Heliyon 9 (2023) e18771

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

A novel online multi-task learning for COVID-19 multi-output spatio-temporal prediction

Zipeng Wu [a], Chu Kiong Loo [a,*], Unaizah Obaidellah [a], Kitsuchart Pasupa [b]

[a] Faculty of Computer Science & Information Technology, University of Malaya,Kuala Lumpur, 50603, Malaysia

[b] School of Information Technology, King Mongkut's Institute of Technology Ladkrabang,Bangkok, 10520, Thailand

ARTICLE INFO

Keywords:
Online learning
Multivariate time series
Multi-task learning
COVID-19 prediction

ABSTRACT

In light of the ongoing COVID-19 pandemic, predicting its trend would significantly impact decision-making. However, this is not a straightforward task due to three main difficulties: temporal autocorrelation, spatial dependency, and concept drift caused by virus mutations and lockdown policies. Although machine learning has been extensively used in related work, no previous research has successfully addressed all three challenges simultaneously. To overcome this challenge, we developed a novel online multi-task regression algorithm that incorporates a chain structure to capture spatial dependency, the ADWIN drift detector to adapt to concept drift, and the lag time series feature to capture temporal autocorrelation. We conducted several comparative experiments based on the number of daily confirmed cases in 20 areas in California and affiliated cities. The results from our experiments demonstrate that our proposed model is superior in adapting to concept drift in COVID-19 data and capturing spatial dependencies across various regions. This leads to a significant improvement in prediction accuracy when compared to existing state-of-the-art batch machine learning methods, such as N-Beats, DeepAR, TCN, and LSTM.

1. Introduction

The COVID-19 pandemic has presented an unprecedented challenge to the global community, affecting nearly every aspect of our lives. One of the key challenges has been predicting the spread of the virus, which is critical for planning and decision-making. However, traditional prediction models have struggled to keep up with the rapidly changing nature of the pandemic. This paper aims to address this issue by proposing a novel approach to predicting COVID-19 trends.

Two years have passed since the outbreak of COVID-19 with over 542 million people infected worldwide and the total number of deaths related to the outbreak has exceeded 5 million [1]. The pandemic is far from over with a large number of new infected worldwide, although many countries have declared coexistence with the virus.

As the number of infections has increased exponentially, the virus has mutated several times, with some strains proving to be much more transmissible than the original. This has resulted in concept drift in the daily confirmed cases of COVID-19. Concept drift refers to the unforeseen changes over time in the statistical properties of the target variable that a model is attempting to predict [2]. The patterns observed in past data may no longer be applicable to the current situation, leading to inaccurate predictions and suboptimal decision-making. A set of examples denoted as $S_{0,t} = \{d_0, \dots, d_t\}$, where $d_i = (x_i, y_i)$ is one observation (or a data

* Corresponding author.
E-mail address: ckloo.um@um.edu.my (C.K. Loo).

https://doi.org/10.1016/j.heliyon.2023.e18771
Received 20 April 2023; Received in revised form 26 July 2023; Accepted 27 July 2023

Available online 2 August 2023
2405-8440/© 2023 Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

instance), x_i is a feature vector, y_i is their corresponding label, and $S_{0,t}$ follows a certain distribution $F_{0,t}(x, y)$. Concept drift occurs at timestamp $t + 1$, if $F_{0,t}(x, y) \neq F_{t+1,t+n}(x, y)$, denoted as $\exists t : P_t(x, y) \neq P_{t+1}(x, y)$.

The rapid mutation of the virus, from the emergence of SARS-CoV-2 to the Omicron wave, driven by the variants BA.4 and BA.5, has significantly changed the ability of the virus to spread. This change has caused a shift in the pattern of COVID-19 trends, particularly in the number of daily confirmed cases. As a result, models built using historical data of the previous virus become gradually invalid due to concept drift in COVID-19 daily confirmed cases data. This highlights the need for a more dynamic and adaptable approach to predicting COVID-19 trends.

Accurately predicting COVID-19 trends can have a significant impact on decision-making, such as adjusting medical strategies in advance to avoid overwhelming the healthcare system. However, predicting the number of daily confirmed COVID-19 cases in multiple regions is a complex problem due to the simultaneous presence and interaction of the three challenging perspectives.

- **Temporal Autocorrelation:** The number of future COVID-19 cases is correlated with the number of cases in previous time periods, since the virus is spread from infected individuals. Additionally, the spatial autocorrelation between cities may vary over time because of human mobility. Therefore, the first challenge is to determine how to model the dynamic temporal autocorrelation of each city.
- **Spatial Dependency:** The daily confirmed COVID-19 cases of a city are not only affected by the transmission of the virus in nearby cities but may also synchronize with distant cities due to similarities in their characteristics. The second challenge is how to model the irregular and non-Euclidean autocorrelation of COVID-19 transmission between cities.
- **Concept Drifts:** Since 2019, the new coronavirus has mutated several times. Some variants of the coronavirus, such as Delta and Omicron, are spreading more easily between people, which would lead to changes in the distribution of COVID-19 confirmed cases data. For example, a predictive model built during the spread of the Delta virus may not be suitable for predicting daily confirmed cases during the spread of the Omicron virus. Additionally, vaccination and lockdown policies can cause changes in the trend of COVID-19.

Machine learning methods have recently made prominent achievements in various fields of prediction, such as [3–6]. These machine learning methods generally refer to conventional machine learning methods, i.e., batch machine learning algorithms, which assume that the distribution of data is static. However, the dynamic nature of COVID-19 trend data requires a different approach. In this paper, we propose an online machine learning method that can adapt to changes in data distribution over time, making it more suitable for predicting COVID-19 trends.

Existing research on COVID-19 prediction can be broadly divided into two categories. The first category consists of conventional methods, including conventional machine learning algorithms and time series models, such as ARIMA, random forests, linear regression, and support vector machine regression (SVR) [7]. The second category consists of deep learning algorithms [8], e.g., N-Beats [9], LSTM [10–13]. While these methods have been effective in certain contexts, they have limitations. Although these researchers used different algorithms for modeling, their works only considered the temporal correlation of COVID-19 trend. Moreover, the performance of the prediction models depended heavily on how well the researchers exploited the temporal correlation on the characteristic case dataset. The batch machine learning algorithm achieved excellent predictive performance during the stable phase of the COVID-19 trend when the virus had not mutated into a more transmissible variant, and the embargo policy remained unchanged. However, once concept drift occurs, their batch machine learning model is likely to not work well. Existing research has emphasized the first challenge and has not paid much attention to the second and third challenges. Moreover, no method has yet considered all three challenges as a whole and addressed them simultaneously. In light of these limitations, this paper proposes a novel multi-task online regression algorithm that can address the dynamic nature of the COVID-19 pandemic. Our approach, which we call Multi-Task **C**orrelated **h**oeffiding **a**daptive **t**ree **R**egressor **C**hain (ChatRC), is a type of hoeffiding tree variate for multi-task online regression under concept drift. This approach is designed to handle the temporal autocorrelation, spatial dependency, and concept drifts associated with COVID-19 trends prediction.

The main contribution of this paper is as follows:

- We propose a novel online multi-task regression algorithm to predict the daily new confirmed COVID-19 cases in multiple regions/zones. This algorithm, ChatRC, is a unique combination of online learning, multivariate time series, and multi-task learning, providing a novel approach to the dynamic and complex nature of COVID-19 prediction.
- Provide a clear description of the spatial correlation or dependence among the number of new confirmed COVID-19 cases in each region, which can provide helpful information to decision-makers beyond prediction.
- Make the first attempt to apply online machine learning to COVID-19 prediction. As will be shown, online machine learning can far exceed conventional batch machine learning when dealing with scenarios such as COVID-19 prediction that require frequent or real-time updates of the prediction model.

1.1. Problem statement

This section formally defines the problem in this study and provides mathematical notation annotations. Predicting COVID-19 confirmed cases involves multivariate time series prediction, where these time series exhibit spatial correlation with each other. Therefore, it is a multi-output spatio-temporal prediction problem. Furthermore, it is a subset of multiple-output regression and

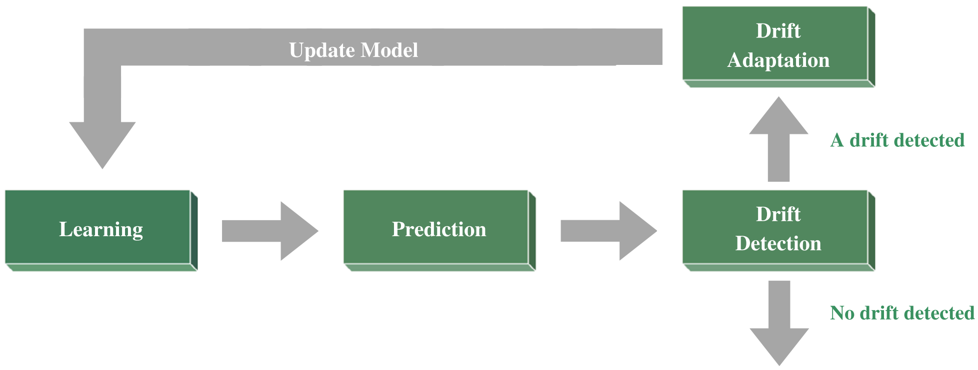

Fig. 1. Prediction System with Drift Adaptation Mechanism.

multivariate time series forecasting, where each output variable is a single time series that is spatially correlated or dependent on the others.

Multi-output spatio-temporal data has the properties and characteristics of a multivariate time series as shown in (1),

$$\boldsymbol{y} = \left\{ y_1, \dots, y_t, \dots, y_N \right\}, t = 1, \dots, N. \tag{1}$$

A multiple time series is composed of multiple one-dimensional time series as components, with each component having the same sampling time points. It can be represented in a matrix form as (2),

$$\mathbf{Y} = \left\{ \boldsymbol{y}_1, \dots, \boldsymbol{y}_m \right\}, \tag{2}$$

where each row corresponds to a single time point, and each column corresponds to a unary time series. The spatial dependency $c_{i,j}$ between two time series $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$, can be measured by their correlation coefficient, $c_{i,j} = corr(\boldsymbol{y}_i, \boldsymbol{y}_i)$.

The temporal autocorrelation is denoted as (3),

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p}, \tag{3}$$

where y_t represents the number of newly confirmed COVID-19 cases on the t-th day, y_{t-p} represents the number of newly confirmed cases on the $(t-p)$-th day, ϕ is a weight variable, and ϕ_p represents the weight corresponding to the early p days.

In simple terms, a forecast where there is spatial dependence or correlation between the outputs of multiple time series is called a multi-output spatio-temporal forecast. This type of forecasting is particularly relevant for predicting COVID-19 trends, given the interconnected nature of the pandemic.

In the batch machine learning perspective, COVID-19 multi-output spatio-temporal prediction can be defined as a multi-output regression problem. Therefore, it is critical to find a function $f(\mathbf{X})$ that maps the (input) feature space $\mathbf{X} \in \mathbb{R}$ to the (target) output space $\mathbf{Y} \in \mathbb{R}$ in the dataset. The function f is used to simultaneously predict the (target) output variables $\boldsymbol{y} \in \mathbb{R}$ with their corresponding input vector $\boldsymbol{x} \in \mathbb{R}$ as $\boldsymbol{y} = f(\boldsymbol{x})$, where $\mathbf{X} = \left\{ \boldsymbol{x}_1, \dots, \boldsymbol{x}_N \right\}$ and $\mathbf{Y} = \left\{ \boldsymbol{y}_1, \dots, \boldsymbol{y}_m \right\}$. If concept drifts occur at time t and time $t+n$, the ideal function should be represented as (4),

$$f(\boldsymbol{x}_i) = \begin{cases} f_{0,t}(\boldsymbol{x}_i), & 0 \leqslant i \leqslant t \\ f_{t+1,t+n}(\boldsymbol{x}_i), & t+1 \leqslant i \leqslant t+n \\ f_{t+n,N}(\boldsymbol{x}_i), & t+n \leqslant i < \infty \end{cases}. \tag{4}$$

In the presence of concept drift, COVID-19 prediction is not a static pattern. Although conventional batch machine learning can achieve good results for a certain period, their models are not suitable once concept drift occurs. One simple way to adapt to concept drift is to retrain the batch machine learning model to learn new patterns whenever concept drift occurs. However, the difficult part is that retraining the model can be costly, and choosing the interval of training samples is a complex problem. Online machine learning, on the other hand, does not face these complexities. One of the most prominent ways to adapt to concept drift is through online machine learning, which relies on the core mechanism of learning samples one by one. This means that the model can be updated in real-time.

The entire prediction system drift adaptation is executed as shown in Fig. 1. When a new example arrives $(\boldsymbol{x}, \boldsymbol{y})$, the model makes a prediction based on $\boldsymbol{x}$ and then receives the ground truth $\boldsymbol{y}$ for the training or learning of the new example $(\boldsymbol{x}, \boldsymbol{y})$. Finally, the drift detector detects whether a concept drift has occurred. If it has occurred, the model will be updated. This real-time update mechanism allows our proposed method, ChatRC, to quickly capture the latest patterns in the data and adapt to the changing nature of the COVID-19 pandemic, providing a more accurate and dynamic prediction model.

Table 1
The related work discusses the predictive methods used and whether the three main challenges were addressed, sorted by date.

Author	Method	Concept Drift Adaptation	Spatial Dependency	Temporal Autocorrelation	Date
[14]	SIPHERD	NO	NO	YES	Jul-2020
[10]	LSTM	NO	NO	YES	May-2020
[15]	Genetic Programming based Model	NO	NO	YES	May-2020
[16]	TP-SMN-AR	NO	NO	YES	May-2020
[17]	Multi-Output Gaussian Processes	NO	NO	YES	Jun-2020
[13]	Hybrid MLP-LSTM	NO	NO	YES	Jul-2020
[11]	RNN, LSTM, BiLSTM, GRUs, and VAE	NO	NO	YES	Jul-2020
[18]	EVDHM-ARIMA	NO	NO	YES	Jul-2020
[19]	ARIMA, NARNN, and LSTM	NO	NO	YES	Jul-2020
[20]	SIR based GCN and LSTM	NO	YES	YES	Oct-2020
[21]	LSTM	NO	NO	YES	Aug-2020
[22]	ARIMA	NO	NO	YES	Oct-2020
[23]	Multi-Head Attention, LSTM, and CNN	NO	NO	YES	Oct-2020
[24]	RNN	NO	NO	YES	Oct-2020
[25]	SVR	NO	NO	YES	May-2020
[26]	VARMAX	NO	NO	YES	Dec-2020
[27]	Bayesian Model Averaging based Ensemble Learning	NO	NO	YES	Jul-2022
[8]	ConvLSTM — CNN merged with an LSTM	NO	YES	YES	Jul-2022
[28]	SIR,Adaptive Phase-Space Approach	NO	NO	YES	Jul-2022
[29]	SVR, MLP, and Prophet	NO	NO	YES	May-2021
[30]	Brute force trained Ridge Regression	YES	NO	YES	Sep-2022

2. Related work

COVID-19 time series forecasting can be classified into three main challenges: temporal autocorrelation, spatial dependency, and concept drift. Based on Table 1, we can observe the methods used in related works and their responses to these three challenges. Almost all related work focuses only on temporal autocorrelation. According to the methods used, these related works can be divided into three categories: (i) traditional time series methods, (ii) Susceptible-Infectious-Recovered (SIR) based methods, and (iii) machine learning methods.

Despite the significant progress made in COVID-19 time series forecasting, there is a clear research gap in addressing all three challenges simultaneously, especially in the context of online learning. Most existing studies focus on temporal autocorrelation, with only a few considering spatial dependency and even fewer addressing concept drift. Moreover, the few studies that do consider concept drift often use computationally intensive methods that are not suitable for real-time implementation. Our study aims to fill this research gap by proposing a novel online multi-task regression algorithm that addresses all three challenges simultaneously in a computationally efficient manner.

In the following paragraphs, we provide a detailed review of the existing methods and highlight their limitations, further emphasizing the novelty and significance of our proposed method.

Some researchers have chosen conventional time series methods, such as AutoRegressive Integrated Moving Average (ARIMA) [22,19], and its variants such as VARMAX [26], EVDHM-ARIMA [18], and TP-SMN-AR [16]. The ARIMA model is a statistical method used for time series forecasting. It is a linear model that uses a combination of autoregressive (AR) and moving average (MA) terms to model the evolution of a time series over time. One limitation of the ARIMA model is that it assumes that the time series data follows a stationary process, meaning that the statistical properties of the data do not change over time. However, the spread of infectious diseases like COVID-19 is influenced by many factors, such as public health measures, human behavior, and other external factors, which can cause the data to exhibit non-stationary behavior. Thus, forecasting using conventional time series methods has some limitations, such as not accounting for the effects of exogenous variables and generally only capturing temporal autocorrelation over short time periods. On the other hand, conventional time series forecasting methods have advantages, such as requiring a small number of data samples, having low model complexity, and high interpretability. For example, conventional methods allow for the decomposition of time series components, visualization of seasonal components, trend components, residual components, and other components.

In addition, the SIR model [31] and its variant SIPGERD [14] have also been used. The SIR model is a mathematical model commonly used to study the spread of infectious diseases. The model assumes that a population can be divided into three categories: susceptible individuals who are susceptible to contracting the disease, infected individuals who can pass the disease along to others, and recovered individuals who have recovered from the disease and are no longer infectious. The SIR model uses a set of differential equations to describe the flow of individuals between these three classes over time. Equations are built using data about the rate of infection, the rate of recovery, and population size. By solving these equations, the model can predict the evolution of the disease over time and the eventual outcome of the epidemic. However, this type of model is sensitive to its parameters and many of its predictions may be incorrect after concept drift occurs.

Many works also exhibit the usage of machine learning methods, such as Support Vector Machine Regression (SVR) [25,29] and neural network-based methods — Recurrent Neural Networks (RNN) [24] and Long-Short Term Memory (LSTM) network [32,11,10,20,23]. RNNs can process input data of varying length and use the previous inputs to inform the current input's processing, making them useful for predicting future trends based on temporal autocorrelation. LSTMs, a type of RNN, can better capture long-term dependencies in data by storing and retrieving information from previous inputs using a specialized memory cell. Although no single method can guarantee the highest prediction performance for all datasets, LSTM and its variants have shown to be more accurate than other deep learning methods for COVID-19 trend prediction. Some studies have considered both spatial dependency and temporal autocorrelation in COVID-19 trend prediction. For instance, ConvLSTM [8] combines convolutional neural networks to capture spatial dependency with LSTM to capture temporal autocorrelation. On the other hand, Graph Convolutional Neural Network is used to capture spatial dependency in [20]. However, it is difficult to explain the captured dependence intuitively. The COVID-19 confirmed number trend prediction with concept drift was first discussed in [30]. However, they only considered it with temporal information by brute-force tuning the model, which is computationally intensive and not suitable for real-time implementation, and did not consider the spatial factor.

Overall, all three of these challenges have been investigated and some progress has been made, providing useful insights into next steps. To the best of our knowledge, there is no single solution that addresses all three challenges simultaneously up to now in the online learning setting.

3. Method

The three challenges mentioned in the introduction — temporal autocorrelation, spatial dependency, and concept drift — exist simultaneously and affect each other, but previous approaches cannot solve all three problems at the same time. To handle all these challenges in COVID-19 multitask spatio-temporal prediction, our system comprises three main components: (i) using the lag time feature to capture temporal autocorrelation in the daily confirmed case numbers from COVID-19; (ii) utilizing the chain structure to capture the spatial dependency between zones; and (iii) employing the Hoeffding tree and ADWIN drift detector to adapt to concept drift in the data. This section first introduces the overall structure of the COVID-19 multi-task spatio-temporal prediction system referred to as "ChatRC" and then describes how the proposed approach copes with the three challenges in detail.

3.1. The ChatRC's structure

In terms of model structure, ChatRC is a Hoeffding adaptive tree chained in order of maximum correlation. ChatRC can be divided into two phases: the Hoeffding adaptive tree model initialization, and the incremental learning process. The first phase has two components: (i) correlation calculation and output sorting, where we calculate the correlation between different output variables and sort them based on their correlation values; and (ii) Hoeffding tree and ADWIN drift detector initialization, where we initialize the Hoeffding tree and the ADWIN drift detector for each output variable. The second phase of ChatRC has five main steps: (i) expanding meta-features, where we add new output as features to the model based on the current state of the data; (ii) updating leaf node information, where we update the information stored at each leaf node based on the new data; (iii) attempting to split, where we try to split the tree based on the Hoeffding bound; (iv) ADWIN drift detection, where we use the ADWIN drift detector to detect any concept drift in the data; and (v) replacing node subtrees, where we replace the subtrees of the model if the ADWIN drift detector detects any concept drift.

In the next three subsections, we will describe how ChatRC tackles each of the three challenges.

3.2. Adapting to concept drift

Online learning is a practical approach for dealing with concept drift, as it enables the learning of examples one-by-one and the real-time updating of the model.

3.2.1. Hoeffding tree

The Hoeffding tree is a decision tree algorithm for input data streaming [33]. It has been very popular due to its attractive property of being guaranteed to perform as well as conventional decision tree algorithms, provided that the number of instances is large enough. ChatRC inherits this feature perfectly, as it is a core component of the model. The pseudocode of the Hoeffding tree is shown in Algorithm 1.

Each time a new example arrives, the algorithm executes the following steps: Traverse the tree based on the example's features until it reaches the corresponding leaf node; it then updates the information stored at that leaf node with the example's features or label information; the algorithm maintains a table for each node containing the observed features and label values, and update the statistical information on numerical attributes by computing the mean and standard deviation of each new example; Each leaf node also stores the examples observed so far.

Hoeffding Tree Regression involves evaluating potential split candidates by measuring the reduction of variance in the target space. When the variance of the label is high at the leaf node, the algorithm calculates the split criterion $\bar{G}$, variance reduction, of each feature, as a regression tree. If the difference in information gain between the two features with the highest split criterion $\Delta\bar{G}$ is greater than Hoeffding's bound, the tree will split at the best feature, creating new leaves and initializing statistical information for them. The Hoeffding bound ϵ is calculated using (5),

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}, \tag{5}$$

where n is a number of independent instances of a random variable r with range R.

Algorithm 1: Hoeffding Tree (Regression).

Data: stream data $(\mathbf{x}, \boldsymbol{y})$
Result: Hoeffding Tree h
Let HT be a tree with a single leaf (the root), the split criterion $\bar{G}$ is variance reduction ;
for *all training examples* $(\boldsymbol{x}, \mathbf{y})$ *in stream* **do**
  Sort example $(\boldsymbol{x}, y_j)$ into leaf l using HT_i;
  Update sufficient statistics in l;
  Increment the number of examples seen at l, n_l;
  if *variance of these labels at the leaf node increases* **then**
    Calculating the $\bar{G}$ of each feature ;
    Compute Hoeffding bound $\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n_l}}$;
    if $\bar{G}_l(X_a) - \bar{G}_l(X_b) > \epsilon$ **then**
      Replace l with an internal node that splits on best feature;
      for *all branches of the split* **do**
        Add a new leaf l' with $n_{ijk}(l') = 0$;
      end
    end
  end
end

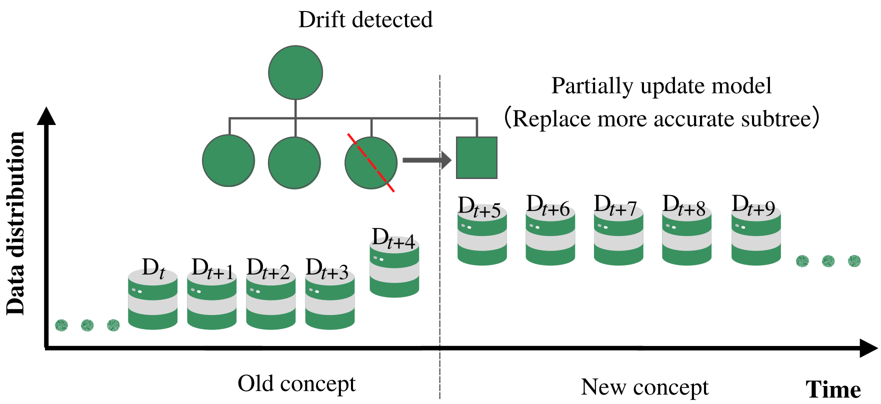

Fig. 2. Mechanisms of HAT adaptation to concept drift.

3.2.2. Hoeffding adaptive tree

Hoeffding Adaptive Tree (HAT) is an extension of the Hoeffding Tree that incorporates a concept drift detector [34]. In our work, we use the HAT with the ADWIN drift detector. This is because the ADWIN is a parameter-free algorithm with a better capability to cope with concept drift compared to the original Hoeffding tree.

Recent studies have shown that individuals who were already vaccinated and infected by the Delta strain of SARS-CoV-2 are also showing some level of protection against the Omicron variant [35]. This indicates that the Omicron variant is a partial mutation that retains some characteristics of the previous strain. Therefore, it is likely that the concept drift caused by the SARS-CoV-2 variant is also partial in nature.

The mechanism of model adaptation to concept drift for the HAT combined with the concept drift detector is shown in Fig. 2. It prunes the old subtrees of the model and replaces them with alternative subtrees with better prediction performance when concept drift occurs. This mechanism of adapting the model to concept drift with partial updates is more suitable for concept drift caused by COVID-19 virus mutations, which are partial in nature. In this way, the model is only partially changed instead of starting from scratch with full retraining.

Bifet and Gavalda proposed ADWIN (Adaptive Windowing), an active detection method for detecting concept drift in data streams [36]. It uses a variable-sized sliding window to compare the difference between two sub-windows based on a Hoeffding bound. This comparison determines whether the target concept has drifted. The window automatically grows when there is no significant change in the data and shrinks when the data changes. The ADWIN algorithm is capable of withstanding drifting data, even if the algorithms are not adapted for it. Concept drift is detected by keeping statistics from a variable-sized window. Using two windows, a statistic's average will be analyzed at different points and the algorithm will determine the size of the window.

3.3. Capturing temporal autocorrelation

The temporal autocorrelation of COVID-19 confirmed cases data has been a major focus in related research. Specifically, this refers to the correlation between the daily number of COVID-19 confirmed cases and the historical daily number of confirmations, which can be expressed mathematically through (6):

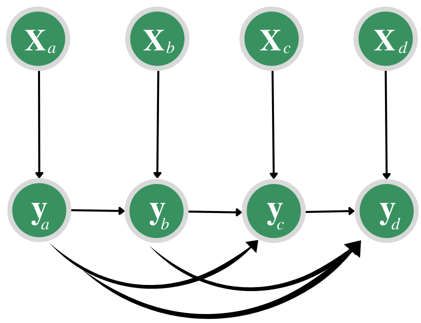

Fig. 3. Multi-task Regressor Chain.

$$y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p}, \tag{6}$$

where y_t represents the number of new confirmed cases on the t-th day, while y_{t-p} represents the number of new confirmed cases on the $(t-p)$-th day. ϕ is a weight given to the historical daily number of confirmations, with ϕ_p representing the weight corresponding to the early p days. We adopt a classic process called *time delay embedding* to capture the temporal autocorrelation in data. Specifically, we use $y_{t-1}, y_{t-2}, \cdots, y_{t-p}$ as the temporal features for the machine learning model. This approach has the advantage of preserving the historical information of the original data, allowing the model to only determine the optimal weighting variables.

3.4. Capturing spatial dependency

We propose a method that incorporates a multi-task chain structure to capture spatial dependencies among prediction variables. This structure is inspired by the Regressor Chain (RC) [37], which is a type of special stack generalization. Chain methods have drawn much attention from the multi-label learning community due to their simple concept and effective performance. The original chain method, Classifier Chain (CC) [38,39], has become a popular method in multi-label learning and has seen success in many real-world applications. Various variants of the chain method have also been developed, such as [40–44].

The key idea behind the original chain method is that the models that come after the first one in the chain use the outputs of the previous models as high-level meta-features, which helps extend the feature space and capture the dependency between the previous outputs and the prediction variables.

3.4.1. Multi task regressor chain

The chain method is originally designed for multi-label learning or multi-target regression. The conventional chain method is not suitable for multi-task learning because each task may have a different feature space. To address this, we propose a multi-task chain method, as illustrated in Fig. 3, where each model has a unique feature space. Our approach builds upon the benefits of regressor chains, allowing it to capture correlations or dependencies between multiple tasks while accommodating different feature spaces for each task. This is in contrast to existing regressor chain methods and their variants, which only work when there are multiple predictor output variables with the same feature space.

The spatial dependency of $\boldsymbol{y}_i$ and $\boldsymbol{y}_j$ can be denoted as their correlation coefficient $c_{i,j}$, $c_{i,j} = corr(\boldsymbol{y}_i, \boldsymbol{y}_j)$. Thus, the correlation coefficient $m \times m$ matrix can be defined by (7),

$$\mathbf{COE} = corr(\mathbf{Y}). \tag{7}$$

By summing each row of $\boldsymbol{y}_i$, we obtain the cumulative correlation value c_i, which indicates the degree of association between $\boldsymbol{y}_i$ and all other variables in $\mathbf{Y}$.

We choose Spearman's rank correlation coefficient to describe the correlation and dependency among COVID-19 news confirmed numbers in different regions. Spearman's rank correlation coefficient has two main advantages: it is parametric-free and does not require any distributional assumptions among the variables. It is represented as (8):

$$r_s = c_{i,j} = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)}, \tag{8}$$

where r_s is the coefficient and n is the number of sample in the $(\boldsymbol{y}_i, \boldsymbol{y}_j)$. For each sample, $\left(y_i, y_j\right)$, the square of the difference in the ranks of the two coordinates is represented by d_i^2, and the sum of each of these squares is represented by the expression $\sum d_i^2$. We sort the order of the target variables y in descending order according to the c of each output variable to obtain the maximum cumulative correlation value for the output of each position in the model chain. This allows us to determine the maximum spatial correlation among COVID-19 news confirmed numbers in different zones.

We propose an algorithm called "ChatRC" that integrates all of the previously described components to address the three challenges. The algorithm is presented in Algorithm 2.

Algorithm 2: Correlated Hoeffiding Adaptive Tree Regressor Chain (ChatRC).

```
Data: History training dataset D = (X_history, y_history), and stream data (X, y)
Result: multi-task regression model h_j, j = {1, ..., m}
COE_{m×m} = corr(Y) ;
c_l = ∑ COE(:, l), l = {1, ..., m} ;
for y_i in y_{1,m} do
    Let HT_i be a tree with a single leaf (the root) for the output y_i;
    Create the detector ADWIN and initialize estimators A_ijk = 0;
end
for all training examples (x, y) in stream do
    for each y_i in y do
        if i > 1 then
            x = x_i ∪ (y_1 ··· y_{i-1})
        end
        Sort example (x, y_j) into leaf l using HT_i;
        Update sufficient statistics in l;
        Increment the number of examples seen at l, n_l ;
        if variance of these labels at the leaf node increases then
            Calculating Ḡ of each feature ;
            Compute Hoeffding bound ε = sqrt(R^2 ln(1/δ) / (2n_l)) ;
            if Ḡ_l(X_a) − Ḡ_l(X_b) > ε then
                Replace l with an internal node that splits on best feature;
                for all branches of the split do
                    Add a new leaf l' with n_ijk(l') = 0;
                end
            end
        end
        if Concept drift is detected by ADWIN drift detector then
            Create an alternate tree with the new best feature at its root, if there is none;
        end
        if the alternate subtree is more accurate then
            Replace the current node with this subtree;
        end
    end
end
```

4. Experimental study

This section introduces the concept of experimental design, the dataset, and the experiment setup, including the baseline. Finally, we present the results of the experiment and the corresponding analysis.

4.1. Experimental design

The experiments were designed to verify the performance of our proposed method in COVID-19 spatio-temporal prediction under concept drift. In this paper, we designed several experiments to answer the following research questions:

1. Does the model with the ability to adapt to concept drift have higher prediction performance in COVID-19 time series prediction?
2. Does the model with the ability of chain structure to capture spatial correlation have higher prediction performance in COVID-19 time series prediction?
3. Do online forecasting models produce higher forecasting performance than offline batch forecasting models in COVID-19 time series prediction?

To verify the effectiveness of our proposed methods, we selectd several state-of-the-art deep learning methods in related works, such as N-Beats, DeepAR, TCN, and LSTM.

4.2. Data description

The study used data sourced from JHU CSSE COVID-19 Data [45]. We focused on the state of California in the United States and selected all counties under its jurisdiction with an average daily infection rate of over 100. Specifically, we examined the following counties in California: Riverside, San Joaquin, Tulare, Solano, ContraCosta, Kern, Orange, Sacramento, San Bernardino, SantaClara, Ventura, Fresno, San Mateo, Alameda, Monterey, Stanislaus, Los Angeles, San Diego, San Francisco. There are twenty target zones in total, which we aimed to predict. The time series data covers the period from March 4, 2020, to January 19, 2021, with daily frequency. There are a total of 322 samples, each containing the values of twenty prediction zones.

4.3. Experiment setup

Our proposed algorithm offers the advantage of simultaneously capturing spatial correlation and adapting to concept drift. To verify this claim, we compared our proposed algorithm with ChtRC and Mhat algorithms. In our COVID-19 time series prediction study, the online machine learning model demonstrated better prediction performance than the offline batch machine learning model. To confirm this finding, we selected several state-of-the-art batch learning deep learning algorithms in related works as comparison models, including N-Beats, TCN, DeepAR, and LSTM.

1. **ChtRC:** ChtRC stands for Correlated Hoeffding Tree Regressor Chain. The only difference between ChtRC and ChatRC is that it does not have the ADWIN drift detector, and therefore, it does not have the ability to adapt to concept drift.
2. **Mhat:** Mhat stands for Multi Hoeffding Adaptive Tree. The only difference between Mhat and ChatRC is that it does not have a chain structure on models for capturing spatial information. It is composed of Multi-label single Hoeffding Adaptive Tree and does not have any structure on output.
3. **N-Beats:** N-Beats is a deep learning model for time series forecasting [46]. The model learns different levels of abstraction from the data, allowing it to detect both short-term and long-term patterns in the data. In addition, N-Beats is trained with a novel objective, which enables it to make multi-step ahead predictions, making it suitable for applications that require long-term predictions. N-Beats has achieved state-of-the-art performance on a wide range of time series forecasting tasks, such as energy demand forecasting and stock price prediction.
4. **TCN:** TCN (Temporal Convolutional Network) is a type of deep learning model designed for processing time series data [47]. It uses a variant of the Convolutional Neural Network (CNN) architecture. TCNs learn spatial and temporal patterns from input data by applying convolutional filters. Compared to conventional time series models, TCNs have several advantages. They can learn complex patterns in the data and make more accurate predictions. Moreover, they can handle large amounts of data effectively. Furthermore, TCNs can be trained using end-to-end learning, meaning they can learn directly from the data without being hand-coded. TCNs are a powerful tool for processing time series data and making predictions based on that data. They have been applied successfully to a wide range of time series prediction tasks, including forecasting, anomaly detection, and classification. Sciannameo et al. also applied TCNs to COVID-19 spatial-temporal prediction with promising results [8].
5. **DeepAR:** DeepAR is a deep learning model for time series forecasting developed by Amazon Web Services [48]. To model temporal dependencies, it uses RNN and Long Short-Term Memory (LSTM). Multivariate time series are easy to forecast with DeepAR. To make predictions, it uses both global and local models, enabling it to learn from the patterns in the whole dataset and also take into account the individual characteristics of each series. Therefore, DeepAR is perfect for forecasting multiple products or multiple roads with many related time series. DeepAR shows state-of-the-art performance on a wide range of time series forecasting tasks, including demand forecasting, supply chain optimization, and financial forecasting.
6. **LSTM:** The Long Short-Term Memory (LSTM) is a popular method for predicting the number of daily confirmed COVID-19 cases because it is a powerful tool for modeling temporal autocorrelation in the data. LSTMs are a variant of RNNs that utilize gates to regulate the information flow through the network, enabling them to capture long-term dependencies in the data. In a time series of daily COVID-19 confirmed cases, there may be complex patterns that are influenced by factors such as the effectiveness of public health measures, individual behavior, and the emergence of new variants of the virus. LSTMs are well-suited to this type of prediction problem because they can learn the dependencies in the data and make accurate predictions about future cases. Furthermore, LSTMs are widely used in many applications and have demonstrated state-of-the-art performance on a broad range of time series prediction tasks, making them a popular choice for predicting daily confirmed cases of COVID-19.

All deep learning baselines are implemented using TensorFlow. The Hoeffding Tree and HAT are implemented using River [49]. For a fair comparison, all parameters of each baseline are carefully tuned according to the recommended settings. We have selected the data from 4 March 2020 to 20 December 2020 as the training data, while the data from 21 December 2020 to 19 January 2021 is used as the test data. To evaluate the predictive performance of these multi-output methods, we have calculated the average root-mean-square error (aRMSE) as (9),

$$\text{aRMSE} = \frac{1}{m}\sum_{m}^{t=1}\sqrt{\frac{\sum_{i=1}^{N}\left(y_i^t - \hat{y}_i^t\right)^2}{N}}, \tag{9}$$

average mean absolute error (aMAE) as (10),

$$\text{aMAE} = \frac{1}{m}\sum_{m}^{t=1}\frac{1}{N}\sum_{N}^{i=1}\left|y_i^t - \hat{y}_i^t\right|, \tag{10}$$

and average mean-square log error (aMSLE) as (11),

$$\text{aMSLE} = \frac{1}{m}\sum_{m}^{t=1}\frac{1}{N}\sum_{N}^{i=1}\left(\log(y_i^t + 1) - \log(\hat{y}_i^t + 1)\right)^2. \tag{11}$$

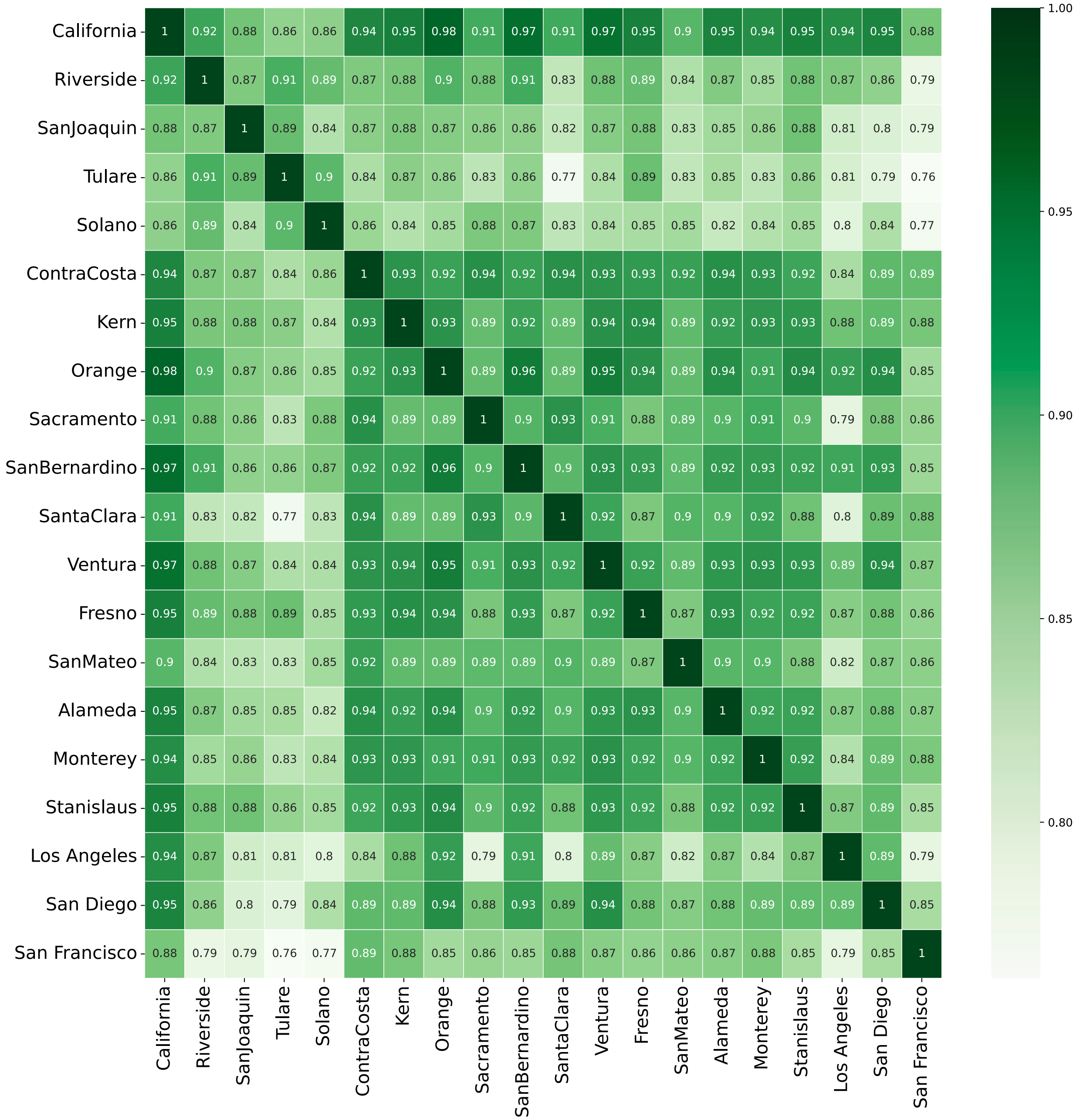

Fig. 4. Heat map of Spearman's rank correlation coefficient matrix between new confirmed for all zones.

4.4. Spatial dependency analysis

Our study reveals significant spatial dependencies in the spread of COVID-19, which is a critical aspect of our proposed prediction model. By capturing these dependencies, our model can provide more accurate and nuanced predictions. Fig. 4 presents a heatmap generated by Spearman's Correlation Coefficient matrix, where higher values indicate stronger monotonicity. The heatmap shows a substantial correlation between the number of daily confirmed COVID-19 cases in each city, which can be attributed to two aspects. Firstly, the similarity in symptoms is due to the same or similar pattern of viral transmission over time. Secondly, the regional mobility of virus carriers and infected individuals can result in an interaction effect on the number of confirmed cases. Fig. 5 shows the cumulative correlations between the target variable and all others. It gives the following order listed in c: 'California' ≻ 'Riverside' ≻ 'SanJoaquin' ≻ 'Tulare' ≻ 'Solano' ≻ 'ContraCosta' ≻ 'Kern' ≻ 'Orange' ≻ 'Sacramento' ≻ 'SanBernardino' ≻ 'SantaClara' ≻ 'Ventura' ≻ 'Fresno' ≻ 'SanMateo' ≻ 'Alameda' ≻ 'Monterey' ≻ 'Stanislaus' ≻ 'Los Angeles' ≻ 'San Diego' ≻ 'San Francisco'. The cumulative

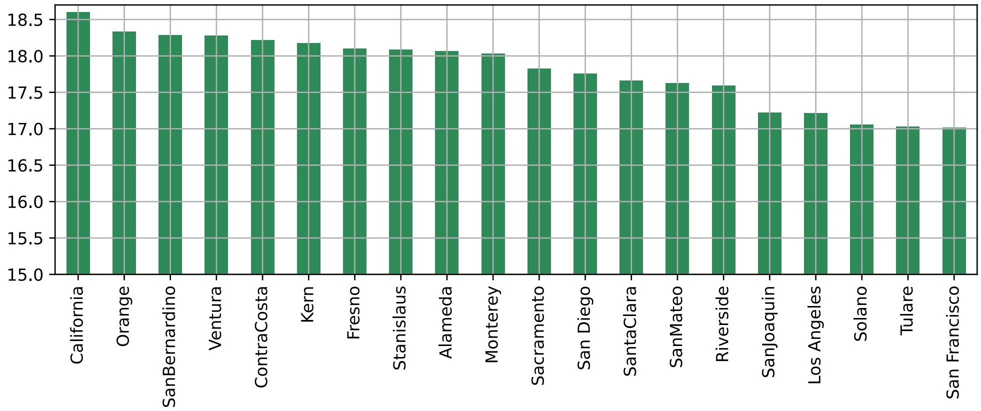

Fig. 5. The cumulative correlations between the target variable and others.

spatial correlations or dependence of 'California' are ranked first, undoubtedly because it refers to the total daily new confirmed cases in the state.

Based on Fig. 4 and 5, we can analyze the causes of the outcomes depicted from multiple views. By taking into account factors such as geographic proximity, coastal location, population density, public health measures, and others, we can better understand the reasons behind the observed synchronicity in the daily COVID-19 caseload among various counties in California.

Our analysis of the spatial dependencies in COVID-19 cases across various counties in California reveals some interesting patterns. These patterns, which are likely influenced by factors such as geographic proximity, coastal location, population density, and public health measures, provide valuable insights into the spread of the virus.

The highest correlation value of 0.98 between Orange County and California suggests a strong monotonic relationship between their respective COVID-19 daily cases. Orange County, which is the third-most populous county in California [50] and the sixth-most populous in the United States, has several similarities with California, such as a coastal location and high population density. It is also the second-most densely populated county in the state behind San Francisco [51]. Orange County is located along the Southern California coast, and California boasts a significant coastal region, with many of its largest cities situated along the coast. Interestingly, despite not being the most densely populated or populous city in California, Orange County exhibits a strong monotonic relationship with the state in terms of COVID-19 daily cases, with the highest correlation value of 0.98. This suggests that the rate of spread of the epidemic in Orange County is similar to the rate of spread in the majority of other cities in California. These similarities may contribute to the observed correlation.

The three cities with the highest correlation values to California as a whole are Orange, San Bernardino, and Ventura, with correlation values of 0.98, 0.97, and 0.97, respectively. These three counties in California share several similarities, such as their coastal location and high population density. Orange County is situated on the Southern California coast, while San Bernardino County is located east of Los Angeles, and Ventura County is located north of Los Angeles along the Pacific coast. All three counties experience a similar maritime climate due to their coastal location. Additionally, these counties have a significant number of residents, with Orange County being the third-most populous county in California, San Bernardino County being the fifth-most populous county in the state, and Ventura County being the 14th-most populous. These similarities may contribute to the observed monotonic correlation in the daily COVID-19 cases between these counties and the state.

The lowest correlation value of 0.76 between San Francisco and Tulare counties suggests relatively low synchronicity between their COVID-19 daily cases. San Francisco is a densely populated coastal county, while Tulare is an inland county with lower population density. Additionally, San Francisco is part of the San Francisco Bay Area, which has a strong technology industry, while Tulare is a major agricultural county. These differences in population density and geographic location may contribute to the observed low synchronicity in their COVID-19 daily cases. The city with the highest synchronicity with San Francisco, Contra Costa, has a correlation value of 0.89 and is a coastal city in close proximity to San Francisco. This proximity can lead to increased movement of people, goods, and services between these cities and San Francisco, which may contribute to the higher synchronicity. They may also share similar industries, such as tourism and shipping, and experience a similar maritime climate.

Taking Los Angeles as another example, the four cities with the highest monotonicity with Los Angeles are Orange, San Bernardino, Ventura, and San Diego, with correlation values of 0.92, 0.91, 0.89, and 0.89, respectively. These four cities are relatively close to Los Angeles, with Orange, Ventura, and San Diego all being densely populated coastal cities. San Bernardino, on the other hand, is not a coastal city but still has a high correlation with Los Angeles. The high population density in these cities may affect the transmission dynamics of COVID-19, as it can lead to increased human contact and higher transmission rates.

Understanding the complex interplay of these factors is crucial for effective public health strategies. However, as this paper focuses on prediction, we provide only a brief analysis of these aspects. In terms of prediction, the later meta-models employ higher-level information from across the entire state of California, enhancing prediction accuracy by utilizing hierarchical forecasting data. Furthermore, the spatial dependencies among regions allow these meta-models to extract valuable insights, thus improving prediction precision. The arrangement of these predictive meta-models enables the overarching model to effectively maximize spatial

Table 2
Performance comparison of the proposed model and its benchmarks.

	aRMSE	aMAE	aMSLE
ChatRC	**443.35**	**114.85**	**0.14**
ChtRC	630.69	156.84	0.18
Mhat	645.26	162.92	0.19
N-Beats	9375.44	3417.35	2.73
TCN	1254.87	822.62	0.57
DeepAR	9539.31	3800.75	28.43
LSTM	9507.02	3740.12	9.69

dependence or correlation, potentially leading to increased prediction performance. This is a key strength of our proposed model, as it allows us to leverage spatial dependencies to improve prediction accuracy. The use of spatial correlation to improve predictive performance is also evidenced in related works [8,20].

4.5. *Performance comparisons with benchmarks*

4.5.1. *Adapting to concept drift can improve performance*

Our study demonstrates the importance of adapting to concept drift in improving the performance of predictive models. To assess if algorithms adapting to concept drift are better suited for predicting new COVID-19 confirmed cases, we compared ChatRC with ChtRC, where ChtRC is ChatRC without the ADWIN drift detector. According to Table 2, we can see that ChatRC outperforms ChtRC in all three evaluation metrics, i.e., aRMSE, aMAE, and aMSLE. Specifically, ChatRC achieves an aRMSE of 443.35, which is significantly lower than ChtRC's aRMSE of 630.69. Similarly, ChatRC's aMAE and aMSLE values are also lower than those of ChtRC. These results suggest that incorporating the ADWIN concept drift detector in ChatRC can improve its prediction accuracy compared to the model without the detector (ChtRC).

Moreover, concept drift is a common phenomenon in COVID-19 daily case time series data. For example, Omicron spreads more effectively than the original strain, leading to unsatisfactory prediction performance from models based on the original strain during the Omicron propagation period. ChatRC's ADWIN detector enables the model to adapt to concept drift by replacing the original subtree with a more accurate one. This partial model update leads to better prediction accuracy, allowing the model to capture changing patterns in the data more effectively. In contrast, ChtRC lacks this ability to adapt to concept drift, which could explain its inferior performance compared to ChatRC. Thus, the performance comparison between ChatRC and ChtRC indicates that adapting to concept drift through the ADWIN detector can improve the model's prediction accuracy. The suitability of online machine learning for addressing this problem also confirms the impact of concept drift on model prediction performance.

4.5.2. *Capturing spatial dependency can improve performance*

Our study also highlights the importance of capturing spatial dependencies in improving the performance of predictive models. To evaluate whether algorithms capturing the spatial dependency between daily confirmed COVID-19 cases across regions can enhance predictive performance, we compared ChatRC and Mhat. Mhat is ChatRC without a chain structure for capturing spatial dependency, meaning that it is just multiple single Hoeffding adaptive trees for each single target without modeling them together. Mhat lacks the ability to capture spatial dependency among multiple regional COVID-19 caseloads. According to Table 2, ChatRC outperforms Mhat with a lower aRMSE of 443.35 compared to Mhat at 645.26, a lower aMAE of 114.85 compared to Mhat at 162.92, and a lower aMSLE of 0.14 compared to Mhat at 0.19. This result indicates that ChatRC, which captures the spatial dependency between daily confirmed COVID-19 cases across regions, enhances predictive performance compared to Mhat, which does not capture spatial dependencies.

ChatRC captures the spatial dependency between daily confirmed COVID-19 cases across regions by using a chain structure. This allows the model to consider the relationship between the daily case numbers in neighboring regions and incorporate that information into its predictions. On the other hand, Mhat is based on multiple single Hoeffding adaptive trees for each single target, without modeling the regions together. This approach lacks the ability to capture spatial dependency among multiple regional COVID-19 caseloads.

In addition, ChatRC's performance improvement can be attributed to the inherent spatial dependence and high degree of monotonicity in the number of daily COVID-19 confirmed cases in neighboring regions, as demonstrated by the Spearman correlation analysis in Fig. 4. By incorporating spatial information into its predictions, ChatRC can take advantage of this correlation and improve its prediction accuracy.

This comparison underscores the strength of our proposed model, ChatRC, in capturing spatial dependencies in the spread of COVID-19. By using a chain structure, our model can consider the relationship between daily case numbers in neighboring regions and incorporate that information into its predictions, leading to improved predictive performance. This is a significant advantage over models that do not capture spatial dependencies, as demonstrated by the inferior performance of Mhat.

4.5.3. *Online learning more suitable*

Our study also highlights the importance of capturing spatial dependencies in improving the performance of predictive models. To determine whether online machine learning outperforms batch machine learning in predicting daily confirmed COVID-19 cases, we compared various batch machine learning algorithms with online learning algorithms. The batch learning algorithms we tested were N-Beats, TCN, DeepAR, and LSTM, while the online learning models included ChatRC, ChtRC, and Mhat.

Among the online learning models, ChatRC achieved the best performance, with an aRMSE of 443.35, aMAE of 114.85, and aMSLE of 0.14. ChtRC and Mhat achieved slightly worse performance, with aRMSE of 630.69 and 645.26, aMAE of 156.84 and 162.92, and aMSLE of 0.18 and 0.19, respectively. On the other hand, among the conventional batch machine learning models, LSTM achieved the best performance, with an aRMSE of 9507.02, aMAE of 3740.12, and aMSLE of 9.69. The other models in batch learning algorithms — N-Beats, TCN, and DeepAR — all had significantly worse performance, with aRMSE ranging from 1254.87 to 9375.44, aMAE ranging from 822.62 to 3417.35, and aMSLE ranging from 0.57 to 28.43. These ranges of errors further emphasize the superiority of online learning models, particularly ChatRC, over conventional batch machine learning group models in predicting rapidly changing data patterns, such as those seen in the daily confirmed cases of COVID-19.

Online machine learning is particularly well-suited for dynamic, non-stationary time series data, which often exhibit concept drift or changes in the underlying data distribution over time. This is because online learning algorithms can continuously learn and update their models in response to new data, without requiring the entire dataset to be reprocessed. In contrast, conventional batch learning methods require the entire dataset to be processed during training, and do not adapt well to changing patterns in the data. In the case of COVID-19 time series data, the patterns and underlying data distributions have been changing rapidly due to various factors such as virus mutations, policy changes, and vaccination campaigns. Therefore, online machine learning algorithms like ChatRC, ChtRC, and Mhat are better suited for predicting the number of daily confirmed COVID-19 cases compared to conventional batch learning algorithms like N-Beats, TCN, DeepAR, and LSTM.

Moreover, since all conventional batch machine learning models used here are the deep learning family and the sample size is relatively small at 322, it is not surprising that these models have lower predictive performance than the online learning group models. Deep learning models typically require a large amount of training data to achieve high performance due to their high model capacity and the large number of parameters to be learned. Therefore, training deep learning models with a small sample size may result in overfitting, where the model is too complex and tailored to the training data, causing poor generalization performance on new data.

In contrast, online learning models update their models with each new data point, allowing them to adapt quickly to changing patterns in the data. This is particularly important for predicting COVID-19 daily confirmed cases, where the data patterns can change rapidly due to various factors, such as virus mutations and policy changes. Online learning models also tend to have lower model capacity and fewer parameters, making them more suitable for training with a smaller sample size.

This comparison highlights the strength of our proposed model, ChatRC, and other online learning models in handling dynamic, non-stationary time series data, such as the daily confirmed cases of COVID-19. Our model and other online learning models can continuously learn and update their models in response to new data, leading to improved prediction accuracy. This is a significant advantage over conventional batch learning methods, which require the entire dataset to be processed during training and do not adapt well to changing patterns in the data.

5. Conclusion

The ChatRC proposed in this paper is capable of capturing both temporal autocorrelations and spatial dependencies in data, updating the model, and making more accurate predictions than conventional machine learning models in real-time scenarios with concept drift. Online machine learning is better suited for predicting COVID-19 than conventional machine learning, and this improvement in prediction performance due to the learning mechanism is significant. In addition to temporal autocorrelation, capturing spatial dependence can also enhance the model's predictive performance. The high monotonic correlation of trend data between multiple regions of COVID-19 in California, USA, used in this paper is one of the key reasons why capturing spatial correlation improves the model's prediction performance.

The limitations of this study can be analyzed from multiple perspectives, particularly with respect to addressing the three main challenges of COVID-19 daily case prediction: concept drift, temporal autocorrelation, and spatial dependence. Future applications of our algorithms for 3D engineering tasks will also be interesting and worth exploring [52,53].

In terms of handling conceptual drift, this paper employs a Hoeffding tree with ADWIN concept drift detectors. While the Hoeffding tree's performance has demonstrated advantages in numerous works, this integrated solution may not be the optimal choice. As the Hoeffding tree is inherently a tree-based method, it cannot output prediction intervals using quantile loss, providing only a single prediction. The ADWIN concept drift detector is a generic detector, and although our experiments have shown it improves the model's predictive performance, we have not compared it to other concept drift detectors.

In terms of capturing spatial dependence, our experiments have shown the effectiveness of the chain structure. However, we have not compared it to the increasingly popular graph neural networks and graph algorithms, which may be better suited for capturing spatial dependence using graph-based components.

Another limitation of this study pertains to its scope. While the research focuses on designing the most suitable data-driven algorithm for predicting daily new COVID-19 cases, other daily data related to COVID-19, such as the number of deaths and recoveries, may be correlated. Joint modeling of these variables could potentially yield better predictions and extract useful information from data correlations. Future work should consider exploring these aspects to further improve prediction models and gain deeper insights into the relationships among various COVID-19 data types.

Author contribution statement

Zipeng Wu: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper. **Chu Kiong Loo; Unaizah Hanum Binti Obaidellah; Kitsuchart Pasupa**: Conceived and designed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Data availability statement

Data included in article/supp. material/referenced in article.

Additional information

No additional information is available for this paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by King Mongkut's Institute of Technology Ladkrabang Research Fund [KREF206306].

References

[1] P. Jha, P.E. Brown, Rashid Ansumana, Counting the global COVID-19 dead, Lancet 399 (10399) (2022) 1937–1938.
[2] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, G. Zhang, Learning under concept drift: a review, IEEE Trans. Knowl. Data Eng. 31 (12) (2019) 2346–2363, https://doi.org/10.1109/TKDE.2018.2876857.
[3] S. Sonoda, N. Murata, Transport analysis of infinitely deep neural network, J. Mach. Learn. Res. 20 (2019) 2:1–2:52.
[4] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Trans. Intell. Transp. Syst. 15 (5) (2014) 2191–2201, https://doi.org/10.1109/TITS.2014.2311123.
[5] W. Tang, F. He, Y. Liu, Y. Duan, MATR: multimodal medical image fusion via multiscale adaptive transformer, IEEE Trans. Image Process. 31 (2022) 5134–5149, https://doi.org/10.1109/TIP.2022.3193288.
[6] J. Zhang, F. He, Y. Duan, S. Yang, Aidednet: anti-interference and detail enhancement dehazing network for real-world scenes, Front. Comput. Sci. 17 (2) (2022) 172703, https://doi.org/10.1007/s11704-022-1523-9.
[7] M.H.D.M. Ribeiro, R.G. da Silva, V.C. Mariani, L. dos Santos Coelho, Short-term forecasting COVID-19 cumulative confirmed cases: perspectives for Brazil, Chaos Solitons Fractals 135 (2020), https://doi.org/10.1016/j.chaos.2020.109853.
[8] V. Sciannameo, A. Goffi, G. Maffeis, R. Gianfreda, D.J. Pagliari, T. Filippini, P. Mancuso, P. Giorgi-Rossi, L.A.D. Zovo, A. Corbari, M. Vinceti, P. Berchialla, A deep learning approach for spatio-temporal forecasting of new cases and new hospital admissions of COVID-19 spread in reggio emilia, northern Italy, J. Biomed. Inform. 132 (2022) 104132, https://doi.org/10.1016/j.jbi.2022.104132.
[9] V. Papastefanopoulos, P. Linardatos, S. Kotsiantis, COVID-19: a comparison of time series methods to forecast percentage of active cases per population, Appl. Sci. (Switzerland) 10 (2020) 1–15, https://doi.org/10.3390/app10113880.
[10] V.K.R. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using lstm networks, Chaos Solitons Fractals 135 (2020) 109864, https://doi.org/10.1016/j.chaos.2020.109864.
[11] A. Zeroual, F. Harrou, A. Dairi, Y. Sun, Deep learning methods for forecasting COVID-19 time-series data: a comparative study, Chaos Solitons Fractals 140 (2020), https://doi.org/10.1016/j.chaos.2020.110121.
[12] T. Wang, Z. Zhang, K. Tsui, PSTN: periodic spatial-temporal deep neural network for traffic condition prediction, CoRR, arXiv:2108.02424 [abs], 2021.
[13] N. Zheng, S. Du, J. Wang, H. Zhang, W. Cui, Z. Kang, T. Yang, B. Lou, Y. Chi, H. Long, M. Ma, Q. Yuan, S. Zhang, D. Zhang, F. Ye, J. Xin, Predicting COVID-19 in China using hybrid ai model, IEEE Trans. Cybern. 50 (2020) 2891–2904, https://doi.org/10.1109/TCYB.2020.2990162.
[14] A. Mahajan, N.A. Sivadas, R. Solanki, An epidemic model sipherd and its application for prediction of the spread of COVID-19 infection in India, Chaos Solitons Fractals 140 (2020) 110156, https://doi.org/10.1016/j.chaos.2020.110156.
[15] R. Salgotra, M. Gandomi, A.H. Gandomi, Time series analysis and forecast of the COVID-19 pandemic in India using genetic programming, Chaos Solitons Fractals 138 (2020), https://doi.org/10.1016/j.chaos.2020.109945.
[16] M. Maleki, M.R. Mahmoudi, D. Wraith, K.H. Pho, Time series modelling to forecast the confirmed and recovered cases of COVID-19, Trav. Med. Infect. Dis. 37 (2020) 101742, https://doi.org/10.1016/j.tmaid.2020.101742.
[17] S. Zio, B. Lamien, S. Tiemounou, Y. Adaman, I. Tougri, M. Beidari, O.W. Boris, Multi-outputs Gaussian process for predicting Burkina Faso COVID-19 spread using correlations from the weather parameters, Infect. Dis. Model. 7 (2022) 448–462, https://doi.org/10.1016/j.idm.2022.06.006.
[18] R.R. Sharma, M. Kumar, S. Maheshwari, K.P. Ray, Evdhm-arima-based time series forecasting model and its application for COVID-19 cases, IEEE Trans. Instrum. Meas. 70 (2021) 1–10, https://doi.org/10.1109/TIM.2020.3041833.
[19] İsmail Kırbaş, A. Sözen, A.D. Tuncer, F. Şinasi Kazancıoğlu, Comparative analysis and forecasting of COVID-19 cases in various European countries with arima, narnn and lstm approaches, Chaos Solitons Fractals 138 (2020), https://doi.org/10.1016/j.chaos.2020.110015.
[20] V.L. Gatta, V. Moscato, M. Postiglione, G. Sperlì, An epidemiological neural network exploiting dynamic graph structured data applied to the COVID-19 outbreak, IEEE Trans. Big Data 7 (1) (2021) 45–55, https://doi.org/10.1109/TBDATA.2020.3032755.
[21] P. Wang, X. Zheng, G. Ai, D. Liu, B. Zhu, Time series prediction for the epidemic trends of COVID-19 using the improved lstm deep learning method: case studies in Russia, Peru and Iran, Chaos Solitons Fractals 140 (2020) 110214, https://doi.org/10.1016/j.chaos.2020.110214.
[22] V. Chaurasia, S. Pal, Application of machine learning time series analysis for prediction COVID-19 pandemic, Res. Biomed. Eng. (2020), https://doi.org/10.1007/s42600-020-00105-4.

[23] H. Abbasimehr, R. Paki, Prediction of COVID-19 confirmed cases combining deep learning methods and bayesian optimization, Chaos Solitons Fractals 142 (2021) 110511, https://doi.org/10.1016/j.chaos.2020.110511.
[24] M. Mousavi, R. Salgotra, D.S. Holloway, A.H. Gandomi, COVID-19 time series forecast using transmission rate and meteorological parameters as features, IEEE Comput. Intell. Mag. 15 (4) (2020) 34–50, https://doi.org/10.1109/MCI.2020.3019895.
[25] D. Parbat, M. Chakraborty, A python based support vector regression model for prediction of COVID-19 cases in India, Chaos Solitons Fractals 138 (2020) 109942, https://doi.org/10.1016/J.CHAOS.2020.109942.
[26] P.G. Jamdade, S.G. Jamdade, Modeling and prediction of COVID-19 spread in the Philippines by October 13, 2020, by using the varmax time series method with preventive measures, Results Phys. 20 (2021) 103694, https://doi.org/10.1016/j.rinp.2020.103694.
[27] A. Ashofteh, J.M. Bravo, M. Ayuso, An ensemble learning strategy for panel time series forecasting of excess mortality during the COVID-19 pandemic, Appl. Soft Comput. 128 (2022) 109422, https://doi.org/10.1016/j.asoc.2022.109422.
[28] M. Liu, R. Thomadsen, S. Yao, Forecasting the spread of COVID-19 under different reopening strategies, Sci. Rep. 10 (2020) 1–8, https://doi.org/10.1038/s41598-020-77292-8.
[29] M.S. Satu, K.C. Howlader, M. Mahmud, M.S. Kaiser, S.M.S. Islam, J.M. Quinn, S.A. Alyami, M.A. Moni, Short-term prediction of COVID-19 cases using machine learning models, Appl. Sci. (Switzerland) 11 (2021), https://doi.org/10.3390/app11094266.
[30] T. Uchida, K. Yoshida, Concept drift in Japanese COVID-19 infection data, Proc. Comput. Sci. 207 (2022) 380–387, https://doi.org/10.1016/J.PROCS.2022.09.072.
[31] V.Z. Marmarelis, Predictive modeling of COVID-19 data in the US: adaptive phase-space approach, IEEE Open J. Eng. Medicine Biol. 1 (2020) 207–213, https://doi.org/10.1109/ojemb.2020.3008313.
[32] A. Tomar, N. Gupta, Prediction for the spread of COVID-19 in India and effectiveness of preventive measures, Sci. Total Environ. 728 (2020) 138762, https://doi.org/10.1016/j.scitotenv.2020.138762.
[33] P.M. Domingos, G. Hulten, Mining high-speed data streams, in: R. Ramakrishnan, S.J. Stolfo, R.J. Bayardo, I. Parsa (Eds.), Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, USA, August 20-23, 2000, ACM, Boston, MA, USA, 2000, pp. 71–80, https://doi.org/10.1145/347090.347107.
[34] A. Bifet, R. Gavaldà, Adaptive learning from evolving data streams, in: N.M. Adams, C. Robardet, A. Siebes, J. Boulicaut (Eds.), Advances in Intelligent Data Analysis VIII, 8th International Symposium on Intelligent Data Analysis, IDA 2009, Lyon, France, August 31 - September 2, 2009, in: Lecture Notes in Computer Science, vol. 5772, Springer, 2009, pp. 249–260, https://doi.org/10.1007/978-3-642-03915-7_22.
[35] M. Shah, H.G. Woo, Omicron: a heavily mutated sars-cov-2 variant exhibits stronger binding to ace2 and potently escapes approved COVID-19 therapeutic antibodies, Front. Immunol. 12 (2022), https://doi.org/10.3389/fimmu.2021.830527.
[36] A. Bifet, R. Gavaldà, Learning from time-changing data with adaptive windowing, in: Proceedings of the Seventh SIAM International Conference on Data Mining, April 26-28, 2007, SIAM, Minneapolis, Minnesota, USA, 2007, pp. 443–448, https://doi.org/10.1137/1.9781611972771.42.
[37] E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas, Multi-target regression via input space expansion: treating targets as inputs, Mach. Learn. 104 (2016) 55–98, https://doi.org/10.1007/s10994-016-5546-z.
[38] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, Mach. Learn. 85 (2011) 333–359, https://doi.org/10.1007/s10994-011-5256-5.
[39] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains: a review and perspectives, J. Artif. Intell. Res. 70 (2021) 683–718, https://doi.org/10.1613/JAIR.1.12376.
[40] G. Melki, A. Cano, V. Kecman, S. Ventura, Multi-target support vector regression via correlation regressor chains, Inf. Sci. 415–416 (2017) 53–69, https://doi.org/10.1016/j.ins.2017.06.017.
[41] C. Geiß, E. Brzoska, P.A. Pelizari, S. Lautenbach, H. Taubenböck, Multi-target regressor chains with repetitive permutation scheme for characterization of built environments with remote sensing, Int. J. Appl. Earth Obs. Geoinf. 106 (2022), https://doi.org/10.1016/j.jag.2021.102657.
[42] S.M. Mastelini, V.G.T. da Costa, E.J. Santana, F.K. Nakano, R.C. Guido, R. Cerri, S. Barbon, Multi-output tree chaining: an interpretative modelling and lightweight multi-target approach, J. Signal Process. Syst. 91 (2019) 191–215, https://doi.org/10.1007/s11265-018-1376-5.
[43] Z. Wu, G. Lian, A novel dynamically adjusted regressor chain for taxi demand prediction, in: 2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020, IEEE, 2020, https://doi.org/10.1109/IJCNN48605.2020.9207160.
[44] J. Read, L. Martino, Probabilistic regressor chains with Monte Carlo methods, Neurocomputing 413 (2020) 471–486, https://doi.org/10.1016/j.neucom.2020.05.024.
[45] E. Dong, H. Du, L. Gardner, An interactive web-based dashboard to track COVID-19 in real time, Lancet Infect. Dis. 20 (2020) 533–534, https://doi.org/10.1016/S1473-3099(20)30120-1.
[46] B.N. Oreshkin, D. Carpov, N. Chapados, Y. Bengio, N-BEATS: neural basis expansion analysis for interpretable time series forecasting, in: 8th International Conference on Learning Representations, ICLR 2020, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, 2020, pp. 1–21.
[47] S. Bai, J.Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, CoRR, arXiv:1803.01271 [abs], 2018.
[48] D. Salinas, V. Flunkert, J. Gasthaus, T. Januschowski, Deepar: probabilistic forecasting with autoregressive recurrent networks, Int. J. Forecast. 36 (2020) 1181–1191, https://doi.org/10.1016/j.ijforecast.2019.07.001.
[49] J. Montiel, M. Halford, S.M. Mastelini, G. Bolmier, R. Sourty, R. Vaysse, A. Zouitine, H.M. Gomes, J. Read, T. Abdessalem, A. Bifet, River: machine learning for streaming data in python, J. Mach. Learn. Res. 22 (2021) 4945–4952.
[50] County statistics of the United States of orange county, https://data.census.gov/profile?g=050XX00US06059, 2022. (Accessed 29 July 2023), an official website of the United States government.
[51] Decennial census https://www.census.gov/data/developers/data-sets/decennial-census.html, 2022. (Accessed 29 July 2023) [an official website of the United States government].
[52] Y. Liang, F. He, X. Zeng, J. Luo, An improved loop subdivision to coordinate the smoothness and the number of faces via multi-objective optimization, Integr. Comput.-Aided Eng. 29 (1) (2022) 23–41, https://doi.org/10.3233/ICA-210661.
[53] Y. Wu, F. He, D. Zhang, X. Li, Service-oriented feature-based data exchange for cloud-based design and manufacturing, IEEE Trans. Serv. Comput. 11 (2) (2018) 341–353, https://doi.org/10.1109/TSC.2015.2501981.