Ghosh, "Real-time forecasts and risk assessment of novel coronavirus (COVID-19) cases: A data-driven analysis," *Chaos, Solitons Fractals*, vol. 135, Jun. 2020, Art. no. 109850. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077920302502>
- [10] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proc. Roy. Soc. London A, Math. Phys. Sci.*, vol. 115, no. 772, pp. 700–721, 1927. [Online]. Available: <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1927.0118>
- [11] K. L. Cooke, "Stability analysis for a vector disease model," *Rocky Mountain J. Math.*, vol. 9, no. 1, pp. 31–42, 1979. [Online]. Available: <http://www.jstor.org/stable/44238836>
- [12] Q.-H. Liu, M. Ajelli, A. Aleta, S. Merler, Y. Moreno, and A. Vespignani, "Measurability of the epidemic reproduction number in data-driven contact networks," *Proc. Nat. Acad. Sci. USA*, vol. 115, no. 50, pp. 12680–12685, Dec. 2018. [Online]. Available: <https://www.pnas.org/content/115/50/12680>
- [13] D. Balcan, V. Colizza, B. Gonçalves, H. Hu, J. J. Ramasco, and A. Vespignani, "Multiscale mobility networks and the spatial spreading of infectious diseases," *Proc. Nat. Acad. Sci. USA*, vol. 106, no. 51, pp. 21484–21489, Dec. 2009. [Online]. Available: <https://www.pnas.org/content/106/51/21484>
- [14] L. Danon, A. P. Ford, T. House, C. P. Jewell, M. J. Keeling, G. O. Roberts, J. V. Ross, and M. C. Vernon, "Networks and the epidemiology of infectious disease," *Interdiscipl. Perspect. Infectious Diseases*, vol. 2011, p. 284909, Jan. 2011.
- [15] A. Aleta, D. Martín-Corral, A. Pastore Y Piontti, M. Ajelli, M. Litvinova, M. Chinazzi, N. E. Dean, M. E. Halloran, I. M. Longini, Jr., S. Merler, A. Pentland, A. Vespignani, E. Moro, and Y. Moreno, "Modelling the impact of testing, contact tracing and household quarantine on second waves of COVID-19," *Nature Hum. Behav.*, vol. 4, no. 9, pp. 964–971, Sep. 2020, doi: [10.1038/s41562-020-0931-9](https://doi.org/10.1038/s41562-020-0931-9).
- [16] P. Kumar, H. Kalita, S. Patariya, Y. D. Sharma, C. Nanda, M. Rani, J. Rahmani, and A. S. Bhagavathula, "Forecasting the dynamics of COVID-19 pandemic in top 15 countries in April 2020: ARIMA model with machine learning approach," *MedRxiv*, Jan. 2020.
- [17] S. I. Alzahrani, I. A. Aljamaan, and E. A. Al-Fakih, "Forecasting the spread of the COVID-19 pandemic in Saudi Arabia using ARIMA prediction model under current public health interventions," *J. Infection Public Health*, vol. 13, no. 7, pp. 914–919, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1876034120304937>
- [18] Y. Wu, Y. Yang, H. Nishiura, and M. Saitoh, "Deep learning for epidemiological predictions," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, New York, NY, USA, 2018, pp. 1085–1088, doi: [10.1145/3209978.3210077](https://doi.org/10.1145/3209978.3210077).
- [19] E. L. Aiken, A. T. Nguyen, C. Viboud, and M. Santillana, "Toward the use of neural networks for influenza prediction at multiple spatial resolutions," *Sci. Adv.*, vol. 7, no. 25, Jun. 2021. [Online]. Available: <https://www.science.org/doi/abs/10.1126/sciadv.abb1237>
- [20] B. Fu, Y. Yang, Y. Ma, J. Hao, S. Chen, S. Liu, T. Li, Z. Liao, and X. Zhu, "Attention-based recurrent multi-channel neural network for influenza epidemic prediction," in *Proc. IEEE Int. Conf. Bioinf. Biomed. (BIBM)*, Dec. 2018, pp. 1245–1248.
- [21] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-series data: A comparative study," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110121. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S096007792030518X>
- [22] S. Shastri, K. Singh, S. Kumar, P. Kour, and V. Mansotra, "Time series forecasting of Covid-19 using deep learning models: India-USA comparative case study," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110227. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077920306238>
- [23] İ. Kirbaş, A. Sözen, A. D. Tuncer, and F. Kazancıoğlu, "Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches," *Chaos, Solitons Fractals*, vol. 138, Sep. 2020, Art. no. 110015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077920304136>
- [24] F. Shahid, A. Zameer, and M. Muneeb, "Predictions for COVID-19 with deep learning models of LSTM, GRU and Bi-LSTM," *Chaos, Solitons Fractals*, vol. 140, Nov. 2020, Art. no. 110212. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0960077920306081>
- [25] H. Mendoza, A. Klein, M. Feurer, J. T. Springenberg, M. Urban, M. Burkart, M. Dippel, M. Lindauer, and F. Hutter, *Towards Automatically-Tuned Deep Neural Networks*. Cham, Switzerland: Springer, 2019, pp. 135–149, doi: [10.1007/978-3-030-05318-5\\_7](https://doi.org/10.1007/978-3-030-05318-5_7).
- [26] F. Hutter, H. H. Hoos, K. Leyton-Brown, and K. P. Murphy, "An experimental investigation of model-based parameter optimisation: SPO and beyond," in *Proc. 11th Annu. Conf. Genetic Evol. Comput.*, New York, NY, USA, 2009, pp. 271–278, doi: [10.1145/1569901.1569940](https://doi.org/10.1145/1569901.1569940).
- [27] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Learning and Intelligent Optimization*, C. A. C. Coello, Ed. Berlin, Germany: Springer, 2011, pp. 507–523.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, A. Müller, J. Nothman, G. Louppe, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2012.
- [29] R. S. Olson and J. H. Moore, "TPOT: A tree-based pipeline optimization tool for automating machine learning," in *Proc. Workshop Autom. Mach. Learn.*, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., vol. 64. New York, NY, USA, Jun. 2016, pp. 66–74. [Online]. Available: [https://proceedings.mlr.press/v64/olson\\_tpot\\_2016.html](https://proceedings.mlr.press/v64/olson_tpot_2016.html)
- [30] E. LeDell and S. Poirier, "H2O AutoML: Scalable automatic machine learning," in *Proc. AutoML Workshop*, Jul. 2020.

- [31] T. Han, F. N. B. Gois, R. Oliveira, L. R. Prates, and M. M. D. A. Porto, "Modeling the progression of COVID-19 deaths using Kalman filter and AutoML," *Soft Comput.*, pp. 1–16, Jan. 2021, doi: [10.1007/s00500-020-05503-5](https://doi.org/10.1007/s00500-020-05503-5).
- [32] J. A. L. Marques, F. N. B. Gois, J. Xavier-Neto, and S. J. Fong, *Artificial Intelligence Prediction for the COVID-19 Data Based on LSTM Neural Networks and H2O AutoML*. Cham, Switzerland: Springer, 2021, pp. 69–87, doi: [10.1007/978-3-030-61913-8\\_5](https://doi.org/10.1007/978-3-030-61913-8_5).
- [33] Apple. (2021). *Mobility Trends Reports*. Accessed: Nov. 15, 2021. [Online]. Available: <https://covid19.apple.com/mobility>
- [34] Google. (2021). *Covid-19 Community Mobility Reports*. Accessed: Nov. 15, 2021. [Online]. Available: <https://www.google.com/covid19/mobility>
- [35] J. Gama, I. V. Z. E. Ilobait, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–12, Mar. 2014, doi: [10.1145/2523813](https://doi.org/10.1145/2523813).
- [36] ECDC. (2021). *Historical Data on the Daily Number of New Reported Covid-19 Cases and Deaths Worldwide*. Accessed: Dec. 14, 2020. [Online]. Available: <https://www.ecdc.europa.eu/en/publications-data/download-todays-data-geographic-distribution-covid-19-cases-worldwide>
- [37] ECDC. (2021). *Data on the Daily Number of New Reported Covid-19 Cases and Deaths*. [Online]. Available: <https://www.ecdc.europa.eu/en/publications-data/data-daily-new-cases-covid-19-eueea-country>
- [38] R. Caruana, A. Niculescu-Mizil, G. Crew, and A. Ksikes, "Ensemble selection from libraries of models," in *Proc. 21st Int. Conf. Mach. Learn.* New York, NY, USA, 2004, p. 18, doi: [10.1145/1015330.1015432](https://doi.org/10.1145/1015330.1015432).
- [39] J. Vanschoren, J. N. Van Rijn, B. Bischl, and L. Torgo, "OPENML: Networked science in machine learning," *ACM SIGKDD Explorations Newsltr.*, vol. 15, no. 2, pp. 49–60, 2014, doi: [10.1145/2641190.2641198](https://doi.org/10.1145/2641190.2641198).
- [40] C. Wang, M. Baratchi, T. Bäck, H. H. Hoos, S. Limmer, and M. Olhofer, "Towards time-series-specific feature engineering in automated machine learning frameworks for time-series forecasting," *Eng. Proc.*, vol. 18, no. 1, p. 17, 2022.
- [41] B. Celik and J. Vanschoren, "Adaptation strategies for automated machine learning on evolving data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3067–3078, Mar. 2021.
- [42] H. Akaike, *Information Theory and an Extension of the Maximum Likelihood Principle*. New York, NY, USA: Springer, 1998, pp. 199–213, doi: [10.1007/978-1-4612-1694-0\\_15](https://doi.org/10.1007/978-1-4612-1694-0_15).
- [43] P. B. Nemenyi, *Distribution-Free Multiple Comparisons*. Princeton, NJ, USA: Princeton Univ. Press, 1963.
- [44] X. Wang, K. Smith, and R. Hyndman, "Characteristic-based clustering for time series data," *Data Mining Knowl. Discovery*, vol. 13, no. 3, pp. 335–364, Nov. 2006, doi: [10.1007/s10618-005-0039-x](https://doi.org/10.1007/s10618-005-0039-x).



**JACO TETTEROO** received the M.Sc. degree in computer science from Leiden University, The Netherlands, in 2021. During his studies, he focused on artificial intelligence and advanced data analytics and retrieval. Afterwards, he started his career in the private sector in the role of data specialist at Fortezza, The Hague.



**MITRA BARATCHI** received the M.Sc. degree in computer engineering from the University of Isfahan, Iran, in 2011, and the Ph.D. degree in computer science from the University of Twente, The Netherlands, in 2015. She joined the University of Twente as an EU Erasmus Mundus Ph.D. Fellow, in 2011, where she was a Postdoctoral Researcher, until 2017. In 2017, she joined the Leiden Institute of Advanced Computer Science (LIACS), Leiden University, as an Assistant Professor. Her research interests include machine learning for spatio-temporal and time series data targeting various environmental and industrial applications.



**HOLGER H. HOOS** holds an Alexander von Humboldt Professorship in AI at RWTH Aachen University, Germany, a professorship in machine learning at Universiteit Leiden, The Netherlands, and an Adjunct Professorship in computer science at the University of British Columbia, Canada. He is a fellow of the Association of Computing Machinery (ACM), the Association for the Advancement of Artificial Intelligence (AAAI) and the European AI Association (EurAI), the past President of the Canadian Association for Artificial Intelligence and one of initiators of CLAIRE, and an initiative by the European AI community that seeks to strengthen European excellence in AI research and innovation ([claire-ai.org](http://claire-ai.org)).

...