Modelling the COVID-19 Virus Evolution With Incremental Machine Learning

Andrés L. Suárez-Cetrulo[1][0000−0001−5266−5053], Ankit Kumar[1][0000−0001−8623−6548], and Luis Miralles-Pechuán[2][0000−0002−7565−6894]

[1] Ireland's Centre for Applied AI (CeADAR). University College Dublin, Dublin, Ireland {andres.suarez-cetrulo,ankit.kumar}@ucd.ie
[2] Technological University Dublin, Grangegorman, Dublin, Ireland luis.miralles@tudublin.ie

Abstract. The investment of time and resources for better strategies and methodologies to tackle a potential pandemic is key for dealing with potential outbreaks of new variants or other viruses in the future. In this work, we recreated the scene of 2020 for the fifty countries with more COVID-19 cases reported. We performed some experiments to compare state-of-the-art machine learning algorithms, such as LSTM, against online incremental learning methods (ILMs) in terms of how well they adapted to the daily changes in the spread of the disease and predict future COVID-19 cases. To compare the methods, we performed two experiments: In the first experiment, we trained the models using only data from the country we predicted. In the second one, we used data from the fifty countries to train and predict each one of them. In these two experiments, we used a static hold-out approach for all the methods. Results show that ILMs are a promising approach to model the disease changes over time; ILMs are always up-to-date with the latest state of the data distribution, and they have a significantly lower computational cost than other techniques such as LSTMs.

Keywords: Incremental Machine Learning · Modelling COVID-19 · COVID-19 cases prediction.

1 Introduction

Pandemic curves are non-stationary by nature. Depending on the period, they can show different characteristics such as clear trends, cycles, or seasons where a random component is more prevalent. Moreover, the COVID-19 spread in each region is affected by external factors not captured in the available data [13]. Under these circumstances, incremental and online machine learning (ML) techniques [5] that adapt to the evolution of the trend and its changes, are gaining traction in different domains [8].

The purpose of this work is to explore the suitability of online ILMs to accurately predict the evolution of the COVID-19 virus spread. It is crucial to do more research to find the best strategies and methodologies to tackle potential

outbreaks of other potential viruses so that their effects on public health can be addressed in a better way in the future. Online ILMs have not been exploited yet to this end. Online ILMs represent a relevant alternative due to their ability to adapt to non-stationary behaviours, which is very characteristic of epidemic curves.

These methods and the notion of concept drift have not gained enough attention yet in the coronavirus prediction domain. Concept drift means that the relationships between the inputs and the outputs can change over time due to different circumstances. However, ILMs can deal actively or passively [7] with the non-stationary nature of data streams such as the COVID-19 curve evolution [12]. ILMs do this by adapting (passive) to the non-stationary nature of the data or through the use of drift detectors (active). These methods can find an equilibrium between prioritising new knowledge, adapting to changes, and retaining relevant information through different forgetting mechanisms. This balance is known as the stability-plasticity dilemma [11].

This work aims to forecast the number of positive COVID-19 cases in multiple countries using ILMs and compare their performance with static ML methods. Our contribution consists of proposing a framework to predict the number of new cases while dealing with the evolution in the spread of the curve in different countries. We created a framework (see Github link [3]) to encourage the scientific community to use it as a benchmark to develop new models and strategies to predict COVID-19 cases. To our knowledge, no other publications show the benefits of applying ILMs to predict the number of cases in a pandemic, which makes our research a significant contribution to this area.

The rest of the paper is organised as follows: Section 2 presents an overview of the ILMs. Section 3 presents the conducted experiments to compare the performance of the ILMs methods against other popular methods such as LSTM networks. Then, the performance of all methods is compared under different scenarios and training schemes to find out the optimal configuration. Section 4 compares the obtained results for each of the models and presents an analysis for both the static and the ILMs. Finally, section 5 presents the main findings of our investigation and recommends some interesting lines of research for future work.

2 State of the art

We selected a set of four incremental regression ML models to compare them with the following ML methods: Bayesian Ridge Regression, Linear Support Vector Regression, Random Forest, Decision Tree, Gradient Boosting, and LSTM. Our choice of ILMs covers Incremental Decision Trees that are frequently used for regression problems in the literature [1] and also Adaptive Random Forest for Regression [9], which are an ensemble of incremental trees that have state-of-the-art results in online incremental learning. A description of them can be seen below.

[3] `http://www.github.com/ankitk2109/Covid_Evolution_Using_Incremental_ML`.

- A Hoeffding tree (HT) is an ILMs that assumes that the data distribution is constant over time. This relies mathematically on Hoeffding bounds, which supports that a small sample may suffice to choose an optimal splitting attribute. Hoeffding Trees for regression calculate the decrease of the variance of the target to decide the splits. Its leaf nodes fit linear perceptron models by default [6].
- The Hoeffding Adaptive Tree (HAT) is an adaptive version of the Hoeffding Tree. It replaces old branches with new ones if the error of the old ones increases over time and new branches perform better. To monitor the evolution of the errors, it uses the Adaptive Windowing (ADWIN) algorithm [2]. HAT also proposes a bootstrap sampling as an improvement over Hoeffding Trees.
- Adaptive Random Forest (ARF) [9] is an Adaptive version of the Random Forest ensemble for Data Streams. It manages a pool of trees that are replaced with new ones when a concept drift is detected. As an improvement of RF, each adaptive tree is trained with different samples and feature sets as part of the bagging and the feature bagging process.
- The Passive-Aggressive algorithm (PA) is an online learning algorithm that updates the model depending on the obtained error [4].

3 Experiments and results

In this section, we conduct some experiments in which we compare the performance of ILMs with that of static ML methods; among them, we emphasise the popular deep learning method called LSTM. Our goal is to see if ILMs can quickly adapt to the COVID-19 spread for predicting the number of cases in each country.

3.1 Dataset description

For this work, we used the dataset "COVID-19 Coronavirus data - daily (up to December 14th 2020)" available in the European Open Data Portal[4] and provided by the European Centre for Disease Prevention and Control. The original dataset contains twelve columns with daily information about the disease in 213 countries during 2020; it is structured as follows. One column represents the number of positive cases, another column the number of deaths. Four columns are related to the current date, four other columns are related to country-specific information, one column refers to the continent. And lastly, there is one column that represents the cumulative number of the COVID-19 cases for 14 days per 100,000 inhabitants.

Regarding the preprocessing steps performed before creating the final dataset for the experiments, the columns related to dates and countries were used to split

[4] https://data.europa.eu/euodp/en/data/dataset/covid-19-coronavirus-data-daily-up-to-14-december-2020

the training and testing datasets. The number of new cases is the only column used to generate the feature set (input of the model) and the target (output of the model). The rest of the columns were removed. From the countries with eight or more months of data by November 30th, 2020 (66 in total), to conduct the experiments, we selected the 50 with the highest number of accumulated cases of COVID-19.

Each data example corresponds to a moving time window of fifty consecutive days representing the inputs of the regression model. The target/output of the model is the average of ten consecutive days, where the first of those ten days is 30 days ahead of the last day of the input. The main reason for using the average of ten days is to soften some spikes due to potential delays when reporting the test results. We also wanted to predict 30 days ahead because it allows governments to plan the next few weeks to lift or apply new restrictions on the population. The number of rows for each trained model varies according to the experiment as explained in more detail in section 3.3. The feature set (inputs of the model) was created according to the scheme shown in Figure 1.

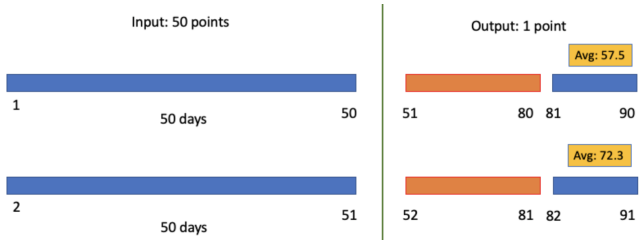

Fig. 1. Fifty points are taken as the models' input to predict a single point, which is the average of ten consecutive days after 30 days.

3.2 Experimentation Methodology

Our primary concern in these experiments is to put ourselves in the shoes of a country facing a pandemic that only has certain information available at a particular date, and that needs to generate models to predict the future COVID-19 cases so that it can provide the government with useful information for taking the optimal actions. Those actions could be closing schools, limiting public transport, applying lock-downs, among others.

The experiments were conducted to answer the following two questions. First, between incremental and static methods, which have higher performance for predicting the number of new COVID-19 cases? And, second, what is the best option between these two for training a model to predict the future number of cases of COVID-19 for a given country? a) Training the model only with the samples of that same country which the model was going to predict. Or b) Training the model with the complete dataset of fifty countries.

To respond to these questions, initially, we performed two experiments. In experiment I, we trained the static and incremental methods with only one country, and we predicted the future COVID-19 cases for that same country. In the second experiment, we predicted over one country, but this time, we trained the supervised models with the 50 countries with most cases. Then, we compared the performance of training the models using a single country with the results obtained using multiple countries (MC). To make ILMs comparable, we trained and tested all the algorithms using the same training and testing sets. However, ILMs for data streams are designed to be trained continuously, and static train and test splits are not generally applied to ILMs in the literature.

This continuous training setting is not usually applied in static ML models due to the computational burden of training an algorithm for each new training batch. Static models need re-training strategies to keep the models up to date when dealing with non-stationary or continuous learning scenarios. In any case, the use or optimisation of re-training strategies is outside the scope of this paper. Still, to make a fair comparison, we used the training sets (input models) shown in Figure 2 as a pre-trainining set and used the test splits (output models) shown in the same figure as a test-then-train set.

To calculate the average performance of the different algorithms during the pandemic, these methods were evaluated at eight points in time, which we called milestones. Each milestone represents a date on which we predicted future cases considering only previous information to that point. Figure 2 illustrates the evaluation of the applied ML methods. Each milestone's test set covers a month interval after its respective training set. Using the subset of dates given by each monthly milestone, we created samples that contain train and test data for the subsequent experiments.

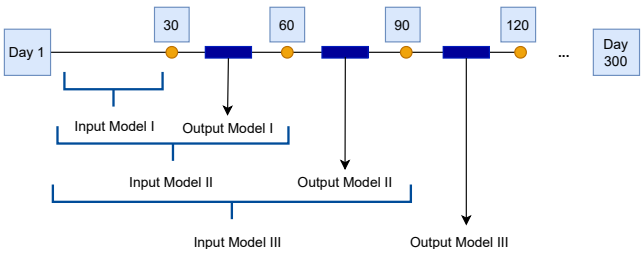

Fig. 2. For evaluating the models we defined eight milestones (represented with orange circles). The performance of the methods was calculated as the average of all the evaluations.

Since the number of models was four hundred, eight milestones per fifty models, all algorithms are implemented using their default vanilla configuration. A validation set for back-propagation was created using the last ten days of the training set at each respective milestone for the LSTM only. In this paper, we

use the version of Passive-Aggressive Regressor provided in Scikit-Learn [10]. For the rest of the approaches, we have used the implementation provided in Scikit-Multiflow.

The LSTM was trained for 500 epochs in both experiments. However, the batch size and *patience* were different in the SC and MC approaches. The *batch size* refers to the number of training examples used in one iteration when training the LSTM sequentially. *Patience* represents the number of epochs to wait before early stop training the algorithm if the model does not lower its error.

In experiment I, we defined a batch size of 10 examples (one example per day) and a *patience* of 20 examples for the LSTM. In experiment II, we defined a batch size of 500 examples (10 days multiplied by 50 countries) and a *patience* of 1000 examples (20 days multiplied by 50 countries). This methodology of dividing the dataset into milestones and calculating the error as the average of the milestones was used throughout all the experiments. The performance of the models was measured using the Mean Absolute Percentage Error (MAPE) according to the state-of-the-art metrics for regression [3]. The results are calculated by comparing the model's predictions to the target feature in the test set (or test-then-train set in ILM).

MAPE describes how far the predictions of a model are from their corresponding outputs on average. MAPE allows comparing forecasts of different series in different scales as it is expressed in percentage-like terms. Results from MAPE can also differ from MAE as MAPE's values are undefined for data examples where the target or prediction value is zero. Thus, MAPE would be higher for an algorithm compared to other metrics if the target values are close to zero. This paper considers MAPE as the evaluation metric mainly because it is a unit free metric and reports percentage-like terms.

Section 3.3 shows and discusses the results of three experiments performed in this work. Experiment I is focused on models trained for a single-country (SC). Each algorithm is trained and tested at eight different points in time, as explained in Figure 2. Experiment II applies the same process and periods as Experiment I, but each algorithm is trained with all the dataset covering all countries. Rows in these datasets are sorted first by date and then by country to respect the chronicle order of the time series for the different training, test splits and batches already mentioned.

To compare the SC results with the multi-country (MC) results from Experiment 2, the errors from the 50 models trained for an SC are averaged. Another difference between Experiment I and Experiment II relies on the batch size for the incremental and sequential learners. Batches are time-wise for a set of days. Thus, a batch of data during training or testing is 50 times bigger in Experiment II because we are training the models with data from 50 countries rather than with a single one. Finally, we compare the results from the SC with the MC approach.

3.3 Experimentation

In this section, we show the results in different plots for the two different experiments. We also add a subsection in which we compare the performance of the single country approach with the MC approach.

Experiment I: Single-Country training This experiment trains the supervised ML models with an SC and predicts the cases for the same SC with which the model is trained. Results are obtained averaging eight points called milestones for which predictions are made. Albeit the top performers can change in different countries, it is clear that the static methods show an overall performance higher than the ILMs

Although Gradient Boosting and the Decision Tree are the algorithms with the lowest MAPE, we also consider that LSTM is one the best performers for SC experiments in the context of the COVID-19 crisis since it obtains the lowest average MAPE across all milestones.

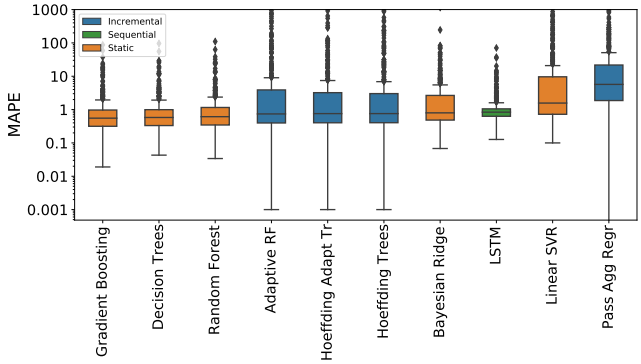

Fig. 3. Boxplot for MAPE per algorithm for the single-country approach (SC). Results aggregate 50 experiments for different single countries.

Experiment II: Multiple-Countries training This second experiment predicts over the same 50 countries at the same eight points as Experiment I, but this time we train the model with 50 countries rather than training it with a single country as in Experiment I. In the MC experiment, Support Vector and the tree-based ILMs (HT, HAT and ARF) obtain the lowest MAPE across the eight-time points. Figure 4 shows how HT and HAT have a lower median MAPE than Support Vector Regression. According to Figure 4, HT and HAT seem the most reliable predictor for the MC experiment, as they offer the lowest medians, and most of their experiments fall into a normal distribution. Using the median MAPE rather than its mean as a comparative metric is helpful since the value of the mean can be distorted by the outliers.

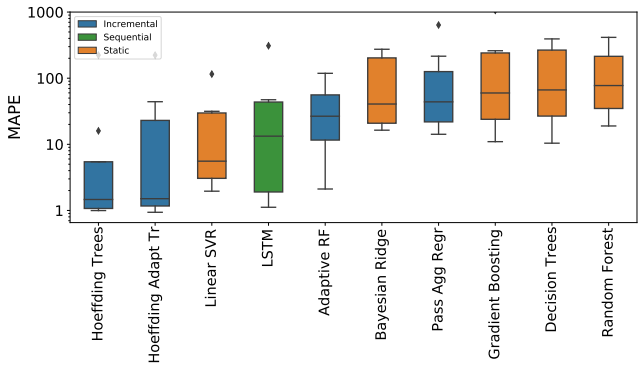

Fig. 4. MAPE per algorithm for the multi-country experiment (MC) covering the 50 countries from Figure 3.

Regarding the time for training the methods, Support Vector Regression proves to be the most cost-effective solution across the eight milestones. Support Vector Regression obtains one of the three lowest MAPE and is the fourth fastest method of all the algorithms benchmarked in experiment II. Figures 4 and 3 are both ordered by the median of the MAPE values in eight-time points. While LSTM, Decision trees, Gradient Boosting and Random Forest offer overall better results in SC, the ILMs have better performance in terms of MAPE in the MC approach. We believe that the dominance of the ILMs in the results of the MC approach is because these can handle complex scenarios better, due to their sequential training nature. We give more details on these thoughts in subsection 3.3.

The evolution of the MAPE obtained by each model per time-point can be seen in Figure 5. It can be seen how the ILMs HT and HAT are the best performers in the last milestone, followed by LSTM, which shows a smooth evolution across the milestones. The next best performer in the last milestone is Linear SVR, which shows the lowest MAPE mean due to its good performance for less training data in the initial milestones.

Comparison between the single country training and Multiple-countries training In the second experiment, all countries were concatenated under a single dataset. The MC approach was proposed to provide both static and ILMs with an augmented dataset that includes other countries' information and test their predictive ability with a more complex but broader set. The reader must note that this experiment was conducted to test the capacity of the models to handle a set of multiple countries where the curve of cases may not be aligned in all cases between different countries. We are aware that a batch for the ILMs may feed multiple states of the COVID-19 curve due to the different countries used. However, we consider this could be handled, for instance, by incremental ensembles like ARF.

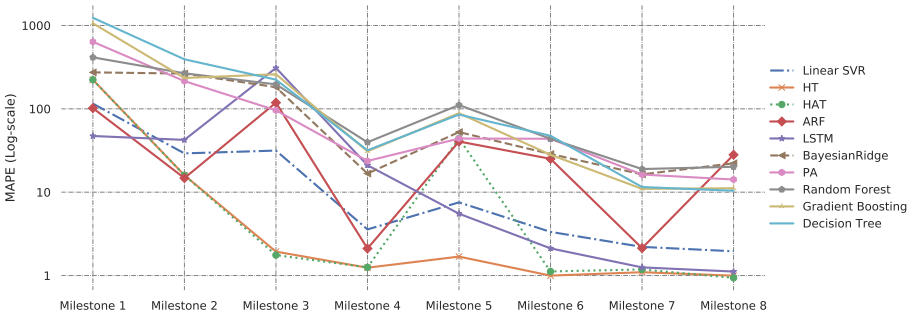

Fig. 5. Evolution of MAPE per algorithm per milestone for the multi-country approach. The right-hand-side legend is sorted by the mean MAPE.

In general, the SC approach exhibits lower MAPE values than the MC approach. Thus, SC can be seen as the best of the two approaches. The LSTM (SC) is the best performer overall across the 50 countries used in the experiments in the eight-time points. We believe that the MC experiment needs a model able to map non-linearity sequentially or incrementally. We believe this could be handled by an incremental ensemble like Adaptive Random Forest (ARF). However, ARF was designed for purely incremental scenarios, and the purpose of its drifts detectors is to replace base regressors when the data distribution requires. Thus, the hold-out, static evaluation scheme used in this paper constrains this designed behaviour.

4 Discussion

Experiments I and II compare traditional machine learning regression algorithms to ILMs for 50 countries, eight-time points and a common hold-out evaluation scheme. Their results show how traditional static ML techniques perform better than ILMs in the SC experiment. Models like HT and ARF are designed for data streaming scenarios and to handle large amounts of data. We see how they beat static methods when trained for a broader set in the MC approach. These tend to adapt better over time and offer the best performance in the last milestones. Furthermore, ILMs are oriented to online scenarios and continuous adaptation.

While one may think that the MC approach should give enough information to most models to improve their performance, the results show the opposite. This is probably because many of these countries have very different behaviours in the evolution of COVID-19 cases and can mislead rather than help when the model predict a particular country. That is to say, at the same time point, different countries may be in states of an outbreak different from each other, such as at the start or the end of a different COVID-19 wave. The presence of different states of the disease across countries adds extra complexity to the MC approach compared to the SC approach. COVID-19 is an example of a concept

drifting environment. This experiment evaluates the ability of the ILMs to deal with different drift concepts at a time, as different countries could be on different moments of waves (or even in different waves).

In the MC experiment, Support Vector regression presents lower MAPE for the first milestones. However, incremental approaches improve their performance as the training data increases. In any case, there is no statistical significance between SVR and LSTM in the MC experiment. The LSTM is overall the best method across experiments. Besides its computational cost (20 times the runnning time of ILM), it offers a smooth adaptation as the data set increases (see Figure 5).

Our results show how ILMs can obtain the lowest error for the MC experiment. ARF, the ensemble of HT, is the incremental algorithm with the lowest MAPE when using a prequential evaluation. We believe that different trees of the adaptive ensemble may be learning about different sets of countries (due to the bagging mechanism) that may perform more or less similar and adapt continuously to any concept changes. As for adaptive single learners or static ensembles, other algorithms can adapt to different concepts by learning incrementally or having a set of base learners for different stationarities using a prequential scheme.

5 Conclusion

In this paper, we backtested the daily information about COVID-19 cases during 2020 for 50 different countries to recreate a situation in which the countries had to face the threat of the number of cases increasing and protect the economy of the country at the same time. In 2020, the information about the spread of the virus and the new outbreaks was very limited. Our research is valuable because of the insights of the experiments in which we compare ML static methods versus the performance of online ILMs for predicting the number of new cases.

Our results show that the proposed approach of using ILMs can outperform the traditional literature static methods when training for multiple countries. ILMs adapt over time and obtain lower errors in the last periods. These algorithms also show their ability to adapt to the non-stationarities exhibited in these time series. ARF obtains a MAPE error four times lower when using a prequential evaluation instead of a hold-out scheme. Across experiments I and II, the SC experiment obtains the best results. In the SC experiment, the LSTM is the algorithm with the lowest MAPE. We should highlight that since the LSTM is designed to handle data of a sequential nature and previous results from the literature, its performance is not a surprise. The LSTM is one of the algorithms that adapt better over time across the milestones in the MC experiment. In any case, even following a hold-out scheme, the ILMs HT and HAT obtain the lowest median MAPE.

Lastly, we proved that models trained with an SC tend to obtain lower errors (better results) and that the error tends to diminish as the models are trained with more information. This is probably because some countries are very different

from each other and can misguide the classifier. For the approach of predicting single countries, ILMs tend to obtain higher errors (worse results) than those in other ML techniques when compared using the static scheme of train and test hold-out splits. In any case, the proposed hold-out static scheme has proved to be a constraint by design for ILMs.

For future work, we would like to explore a new approach. Rather than training a classifier with all the 50 countries, we could train the classifier only with those that behave similarly to the one being predicted. In other words, we will calculate first the distance between the two countries using a time series similarity measure such as euclidean distance, Dynamic Time Warping, or Symbolic Aggregate approXimation (SAX). And then we will use a threshold to select the countries with the lower distance to the predicted country. We would also like to compare our model to other approaches like the SEIR (susceptible-exposed-infected-recovered) or the SIR model that have been widely used for this purpose.

References

1. Bahri, M., Bifet, A., Gama, J., Gomes, H.M., Maniu, S.: Data stream analysis: Foundations, major tasks and tools. WIREs Data Mining and Knowledge Discovery **11**(3), e1405 (2021). https://doi.org/10.1002/widm.1405
2. Bifet, A., Gavaldà, R.: Adaptive learning from evolving data streams. In: International Symposium on Intelligent Data Analysis. pp. 249–260. Springer (2009)
3. Botchkarev, A.: Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. arXiv preprint arXiv:1809.03006 (2018)
4. Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y.: Online passive-aggressive algorithms. Journal of Machine Learning Research **7**(19), 551–585 (2006), `http://jmlr.org/papers/v7/crammer06a.html`
5. Ditzler, G., Roveri, M., Alippi, C., Polikar, R.: Learning in Nonstationary Environments: A Survey (nov 2015). https://doi.org/10.1109/MCI.2015.2471196
6. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 71–80 (2000)
7. Elwell, R., Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE transactions on neural networks **22**(10), 1517–31 (10 2011)
8. Gama, J.A., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A Survey on Concept Drift Adaptation. ACM Comput. Surv **1**(35), 1–37 (mar 2013). https://doi.org/10.1145/0000000.0000000
9. Gomes, H., Bifet, A., Boiko, L., Barddal, J., Enembreck, F., Pfharinger, B., Holmes, G., Abdessalem, T.: Adaptive random forests for evolving data stream regression. Machine Learning **106**(9-10) (2017). https://doi.org/10.1007/s10994-017-5642-8
10. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
11. Singh, B., Sun, Q., Koh, Y.S., Lee, J., Zhang, E.: Detecting protected health information with an incremental learning ensemble: A case study on new zealand clinical text. In: 2020 IEEE 7th International Conference

on Data Science and Advanced Analytics (DSAA). pp. 719–728 (2020). https://doi.org/10.1109/DSAA49011.2020.00082
12. Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. Technical Report: TCD-CS-2004-15, Department of Computer Science Trinity College, Dublin (2004)
13. Wang, L., Didelot, X., Yang, J., Wong, G., Shi, Y., Liu, W., Gao, G.F., Bi, Y.: Inference of person-to-person transmission of covid-19 reveals hidden super-spreading events during the early outbreak phase. Nature communications **11**(1), 1–6 (2020)