Knowledge-Based Systems 233 (2021) 107417

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Hybrid deep learning of social media big data for predicting the evolution of COVID-19 transmission

Alvin Wei Ze Chew [a,1], Yue Pan [b,1], Ying Wang [c], Limao Zhang [c,*]

[a] Bentley Systems Research Office, 1 Harbourfront Pl, HarbourFront Tower One, Singapore 098633, Singapore
[b] Shanghai Key Laboratory for Digital Maintenance of Buildings and Infrastructure, Department of Civil Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai, China
[c] School of Civil and Environmental Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Received 1 March 2021
Received in revised form 14 July 2021
Accepted 18 August 2021
Available online 24 August 2021

Keywords:
COVID-19
Natural language processing
Time-series prediction
Deep learning
Big-data

ABSTRACT

In this study, a hybrid deep-learning model termed as ODANN, built upon neural networks (NN) coupled with data assimilation and natural language processing (NLP) features extraction methods, has been constructed to concurrently process daily COVID-19 time-series records and large volumes of COVID-19 related Twitter data, as representative of the global community's aggregated emotional responses towards the current pandemic, to model the growth rate in the number of confirmed COVID-19 cases globally via a proposed G parameter. Overall, there were 3 key components to ODANN's development phase, namely: (i) data hydration and pre-processing were performed on COVID-19 related Twitter data ranging between 23 January 2020 and 10 May 2020, which amounted to over 100 million Tweets written in English language; (ii) multiple NLP features extraction methods were subsequently leveraged to encode the hydrated Twitter data into useful semantic word vectors for training ODANN under an optimal set of hyperparameters; and (iii) historical time-series data of defined characteristics were also assimilated into ODANN's selected hidden layer(s) to model the G parameter daily with a lead-time of 1 day. By far, our experimental results demonstrated that by adopting a rolling time-window size of 5 days, with respect to the number of historical time-series records for assimilating different data features, enabled ODANN to outperform other traditional time-series models and recent studies, in terms of the computed RMSE and MAE scores attained from the model's testing step. Overall, the summarized results from ODANN demonstrated its competitive edge in modelling and forecasting the growth rate in the number of COVID-19 cases globally.

1. Introduction

Coronavirus disease 2019 (COVID-19) can easily be spread via coughs or sneeze droplets by entering one's body through their eyes, nose, and mouth [1]. When a person communicates with an infected person within a distance of 1.8 m, the infection likelihood of the former individual augments [2]. The highly infectious nature of the virus has since accelerated its spread globally. On January 30, 2020, the World Health Organization (WHO) declared the COVID-19 outbreak as a Public Health Emergency of International Concern. During March 2020, the WHO officially announced COVID-19 as a global pandemic, hence aggravating healthcare and travel concerns globally. Due to its highly infectious nature, the number of confirmed COVID-19 active and death cases have continued to increase till today, even as vaccination rates gradually ramp up in many countries, hence posing intractable challenges to the global health, environment, and economy [3,4]. Within a short time span, COVID-19 has already been considered to be one of the biggest crises faced by humanity in the 21st century [5].

At present, the global community is closely tracking the evolution of COVID-19 by reporting the daily number of infected persons, recovered persons, and deaths, resulting in available datasets on the internet for research investigations to better understand the unknown virus qualitatively and quantitatively. An important problem statement can thus be formulated, namely: how can we estimate the number of confirmed cases globally, with suitable lead-time and a level of confidence, by analysing historical data pertaining to the growth rate in the number of confirmed COVID-19 cases together with relevant social-environment factors. By modelling and forecasting the spread of COVID-19 over time on a global scale, decision-makers around the world can then undertake appropriate preventative and response

[*] Corresponding author.
E-mail addresses: Alvin.Chew@bentley.com (A.W.Z. Chew), panyue001@sjtu.edu.cn (Y. Pan), YING006@e.ntu.edu.sg (Y. Wang), limao.zhang@ntu.edu.sg (L. Zhang).

[1] These authors contributed to the work equally and should be regarded as co-first authors.

https://doi.org/10.1016/j.knosys.2021.107417
0950-7051/© 2021 Elsevier B.V. All rights reserved.

measures which include, but not limited to, social distancing, self-quarantine, travel restriction, lockdown, etc., to better manage the virus spread over time. To build towards this target objective, statistical and mathematical models can serve as useful predictive tools to model the temporal spread of COVID-19 globally.

Regression analysis, which is a very conventional, but yet useful, technique in the machine learning (ML) domain, is generally useful for time-series predictions in many diverse domains. Hence, it can be considered helpful to decision-makers to forecast the spread of COVID-19 over time on the basis that there is some level of correlation between the virus' transmission rate and a range of controlled and/or uncontrolled factors [6]. For example, the well-known auto-regression integrated moving average (ARIMA) model is expected to estimate the trend and seasonal profiles of the virus spread [7]. On the other hand, the susceptible–infectious-removed (SIR) epidemiological model, which is built upon the compartments of susceptible (S), infected (I), and removed (R) individuals, may also be useful to model the transmission rate of COVID-19 [8]. However, SIR and its variants set the rates of transmission and removal to be constants, hence hindering their abilities to adapt to uncontrollable external conditions/factors [9]. At present, the global transmission rate of COVID-19 exhibits complex patterns which are closely related to a range of social, governmental, and environmental factors. The large multitude of factors, however, can be difficult to be leverage effectively, without any significant data pre-processing, to model and forecast the number of confirmed COVID-19 cases over time. Besides the above-mentioned traditional quantitative methods available, deep learning approaches can also serve as good alternatives by learning from historical data and other forms of big-data information to maximize the prediction model's accuracy in forecasting the transmission rate of COVID-19, which in turn can be useful to support clinical and academic research for COVID-19 in the foreseeable future [10].

One important data source pertaining to COVID-19 can be derived from social media platforms which can be helpful to investigate the society's emotional responses towards the current pandemic [11,12]. For example, Twitter enables netizens to post messages and retweet contents pertaining to all forms of issues occurring globally. The platform also provides users instant access to millions of short messages (public responses, opinions, etc.) towards any specific topic of their interest [13]. In today's context, there are currently more than 300 million monthly active users on Twitter and an average of 500 million tweets are being made daily [14]. An advantage of using Twitter information lies in its real-time deployment and publicly available information, which are tagged by different time zones and geographic locations, for easy access in most parts of the world. Not surprisingly, due to the large volume of data available on Twitter's platform, there have already been research works done in leveraging Twitter data for disease surveillance, which can subsequently provide important insights into public health conditions and return instant feedback to healthcare professionals and the different stakeholders.

ML methods have since been incorporated into Twitter-based healthcare framework to monitor, analyse, and predict the outbreak of different types of diseases in the near real-time context [15]. For example, Signorini et al. [16] examined embedded information in H1N1-related tweets using support vector regression (SVR) to quantify the public sentiments towards the flu virus in the United States and estimate the weekly influenza-like illness level with a reasonably low error percentage of 0.28% on average. Hirose and Wang [17] applied multiple linear regression (MLP) methods with ridge regularization on Twitter data combined with other influenza-like data features derived from the Centres for Disease Control and Prevention (CDC). The model predicted the spread of influenza with a resulting root mean square error (RMSE) of 0.002 on average. Santos and Matos [18] built a Naïve Bayes classifier and MLP models to analyse tweets and web search queries to forecast the incidence rate of influenza in Portugal, where the prediction results have an average correlation ratio of 0.849 with the available ground truth data. It has also been demonstrated that analysing Twitter data offers valuable opportunities to track influenza's infection rate in near real-time and to quantitatively understand the potential upward/downward trends. The predictions can subsequently be useful to generate early warnings to improve both clinical and public health responses. Apart from the conventional flu virus, Twitter data has also been leveraged to forecast the outbreak trends of other infectious diseases such as Zika virus, Malaria, Ebola, and others [19]. The above-mentioned studies have thus indicated the possibility of combining Twitter data with suitable ML models to investigate the infection rate of COVID-19 globally.

In this paper, we propose a novel prediction model, termed as optimized data assimilated neural network (ODANN), which unifies the use of natural language processing (NLP) [20], for generating useful input features layer for neural networks (NN), and data assimilation component for concatenating time-series data into optimal location(s) of the NN's hidden layers to forecast the global growth rate in the total number of confirmed COVID-19 cases with a good level of predictive accuracy. Modelling of the cases' growth rate is achieved via a G parameter which considers the total reported number of confirmed COVID-19 cases on a rolling-forward daily basis. In the model development of ODANN, features extraction methods via NLP are leveraged to extract high-level numerical features from vast volumes of Twitter data (millions in exceedance) associated with COVID-19 since late January 2020. The encoded features from the available Twitter data are then assimilated with historical time-series records for the proposed G parameter by optimally concatenating the features with selected hidden layers of a personalized deep NN model to forecast the G parameter with a defined lead-time of 1 day. At present, our study demonstrates that the resulting time-series predictions for the G parameter from our proposed ODANN model are generally more accurate than the corresponding results derived from other classical time-series prediction models on the same dataset being investigated. The overall novelty of ODANN lies in its end-to-end model framework capable of processing large volumes of Twitter data, as representative of the general community's emotional responses towards COVID-19, to construct useful input features to train deep learning models. The hidden layers of the deep NNs are also carefully optimized, in terms of the selected number of neurons and their placement locations, to assimilate with other important time-series data features to maximize the resulting model's predictive accuracy to forecast the proposed G parameter with a lead-time of 1 day. We are hopeful that ODANN can serve as an alternative prediction model to assist the research community to enhance existing mechanism studies for COVID-19, and subsequently provide useful insights into the nuanced relationship(s) between the global spread of COVID-19 over time and the community's aggregated emotional responses towards the virus.

This paper is structured as follows. Section 2 reviews the previous research works done, using ML and/or deep learning (DL), to analyse Twitter data for modelling the temporal transmission behaviour of COVID-19. In Section 3, we describe our proposed engineering workflow for ODANN in detail, followed by running the constructed prediction model via several computational experiments (model training, validation, and testing) in Section 4 to evaluate the model's predictive accuracy on a defined time-series dataset for the G parameter and COVID-19 related Twitter data. In Section 5, we compare the results derived from ODANN with that of other conventional time-series prediction models, coupled with analysing the models' predictive robustness in handling missing data conditions. In addition, we also compare ODANN's predictive capability with other relevant research studies for the same research objective as outlined in this study. Finally, Section 6 succinctly summarizes the key results derived from our analysis using ODANN by far, and the future works to be involved in the same direction.

2. Literature review

The growing availability of COVID-19 related data provides useful quantitative and qualitative information for researchers to leverage on for modelling the virus' transmission behaviour over time. With the advantages provided by machine learning (e.g., scalability, faster computations, etc.) in the field of healthcare [21], researchers across the world have been developing various ML-based frameworks to manage the unprecedented COVID-19 crisis. For example, Google's DeepMind applied an improved AlphaFold system to model the uncharacterized protein structure related to SARS-CoV-2 and generate accurate 3D models to quantitatively understand complex functions of the underlying proteins. Beck et al. [22] designed a drug-target interaction prediction model via deep learning (DL) to efficiently identify the existing drug candidates that can be re-purposed to disrupt viral proteins in the COVID-19 virus.

On the clinical scale, deep convolutional neural networks (DCNN) are developed to automatically identify COVID-19's infections and regions of interest (ROI) via medical images in an economical and efficient manner [23], which can be especially helpful for the less-developed communities. Pre-trained DL models such as ResNet, U-Net, VB-Net, and others, have also been employed for X-ray or CT image segmentation in COVID-19 applications [23], which have generated promising results in distinguishing COVID-19 from community-acquired pneumonia by segmenting and locating infected lung regions and lesions, and hence tracking and evaluating the virus' severity and evolution over time [24]. On the other hand, as numerical data pertaining to the number of suspected, confirmed, cured, and death COVID-19 cases, passengers travel trajectories, etc., are being shared widely on the internet daily, traditional and novel ML methods can be applied to learn from the vastly available information to forecast the transmission of COVID-19 [25].

For example, researchers have developed multiple prediction models to forecast the virus' future trend behaviour and evaluate the impacts of COVID-19 by estimating key indicators/features associated with the virus which include, but not limited to, its prevalence, mortality rate, recovery rate, etc. For example, Rustam et al. [26] trained four ML regression models, namely the linear regression (LR), least absolute shrinkage and selection operator (LASSO), support vector machine (SVM), and exponential smoothing (ES), to perform reliable forecasting on the global numbers of confirmed, recovered and death cases using a 10-days lead-time. The authors, however, underlined the importance to augment the amount of training data to improve their models' resulting predictions. Yesilkanat [27] adopted the random forest (RF) algorithm to estimate the daily increase in the number of confirmed COVID-19 cases globally with estimated coefficient of determination values ranging between 0.843 and 0.995. Researchers [28–30] have also leveraged on long-short-term-memory (LSTM) neural network, which is capable to model long sequential time-series data, and capture long-term dependencies at the same time, to deliver high-quality prediction results on the number of confirmed COVID-19 cases over time. However, the present daily numbers of COVID-19 cases and deaths may not sufficiently large to maximize the forecasting capability of LSTM model. Going forward, it is thus desirable to apply alternative DL methods to fully leverage on the widely available COVID-19 related data to bridge the gap between intelligent computing and COVID-19 prognosis, which can subsequently be useful to assist the different stakeholders to better manage the current pandemic.

The wealth of publicly available Twitter data can be actively collected via Twitter's streaming API and Tweepy on a daily basis [31,32], which thus offers opportunities for large-scale data mining of the global community's emotional responses towards COVID-19 and subsequently enabling us to quantitatively investigate any relationship between the trending emotions and the proposed G parameter daily with defined lead-times on a global scale. As an example, Lwin et al. [33] examined the public's emotions with over 20 million Twitter posts worldwide during the early outbreak of COVID-19 globally, where the reported findings can be useful to support government measures to maintain the general public's mental wellbeing in the present pandemic situation. Park et al. [34] performed network and context analysis using keywords from COVID-19 associated Twitter data, in Korean language, to track information, discover conversation patterns, and capture the public's interests and perceptions towards COVID-19. Since Twitter data can provide aggregated insights into individuals' emotional responses (or attitudes) towards the global transmission rate of COVID-19, the large data availability can thus be numerically processed to forecast the proposed G parameter by modelling the dimensional effects due to the community's aggregated emotions over time. An important pointer to highlight is that we would expect the raw Twitter data to contain 'noises' such as wrong spellings, punctuations, and differing expressions for underlying common messages. Hence, significant data pre-processing, combined with NLP features extraction methods, is required to minimize the inherent 'noises' and encapsulate the nuanced relationship between the different words available and the respective context, and vice versa. An interesting, but physically intuitive, observation is that the rising number of confirmed COVID-19 cases globally is correlated to Twitter data in the negative emotional realm as the community generally become anxious about the deteriorating social conditions since the virus's inception [35]. The key research question is how we can maximize the value of Twitter data to potentially quantify the complex relationship between the global transmission patterns of COVID-19, i.e., the G parameter, and the relevant emotional responses from the global community.

While large volumes of Twitter data, reflecting the emotional state of the public towards COVID-19, can easily be accessed, the available data requires significant data pre-processing to minimize the inherent noises as discussed. In this aspect, Zheng et al. [36] combined semantic features derived from online news and reports into LSTM neural networks, which could effectively minimize the error in estimating the overall infection rate via an improved susceptible–infected (ISI) model. At the same time, Hazarika and Gupta [37] also recently verified that higher data dimensions, which encompass the social, economic, or/and environmental conditions, derived from the pre-processing step can be useful to improve the prediction model's resulting accuracy in forecasting the transmission rate of COVID-19 globally.

The above-discussed information from the present literature generally underlines the rationality and feasibility of combining large-scale Twitter data and historical time-series records of defined data characteristics for useful COVID-19 related predictions. Hence, this study proposes the ODANN model which consists of an engineering workflow process to systematically extract high-level encoded features from pre-processed COVID-19 related Twitter data, followed by optimally assimilating the model's hidden layers with relevant historical time-series records to model and forecast the proposed G parameter over time. To the best of our knowledge, the formulated workflow of ODANN has not been developed in the literature by far, and we are thus hopeful that the ODANN can serve as an alternative prediction model to build towards reliable predictions in the number of confirmed COVID-19 cases globally on a rolling-forward daily basis using social media big-data with defined lead-times. The forecasting results could potentially assist the different stakeholders to implement appropriate measures to manage COVID-19 on a whole as the virus has been widely expected to become an endemic in the foreseeable future.

3. Methodology

Fig. 1 illustrates the overall workflow for ODANN's developmental phase which involves the following systematic tasks, namely: (1) processing more than 100 million of COVID-19 related Twitter data, written in English language, via suitable NLP features extraction methods for constructing useful numerically encoded features; (2) training of deep NN (DNN) model by leveraging on the encoded features derived from the preceding Task 1, and assimilating with historical time-series records for the proposed G parameter and/or other data features during the model training step; and (3) comparing results derived from a trained ODANN prediction model, with its optimized hyperparameters, with other traditional time-series prediction models to forecast the same G parameter on a daily basis with a defined lead-time. At the same time, a pseudo-code in Algorithm 1 is provided to summarize the key computational steps involved in ODANN's proposed workflow.

3.1. Data hydration and pre-processing

Since the inception of COVID-19 in Wuhan, China, in December 2019, over 100 million of related Tweets, have been made on the internet propagating the keywords of **"covid"**, **"coronavirus"**, **"ncov19"**, **"ncov2019"** and etc. The millions of Tweets, written in the English language, were extracted from an open-source Kaggle source (https://www.kaggle.com/lopezbec/covid19-tweets-dataset) for the modelling analysis in this study. In the selected open-source dataset, the total amount of Tweets written in English constitutes to around 60% of the total available Tweets collated. We analysed the available Tweets for the period between 23 January 2020 and 10 May 2020 for extensiveness, while also accounting for the computational resources (memory storage) limitation in our present study. We note that the amount of COVID-19 Tweets increased exponentially on Twitter's platform from March 2020 and thereafter as illustrated in Fig. 2. The training phase of ODANN was performed using Azure's NC12sV2 N-Series virtual machine (VM) which has 12vCPUs, 224GiB RAM and 2 in-built Tesla P100 GPU cards.

The aggregated pool of Tweets (made in English language) was considered to model the proposed G-parameter over time on a global scale, instead of localizing into specific countries, for the following reasons:

- In the extracted Twitter data, only the geographic information of the re-Tweets can be obtained, however, with minimum information about the Tweets contributors in terms of age, race, etc. The geographic information of the re-Tweets revealed the dominant terms of **"global", "worldwide", "around the world", "everywhere" and "linkedin:"** as summarized in Table 1. Hence, this observation enables us to first investigate any aggregated correlation between the complexity of the Tweets made by the global community and the proposed G-parameter which quantifies the growth rate in the confirmed number of COVID-19 cases on the global scale.

Table 1
Top 30 contributors of COVID-19 related re-Tweets in the extracted dataset between 23 January 2020 and 10 May 2020.

Location name	Quantity
Washington, DC	1 898 553
New York, NY	1 049 229
India	704 448
Global[a]	635 534
London	570 490
Washington, D.C.	539 826
France	462 137
New York	461 072
United States	454 821
New York city	426 335
Madrid	397 464
Brasil	372 637
London, UK	342 182
Paris, France	340 845
London, England	337 515
Los angeles, CA	318 437
USA	255 963
Venezuela	252 693
Paris	246 379
United Kingdom	240 403
New York, USA	235 346
New Delhi, India	234 865
Argentina	230 770
Worldwide[a]	222 662
Linkedin:[a]	221 702
Buenos Aires, Argentina	219 081
Around the world[a]	209 210
New Delhi	198 667
Everywhere[a]	195 892
São Paulo, Brasil	191 823

[a] Keywords which indicate the contributors of COVID-19 re-Tweets on a global aspect.

- The top 10 countries (since 2020) having the highest numbers of reported confirmed COVID-19 cases included USA, India, United Kingdom, Brazil, France, etc., were also among the list of top contributors to the re-Tweets as shown in Table 1. This thus underlines our approach to first aggregate all available Tweets, without any prior filtering with respect to any countries, to correlate with the G- parameter over time on the global scale.

As discussed earlier, the large amount of Tweet data available serves as useful data to analyse and quantify the aggregated emotional responses of the general community towards COVID-19 on the global scale. The proposed G parameter, which relates to the global growth rate in the total number of confirmed COVID-19 cases on a rolling-forward daily basis, can be expressed as follows:

$$G_t = \frac{Y_t - Y_{t-1}}{Y_{t-1}} \times 100\% \tag{1}$$

where Y_t represents the global number of confirmed COVID-19 cases at time (t), and Y_{t-1} represents the global number of COVID-19 cases at time $(t - 1)$ from the previous day. For example, the G value on 31 January 2020 (t) is computed by dividing the difference between the respective number of COVID-19 confirmed cases on 31 January (t) and 30 January 2020 $(t - 1)$ by the number of COVID-19 confirmed cases from the latter's date. The number of confirmed COVID-19 cases recorded on the global context, since 31 December 2019, were also extracted from another open-source database (https://ourworldindata.org/coronavirus-data). Our analysis instead models the daily growth rate, on a rolling-forward basis, in the proposed G parameter for the period between 25 January 2020 and 11 May 2020 (see Fig. 3). Note that we started on 25 January 2020 to forecast the G value in order to maintain a lead-time of 1 day in leveraging

Algorithm 1. Computational workflow for training ODANN prediction model

```
1. Input: COVID-19 Twitter text data ($X$), time-series records for proposed G-parameter ($N$) and/or
   other data features, size of rolling time-window ($T_{roll}$), size of lead-time ($T_{lead}$), size of input layer
   ($D$), values of model's hyperparameters ($H_1, H_2, \ldots\ldots$), tolerance values for error metrics
   ($RMSE_{tol}, MPAE_{tol}$, etc.), validation dataset, testing dataset
2. Output: G-parameter predictions with defined lead-time ($Y$) using proposed approach, G-parameter
   predictions with defined lead-time ($Y'$) using other time-series models
3. do
      a. initialize features extractions of $X$ dataset to develop input layer of user-defined $D$ value
      b. initialize training of ODANN model with user-defined $H_1, H_2, \ldots\ldots$ values & assimilate with
         $T_{roll}$ of historical time-series records for G parameter and/or other data features with defined
         $T_{lead}$; $T_{roll} \ll N$
      c. derive $Y$ predictions on validation and testing datasets via trained ODANN model with defined
         $T_{lead}$
      d. model evaluation of predictions from ODANN on validation datasets via $RMSE$ & $MPAE$
         scores
         if $RMSE < RMSE_{tol}$ and $MPAE < MPAE_{tol}$
               proceed
                      e. compare $Y$ predictions with $Y'$ from other time-series models on testing
                         datasets via $RMSE$ & $MPAE$ scores
               while other time-series models perform better via lower $RMSE$ & $MPAE$ scores
                      repeat computational steps (a-e)
         else
              re-train ODANN model with more sets of hyperparameters
4. Store best trained ODANN model in database for near real-time predictions
5. end
```

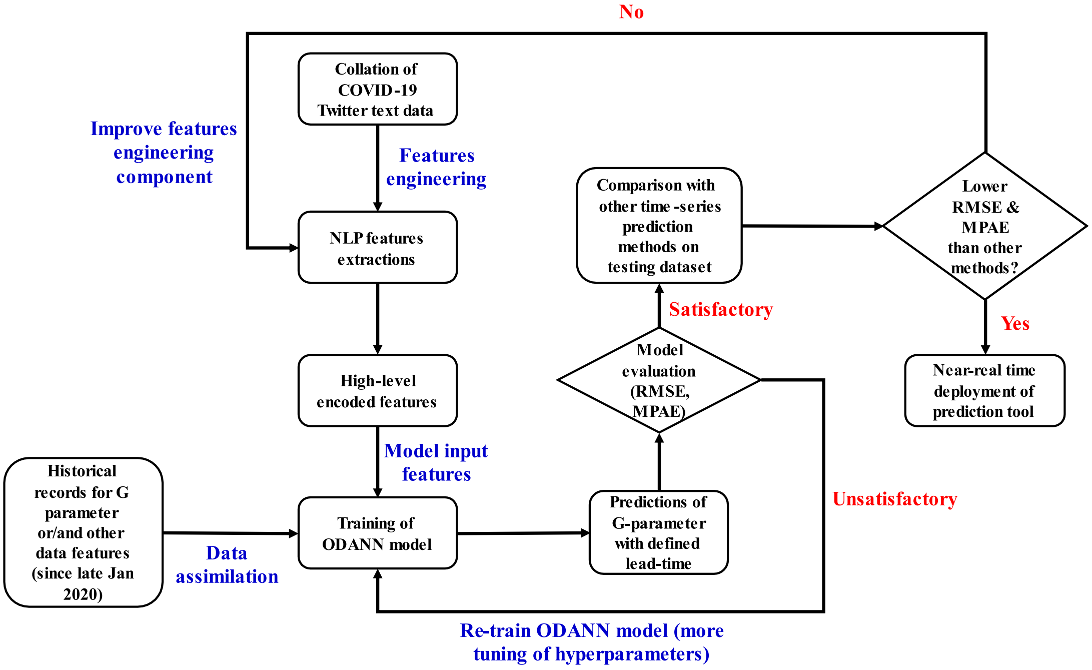

No
Collation of COVID-19 Twitter text data
Improve features engineering component
Features engineering
NLP features extractions
Comparison with other time -series prediction methods on testing dataset
Lower RMSE & MPAE than other methods?
Satisfactory
Yes
Model evaluation (RMSE, MPAE)
Near-real time deployment of prediction tool
High-level encoded features
Model input features
Historical records for G parameter or/and other data features (since late Jan 2020)
Data assimilation
Training of ODANN model
Predictions of G-parameter with defined lead-time
Unsatisfactory
Re-train ODANN model (more tuning of hyperparameters)

Fig. 1. ODANN's workflow to integrate Twitter data and other historical time-series records to model and forecast G parameter over time, coupled with comparison of model's resulting accuracy with alternative time-series prediction methods.

the pre-processed Twitter data from the previous day, i.e. 23 January 2020, for the modelling step. We also note that we stored the full historical time-records for the G parameter since 18 January 2020 for the analysis via a rolling time-window size for data assimilation component in ODANN which will be further discussed later.

Hydration of the known Tweet IDs involves extracting the important text information, which contributes to the features extraction component in this study. Data hydration is subsequently performed using an open-source command-line Tool, *Twarc* which is also programmed in Python language. Each Tweet data is represented as a JSON object that can be extracted using Twitter API by requiring the user to provide 4 key parameters,

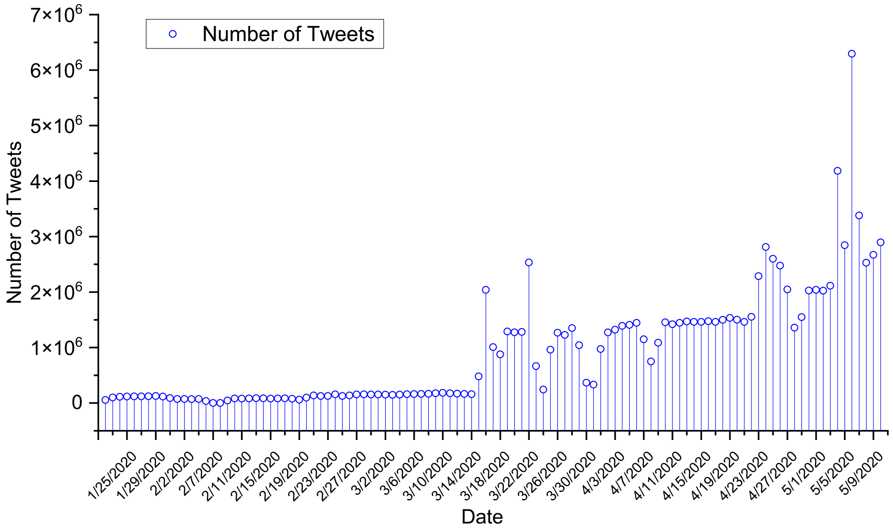


Fig. 2. Number of the collected Tweets between 23 January 2020 and 10 May 2020 in this study.

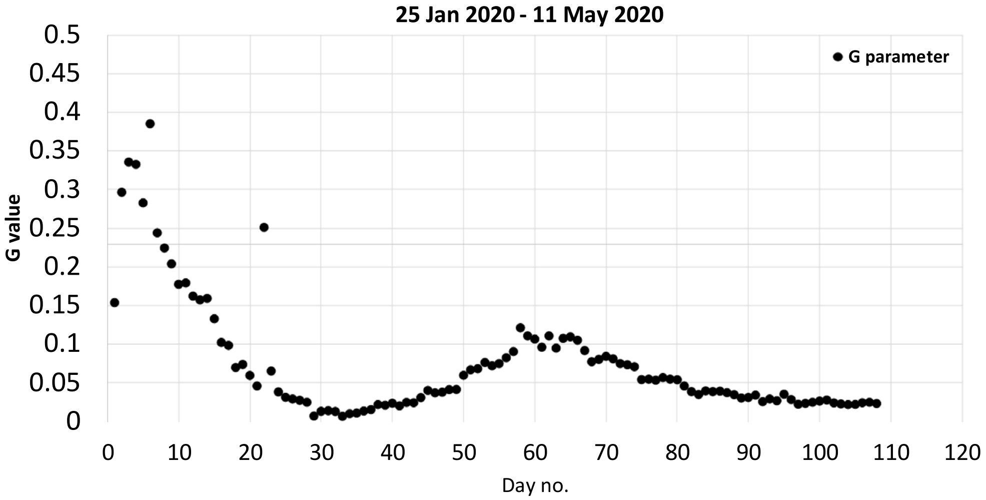


Fig. 3. Plot of temporal variation of G values between 25 January 2020 (Day 1) and 11 May (Day 108) 2020.

namely: (a) *consumer_key*; (b) *consumer_secret*; (c) *access_token*; (d) *access_token_secret*, in our personalized data hydration algorithm for Twitter data. The in-built *Twarc* library can automatically handle Twitter API's ratelimits where the present quantity limit is 45,000 tweets for every 15 min. In the present Twitter dataset extracted for the analysis, there were approximately 1 million Tweet IDs for every date since 23 January 2020 and the average hydration time takes around 6 h with our present computational resources available. Hence, the total time required to hydrate the total available Tweet IDs, ranging between 23 January 2020 and 10 May 2020, took around 45 days from the start of this study.

After hydration, the extracted text data in English language undergoes a series of pre-processing steps to remove their inherent "noises" prior to constructing their respective vectorized semantic word representations. The constructed vectors from each processed Tweet data will subsequently be analysed via ODANN's deep learning algorithms to model the proposed G parameter from Eq. (1). Systematically, the following data pre-processing steps are performed for each of the extracted Tweet data (with a unique Tweet ID), namely:

i. Tokenize each Tweet sentence, as corresponding to each unique Tweet ID, into individual unique words/vocabularies
ii. Examine every individual word against a known pool of stop words as derived from the open-source Natural Language Toolkit (NLTK) library which is also programmed in Python language. If any individual word is identified as a stop word, remove them accordingly, else retain them for further analysis
iii. Remove all punctuations
iv. Remove all other unknown symbols
v. Combine all remaining words to form a new sentence for the specific Tweet ID

3.2. Feature extraction methods via natural language processing

As highlighted earlier, each date, starting from 23 January 2020, had an average of 1 million unique Tweets (including re-Tweets) text data which contributed to their respective corpus (i.e., collection of words/sentences) for the assigned date. The combined corpus of processed Tweet data derived, as ranging between 23 January 2020 and 10 May 2020, will then be further encoded via suitable NLP feature extraction methods to build their corresponding semantic word representations of defined vector sizes. The NLP features extraction algorithms adopted in this study are namely: (a) TfidfVectorizer; (b) Word2Vec – Continuous Bag of Words (CBOW); (c) Word2Vec – Skipgram. In the following, the significance and possible shortcomings for each feature extraction method are briefly described.

3.2.1. *TfidfVectorizer*

TfidfVectorizer stands for *"Term Frequency – Inverse Document Frequency"* which represents the components of the respective resulting scores assigned to each unique word in the original pool of corpus [38]. The inherent *Term Frequency* function computes the frequency of each unique word that exists in the corpus, while the *Inverse Document Frequency* component downscales the respective weights of common words within the same corpus. The latter can be useful to handle imbalanced dataset where the frequency of certain words can far exceed that of others. In general, the TfidfVectorizer algorithm first tokenizes the corpus into individual words, followed by learning the pool of tokenized vocabularies and then inverting the frequency weights of the respective words for further analytics.

The current combined corpus of processed Tweet texts, as ranging between 23 January 2020 and 10 May 2020, contains 94,237 unique words of which each word has its own one-hot encoded representation. However, this method consists of several shortcomings which include: (a) inability to effectively learn and quantify the semantic relationship among the connected words within the corpus or across the different texts for semantic analysis purpose; and (b) the need to re-train the prediction model if new words are introduced into the original corpus, hence constricting the model's scalability.

3.2.2. *Word2Vec (CBOW and Skipgram)*

Word2Vec is a powerful and efficient unsupervised machine learning algorithm, as developed by Google in 2013, which can create neural word embeddings for large corpus. Its inherent algorithm contains an in-built two-layer neural network (shallow–deep network) which processes the input corpus data by forming the word representation for the different words. The output from the neural network model is a set of feature vectors which represent the neighbouring words in the same corpus. It is worth noting that Word2Vec can be classified as a shallow–deep neural network model and works best with big text data where the model can learn the complex relationship among the different words inside the corpus. The derived feature vectors can be processed further by using them as input features to train other machine learning algorithms for predictive analytics or simply queried for semantic analysis purposes. In general, Word2Vec behaves in a similar format as autoencoders by first encoding each word in the concerned corpus and then train the model by mapping the vectorized representation of each word to that of the other surrounding words in the context of the same corpus. This whole process can generally be performed via CBOW or Skipgram algorithms which were also previously developed by Google's research team [39,40].

In the CBOW model, the distributed representations of the context (i.e., surrounding words), built upon a defined window size, are combined accordingly to predict the word in the middle. On the contrary, the Skipgram model maps the distributed representation of the input word by the user to predict its relevant context. CBOW can generally train much faster than Skipgram for the same corpus as the shallow–deep neural network is learning to map the context of the corpus to each available unique word, hence the fitting process is expected to be quicker with more features available in corpus' context as the input layer. In addition, CBOW is expected to derive more effective representations for words that occur more frequently. Skipgram, however, requires a longer training time with large text data, but can better represents less frequently occurring or rare words/sentences in the corpus. The hyper-parameters adopted for both CBOW and Skipgram in this study for analysing the corpus of processed Tweet text are summarized in Table 2 for model reproducibility. We note that the selected values for the min-count and window-size hyper-parameters for CBOW and Skipgram algorithms are based upon recommendations from the literature [39,40]. Hence, the focus is to evaluate the varying orders of magnitudes in the vector-size for the word representation, i.e., embedding layer, for the combined corpus of extracted Tweets in the English language as part of the features extraction analysis.

Table 2
Summary of hyperparameters' values for Word2Vec algorithms (CBOW, Skipgram) adopted in this study.

Hyper-parameters	CBOW	Skipgram
min-count	3	3
window-size	5	7
vector-size	100, 500, 5000	100, 500, 5000
no_of_workers	5	5

3.3. *Development of ODANN prediction model*

The encoded word representation, as derived from the earlier discussed NLP features extraction methods (TfidfVectorizer, CBOW, Skipgram,), are then leveraged as the high-level input features layer for modelling the G parameter over time (see Figs. 4 and 5) via deep learning methods. This study adopted a personalized deep neural network (DNN) model with and without data assimilation (ODANN) component based upon a rolling time-window size for modelling and forecasting the G-parameter over time since 25 January 2020 on the global scale.

3.3.1. *DNN without data assimilation*

The input features layers, as constructed from the Word2Vec algorithms, adopted a pre-defined vector size for training a simple DNN model (see Fig. 4), which has multiple hidden layers of neurons inclusive of the final output layer of 1 neuron for modelling the proposed G parameter on a daily basis (see details in Table 3). Again, we note that the selected vector size for the input layer, ranging among 100, 500, and 5000, does not represent the number of unique words/vocabularies in the analysed corpus as discussed earlier. A series of trial and error was thus performed to determine the optimal size of model's input layer which can achieve the highest possible prediction accuracy between the predicted and measured G values over time from the subsequent testing step. Generally, a smaller vector size for the input features layer aggregates the quantitative contextual relationship among the different words, and vice versa, which usefulness depends on the specific application. This remains to be further explored for modelling the temporal G values on a daily basis via DNN. The values for the batch size were also varied accordingly for tuning DNN during its training phase as summarized in Table 3.

For TfidfVectorizer, however, the vector size for the DNN's input features layer is usually static and equates to the number of unique words/vocabularies which amounts to 94,237 from the present pool of combined corpus as extracted between 25 January 2020 and 11 May 2020. The contextual relationship among the different words is thus not encapsulated in the constructed input features layer using TfidfVectorizer since the size of the input layer is directly built upon the number of unique words in the corpus. The batch size hyperparameter was also tuned when training DNN, as built upon the input layer constructed using TfidfVectorizer features extraction method (see Table 3). Finally, we again note that we maintained a minimum lead-time of 1 day for the modelling step in all scenarios (different input layer sizes, etc.). As another example, to forecast the G value for 31 March 2020 via a trained DNN prediction model, we leveraged on the available Twitter data from 29 March 2020, hence maintaining the 1-day lead-time.

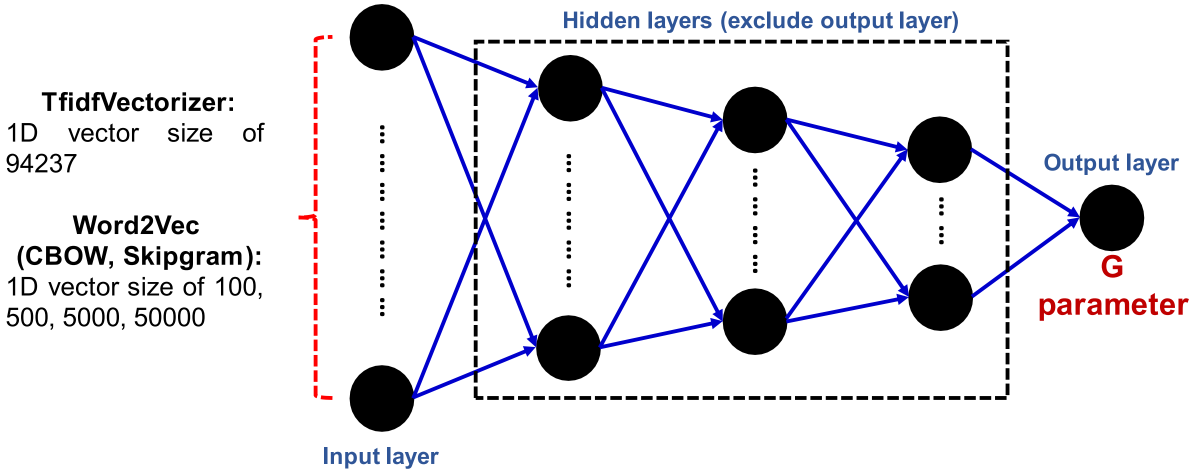

TfidfVectorizer:
1D vector size of 94237
Word2Vec (CBOW, Skipgram):
1D vector size of 100, 500, 5000, 50000
Hidden layers (exclude output layer)
Output layer
G parameter
Input layer

Fig. 4. Illustration of DNN model without data assimilation component to model proposed G parameter.

3.3.2. ODANN

Data assimilation into ODANN's hidden layers, as illustrated in Fig. 5, is achieved by leveraging on a rolling historical time-window size, inclusive of the minimum lead-time of 1 day, which merges the selected hidden layer with additional neurons as representative of the actual G values from the previous days, based upon the defined time-window. For example, with a 3-day rolling time window ($G_{t-2}, G_{t-3}, G_{t-4}$), where the lead-time of 1 day is inherent, 3 additional neurons representing the respective values for $G_{t-2}, G_{t-3}, G_{t-4}$ will then be concatenated with the selected hidden layer from Fig. 3 to forecast the G value on the current day itself (i.e., G_t) via ODANN. In the case of forecasting the G value on 25 January 2020, we leveraged on the historical time records for the G parameter from 21–23 January 2020 for the data assimilation step. The novelty of ODANN (Fig. 5) lies in its capability to concatenate the encoded input features from large volumes of COVID-19 related Twitter data and the historical time-series records for the G parameter into a single end-to-end model framework for near real-time predictions. Additional discussions on the novelty aspect of ODANN to model the G-parameter in the near real-time context will be provided in the later section.

In this study, we explored the rolling historical time-window sizes of 3, 5, and 7 days for training the ODANN model with the proposed data assimilation component. As part of the grid-search process to optimize the initialized weightage values for the hidden layers of ODANN, we define an optimization criteria as: (i) estimated mean squared error (MSE) value must be lower than 0.0100 (RMSE $\approx$ 0.1); and (ii) estimated mean squared error (MAE) value must be lower than 0.100, during ODANN's validation step for the training scenarios with and without data assimilation component, i.e. simple DNN model from Fig. 4.

3.4. Prediction performance evaluation

As discussed previously, to evaluate ODANN's predictive capability (Figs. 4 and 5) during its training (with validation) and testing steps, the following error metrics were adopted, namely: (i) mean squared error (MSE) in Eq. (2); (ii) root mean squared error (RMSE) in Eq. (3); (iii) mean absolute error (MAE) in Eq. (4). We note that MSE was selected as the key cost function for the model training step (see Table 2) to minimize the error difference between the measured and simulated G values, while RMSE and MAE were also computed at the same time for a comprehensive analysis.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(G_{p,i} - G_{m,i}\right)^2 \tag{2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(G_{p,i} - G_{m,i}\right)^2} \tag{3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|G_{p,i} - G_{m,i}\right| \tag{4}$$

where N is the number of data quantity being analysed, $G_{p,i}$ the predicted G value on a specific day ($t = i$), and $G_{m,i}$ the measured G value on the same day itself ($t = i$).

4. Computational experiments and results

4.1. ODANN's model training configuration

The extracted and processed dataset, ranging between 25 January 2020 and 11 May 2020, had a total data quantity of 108 data instances on a daily basis. Training of ODANN is performed with 85% of the total quantity, while the remaining quantity is used for model testing. We note that within the 85% of data quantity for model training, 20% is used for validating ODANN during its training phase. In addition, we note that no random shuffling of the original dataset is performed prior to splitting it into components for model training, validation, and testing steps, for the following reasons:

- Forecasting of the proposed G parameter for COVID-19 should be based upon extrapolation computations using the optimized weights of ODANN's hidden layers where they have been trained, a priori, using continuous dataset for the G parameter since the pandemic's inception, and not segmented discrete data points at separated timestamps. We believe that modelling the continuous evolution of the G parameter on a daily basis, with respect to the available Twitter data made available in near real-time, can more accurately forecast the growth rate in the confirmed number of COVID-19 cases on a global scale.
- Building upon the near real-time requirement in ODANN's predictive capability, the forecasting of the G parameter, as discussed in-detail previously, should be based upon Twitter data being collated on a daily basis with a lead-time of 1 day and a defined number of historical records for the same G metric with respect to the modeller's rolling time-window size. Hence, the forecasting step by ODANN in the near real-time context on any given day requires continuous influx of big-data information, and not randomized sets of historical datasets.

Table 3
Summary of hyperparameter values for training ODANN with and without data assimilation component.

Hyper-parameters	ODANN	DNN without data assimilation
No. of neurons in hidden layer 1	6	
No. of neurons in hidden layer 2	1	
No. of neurons in hidden layer 3	Size of rolling time-window(e.g., number of neurons = 3 for 3 days' time-window)	Nil
No. of neurons in hidden layer 4	Size of rolling time-window(e.g., number of neurons = 3 for 3 days' time-window)	Nil
No. of neurons in output layer	1	
Rolling time-window size	3, 5 & 7 days	
Batch Size	2, 4, 6, 8, 12, 16	
Number of Epochs	50	
Learning rate	0.0001	
Activation function for all hidden layers	Exponential Linear Unit (ELU)	
Optimization function	Adam	
Key cost function	Mean Squared Error (MSE) – set criteria to be below value of 0.0100 for model validation step	

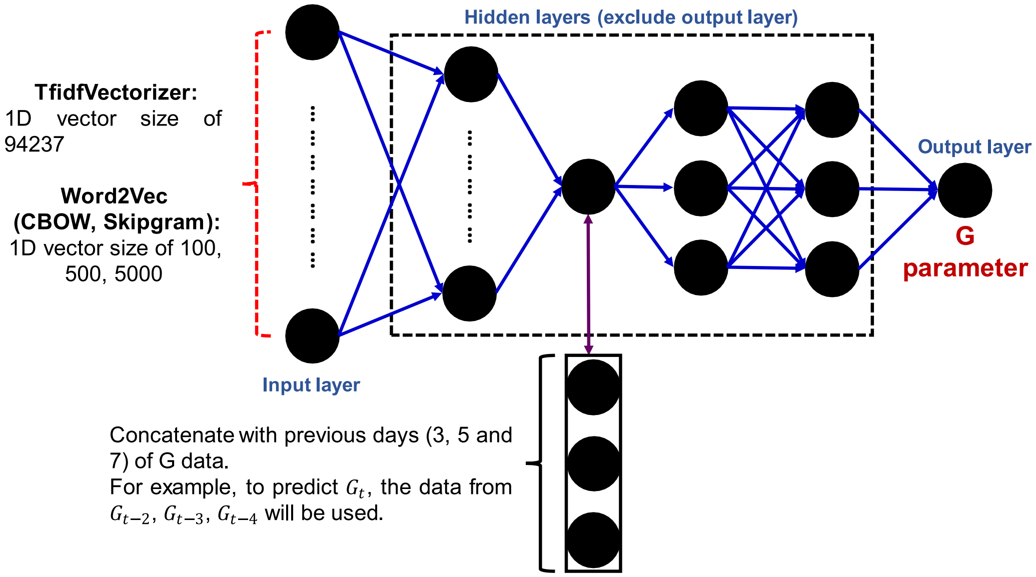

Hidden layers (exclude output layer)
TfidfVectorizer:
1D vector size of 94237
Word2Vec (CBOW, Skipgram):
1D vector size of 100, 500, 5000
Output layer
G parameter
Input layer
Concatenate with previous days (3, 5 and 7) of G data.
For example, to predict G_t, the data from G_{t-2}, G_{t-3}, G_{t-4} will be used.

Fig. 5. Illustration of ODANN model, with data assimilation component, to model proposed G parameter with rolling time-window size.

In summary, the computational protocol for training and validating ODANN (coupled with Algorithm 1) to forecast, with and without data assimilation, the global G parameter on any given day, i.e., G_t, with a lead-time of 1 day is summarized as follows:

- **DANN without data assimilation**: The proposed workflow first constructs the required vectorized word representation for the collated Twitter data from $(t-2)$ day using either: (i) TfidfVectorizer feature extraction method which resulting vector size will be 94 237; or (ii) CBOW or Skipgram feature extraction method which resulting vector size can be 100, 500 or 5000. Then, the vectorized word representations are fed as the input features layer for the simple DNN model, as depicted in Fig. 4, to perform the model training and validation steps using the listed hyper-parameters from Table 3.
- **ODANN with data assimilation**: Perform identical input features layer construction from that of ODANN without data assimilation. Next, previous days of G values, as represented in one-dimensional (1D) arrays, for the required data assimilation component are prepared based on the pre-defined rolling time-window size. For example, a rolling time-window size of 3 days will generate a 1D array size of 3 which compacts the G values from $(t-2)$, $(t-3)$, and $(t-4)$. Then, the vectorized data are assimilated into ODANN's hidden layer from Fig. 5 for the same model training and validation steps. We note that selection of the specific hidden layer for the data assimilation step was determined after a series of trial and error, and additional discussions will be provided in the subsequent section. In the present neural network design for simple DNN and ODANN from Table 3 and Figs. 4 and 5 respectively, each model training and validation run, pertaining to every combination of the defined hyperparameters, required an average of 1 min for completing their computations.

4.2. Analysis of prediction results for ODANN

This section presents a comprehensive comparison of the results derived from ODANN with and without data assimilation, i.e. simple DNN. To ensure clarity to our readers, the following details will be provided. Benchmarking was first performed for DNN without data assimilation using the different types of the NLP features extraction methods, as discussed earlier. The optimized model configuration for the vector size of the input features layer and the rolling time-window size, which can best minimize the computed error metrics in Eqs. (2)–(4) from the testing step of DNN without data assimilation, were subsequently adopted for training and validating ODANN with data assimilation component. The observed differences in the model's resulting predictive capability from the testing step using different model configurations were also discussed. Finally, the best model configuration for ODANN, with data assimilation component, was then justified for its near real-time predictions of the proposed G parameter.

4.2.1. Benchmark results from DNN without data assimilation

Without any data assimilation, Figs. 6 to 8 illustrate the best comparison between the measured and predicted G values obtained from the simple DNN design for its combined training, validation and testing steps, coupled with different types of NLP features extraction methods built upon the respective optimal batch size (from the grid-search step) adopted in this study. Readers are also referred to Table 4 for the summary of the lowest MSE, RMSE, and MAE values computed from DNN's combined training, validation, and testing steps using the optimal batch sizes coupled with the respective NLP features extraction methods. Generally, the benchmark results indicate the following key findings.

- The vector size of the input features layer for the deep learning model must be fine-tuned to best improve its level of agreement, i.e., goodness-of-fit, between the predicted and measured G values from the model's testing step (see Table 4). For example, Figs. 7 and 8 show that increasing the vector size from 100 to 5000, using CBOW or Skipgram feature extraction method, resulted in better fitting between the measured and simulated G values from the model's testing step. On the contrary, Fig. 6 shows that a vector size of almost 20 times bigger than that of Figs. 7c and 8c, resulting from the TfidfVectorizer features extraction method, will instead increase the overall error deviation between the measured and predicted G values from the same testing step. Therefore, the results indicate the need to determine an optimal vector size for the model's input features layer, without data assimilation component, as depending on the type of features extraction method being adopted a priori.
- Using Skip-gram as the features extraction method to build the input features layer for the simple DNN model is likely to overfit the measured G data as shown in Fig. 6, where a low error score was achieved with the training dataset, however, instead resulting in a high error score on the testing dataset subsequently. Hence, the model was likely to have overfitted the encoded input features from Skip-gram on the training data for the proposed G parameter. For comparison, it can be seen from Figs. 7 and 8 that CBOW and Skip-gram, as belonging to Word2Vec algorithms, can most likely mitigate the overfitting issue in the model's combined training, validation, and testing steps.
- Using Twitter data solely is unlikely to fully encapsulate the complex dynamics inherent to the global growth rate of the G parameter, hence indicating that the emotional responses of the global community towards COVID-19 are coupled with other unknown conditions controlling the spread of COVID-19 globally. Further discussions will be provided in the later section which can be useful to further underscore the usefulness (and novelty) of the proposed ODANN model with data assimilation component.

4.2.2. Improved results from ODANN

Building upon the benchmark models, as described in the preceding sub-section, data assimilation with the proposed ODANN design from Fig. 5 is then leveraged for the same modelling step. Likewise, Figs. 9 to 15 illustrate the best comparison between the measured and predicted G values using the different types of NLP feature extraction methods with their respective optimal batch size for the data assimilation component. Readers are also referred to Table 5 for the summary of the lowest possible MSE, RMSE, and MAE values computed from the model's combined model training, validation, and testing steps using the optimal batch sizes which corresponds to each of the NLP features extraction method used, coupled with data assimilation. As compared to the previous benchmark results, the following key findings can be summarized:

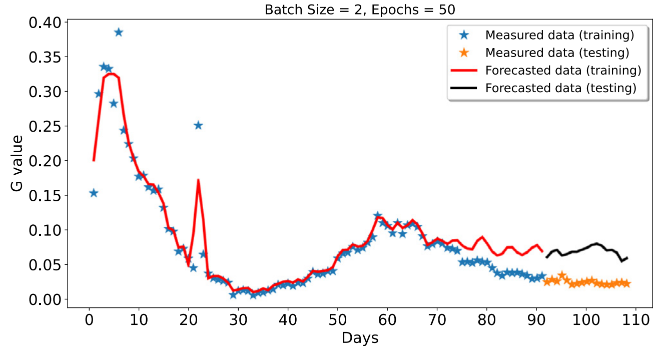


Fig. 6. Comparison between predicted and measured G values, using simple DNN model, for 25 Jan 2020 to 11 May 2020 using TfidfVectorizer pre-processing model with fixed vector size of 94,237.

- For all 3 NLP feature extraction methods used, the proposed data assimilation component, as opposed to no data assimilation, significantly improves the level of agreement between the predicted and measured G values for the validation and testing steps of ODANN under the different combinations of batch size and rolling time-window size. For example, comparing Figs. 6 and 9a under the same batch size value of 2, there was a clear improvement in the level of agreement between the predicted and measured G values from the model's testing step where the respective RMSE value reduced from 0.0448 to 0.00603 by assimilating a 3 days' rolling time-window for the historical G values.
- The current best agreement between the predicted and measured G values for the testing step of ODANN was achieved by using the CBOW feature extraction method together with the batch size of 2 and a rolling time-window size of 5 days (see Fig. 11a) with respect to the historical G values for the ODANN's training phase. The best results indicate that CBOW is generally more effective to quantitatively encapsulate the inherent context of the Tweets (and re-Tweets) made by netizens towards COVID-19.
- Building upon the preceding pointer, the optimal rolling time-window size of 5 days is likely to indicate a relatively fast transformation of the pandemic's behaviour over time. The optimal batch sizes, as ranging between 2 and 6, for CBOW and Skip-gram (Figs. 11a, 12a, and 13a) respectively supports the observation that feeding smaller groups of input features into the ODANN with data assimilation component generally improved the model's predictive accuracy during its testing phase.
- Overall, a smaller vector size for ODANN's input features layer, as derived from either CBOW or Skip-gram features extraction methods, can better capture the complex emotional responses of the general population towards COVID-19 and thus transforming them into more useful high-level input features to maximize the model's predictive accuracy, i.e., lowering the MSE, RMSE, and MAE scores from the testing phase of ODANN (see Table 5), as compared to using larger vector size which can generally incur higher computational time during the model's training and validation steps, especially if a larger model architecture is employed for ODANN in future studies.
- If insufficient historical data records for the proposed G parameter ($\leq$ 3 days) are present in a hypothetical scenario, the current results obtained indicate that using Skip-gram, as the features extraction method, may be more effective in generating more useful high-level input features (see Fig. 13a) for ODANN as compared to that of CBOW. The latter

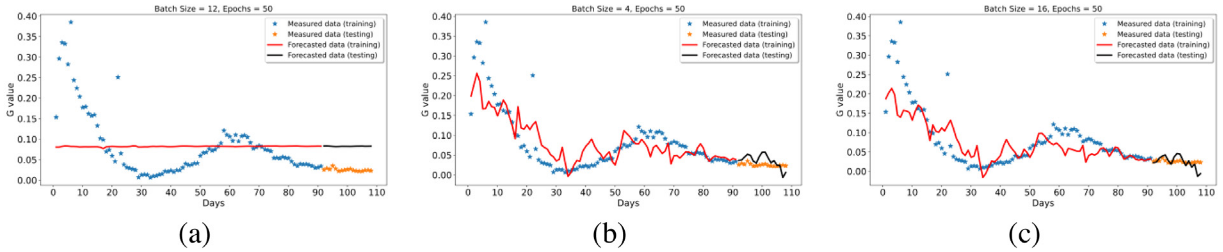


Fig. 7. Comparison between predicted and measured G values, using simple DNN model, for 25 Jan 2020 to 11 May 2020 using CBOW NLP features extraction method with varying vector size for input layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.

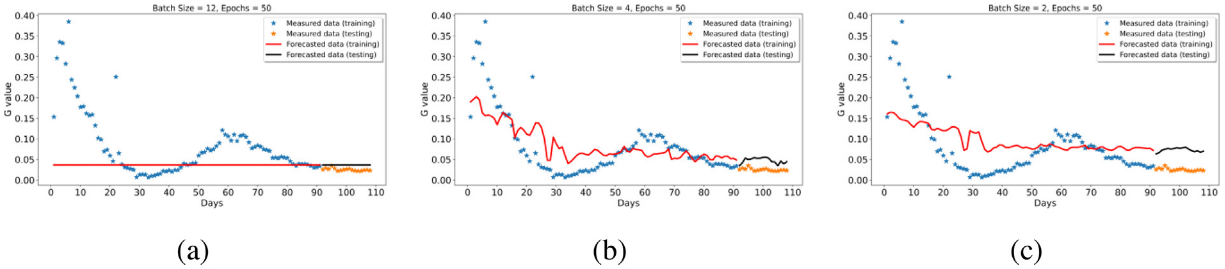


Fig. 8. Comparison between predicted and measured G values, using simple DNN model, for 25 Jan 2020 to 11 May 2020 using Skip-gram NLP features extraction method with varying vector size for input layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.

Table 4
Summary of lowest possible MSE, RMSE, and MAE values derived for DNN model's validation and testing steps, without data assimilation component, using different NLP features extraction methods.

NLP features extraction method	Batch size	Vector size	Error scores on validation dataset			Error scores on testing dataset		
			MSE	RMSE	MAE	MSE	RMSE	MAE
TfidfVectorizer	2	94 237	0.000446	0.0211	0.0133	0.00201	0.0448	0.0442
CBOW	12	100	0.00854	0.0924	0.0562	0.000148	0.0122	0.0117
CBOW	4	500	0.00212	0.0460	0.0332	0.000337	0.0184	0.0160
CBOW	16	5000	0.00256	0.0506	0.0337	0.000221	0.0149	0.0116
Skip-gram	12	100	0.00858	0.0926	0.0564	0.000137	0.0117	0.0112
Skip-gram	4	500	0.00292	0.0540	0.0390	0.000538	0.0232	0.0222
Skip-gram	2	5000	0.00121	0.0347	0.0260	0.000231	0.0152	0.0122

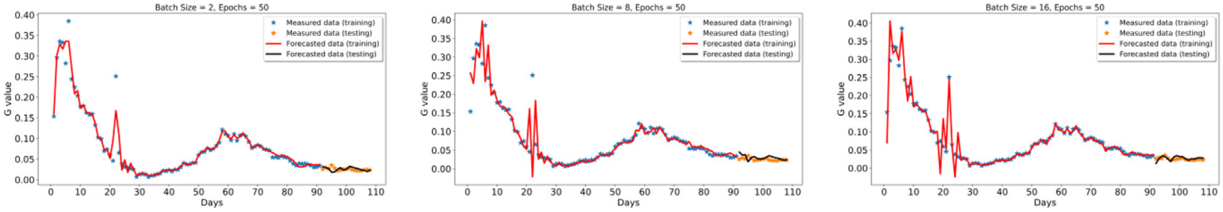
method (CBOW) appears to better complement ODANN's predictive accuracy with additional historical data records, i.e., greater than 3 days, as shown in Figs. 11a and 12a.

At this stage, we have shown that ODANN, coupled with data assimilation using a historical time-window size of 5 days, which input features layers (vector size = 500) built upon CBOW features extraction method can best maximize model's predictive accuracy, in terms of the resulting error scores, as summarized in Table 5. For extensiveness in our proposed analysis, we investigated varying sizes of the training and validation datasets as part of cross-validating ODANN's predictive capability from its training step. Traditionally, cross-validation for ML/DL models requires randomizing the datasets for splitting into training, validation, and testing subsets. However, in our present time-series modelling for the proposed G parameter, we avoid shuffling the available data pool ranging between 25 Jan 2020 and 11 May 2020 for the same reasons as outlined earlier. Instead, our cross-validation analysis solely focuses on investigating the effects of varying the sizes of the training (and validation) and testing datasets into multiple combinations of: (i) 60% for training and validation, 40% for testing; (ii) 70% for training and validation, 30% for testing; (iii) 80% for training and validation, 20% for testing; and (iv) 85% for training and validation, 15% for testing, as listed in Table 6 while retaining the 5-days data assimilation component, coupled with CBOW features extraction method, and batch-size of 2 for the ODANN model. In summary, varying the sizes of datasets assigned for training (and validating) and testing ODANN indicated the following:

- As expected, reduction of the data availability for training and validating ODANN increases the resulting error scores from the model's testing step as shown in Table 6. The likely reason is ascribed to the spike in the G value at around Day 60 (mid-March 2020) which causes the training model to over-predict the remaining period after Day 60, especially with smaller pools of training data as the model may learn that G value is increasing continuously.
- On the contrary, the selection of 85% of the total available data quantity for training and validating ODANN is expected to optimize the model's predictive capability (Tables 5 and 6) as the training data pool consists of 2 independent periods which demonstrated continuous rate of increase and decline in the G values as illustrated in Fig. 3. Specifically, the rate of increase in the G values occurred in the approximate periods of Day 0 to Day 5 and Day 40 to Day 60, while the rate of decline occurred approximately during Day 10 to Day 30 and Day 60 to Day 85. Hence, the data distribution for the 2 contrasting dynamic behaviour in the proposed G parameter over time is balanced to a significant extent which benefits the model's learning capability, especially when coupled with the data assimilation component.

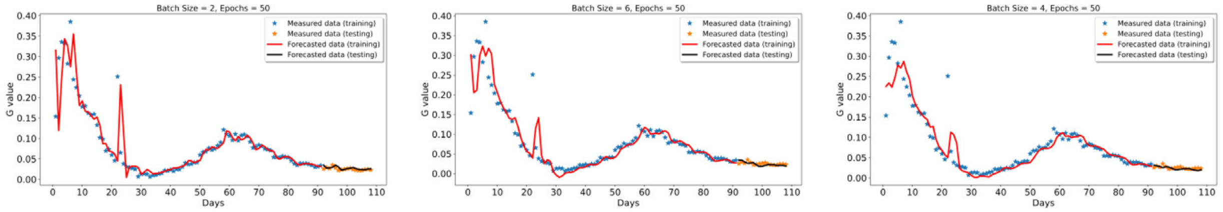
4.3. Time-efficiency of proposed method

On any given day (e.g. 1 May 2020) in the near real-time context, with the defined lead-time of 1 day, our proposed method



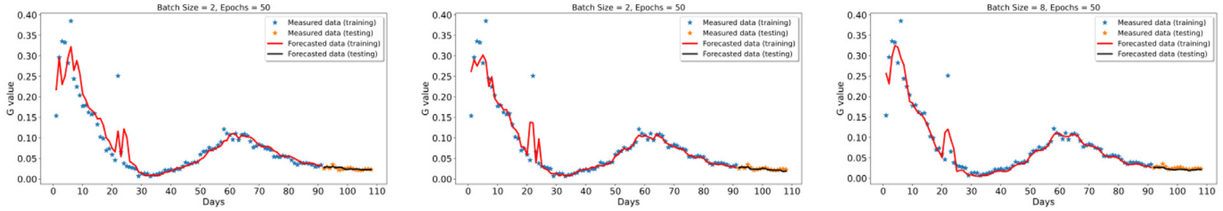
(a) (b) (c)

Fig. 9. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using TfidfVectorizer features extraction method with fixed vector size of 94 237 for ODANN, coupled varying rolling time-window sizes: (a) time-window = 3 days; (b) time-window = 5 days; (c) time-window = 7 days.



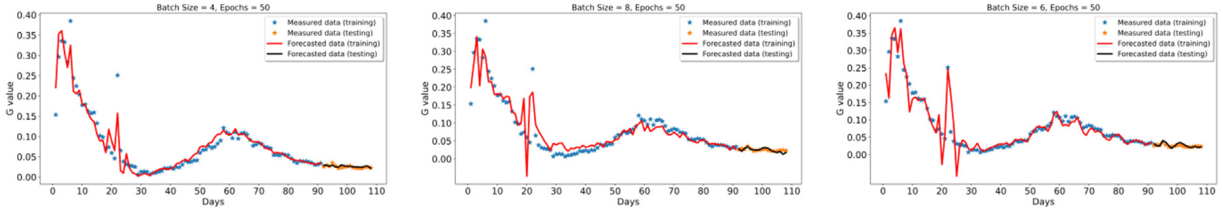
(a) (b) (c)

Fig. 10. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using CBOW features extraction method for ODANN with 3 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.



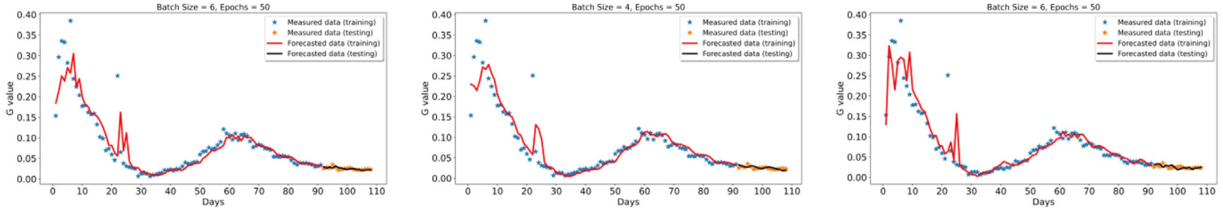
(a) (b) (c)

Fig. 11. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using CBOW features extraction method for ODANN with 5 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.



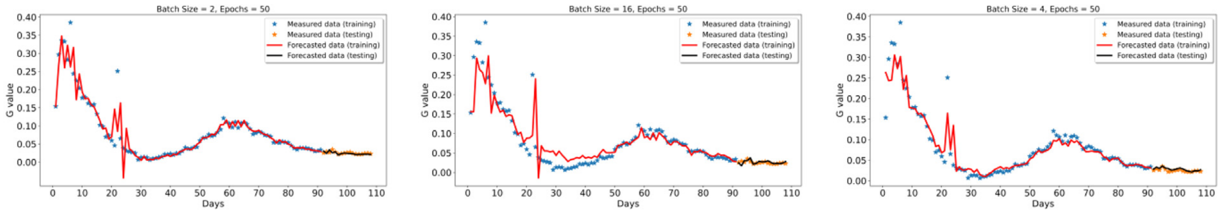
(a) (b) (c)

Fig. 12. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using CBOW features extraction method for ODANN with 7 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.



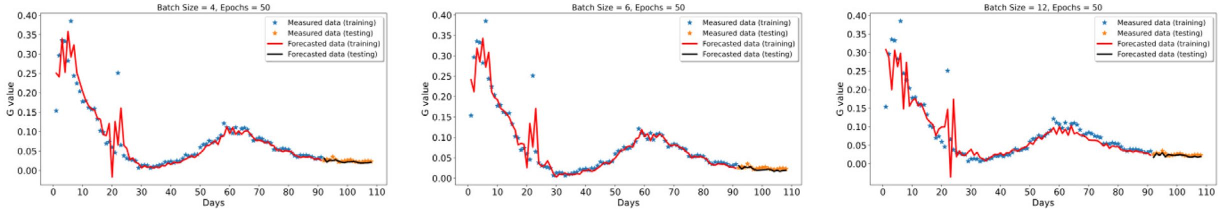
(a) (b) (c)

Fig. 13. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using Skip-Gram features extraction method for ODANN with 3 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.



(a) (b) (c)

Fig. 14. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using Skip-Gram features extraction method for ODANN with 5 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.



(a) (b) (c)

Fig. 15. Comparison between predicted and measured G values for 25 Jan 2020 to 11 May 2020 using Skip-Gram features extraction method for ODANN with 7 days rolling time-window, coupled with varying vector size for input features layer: (a) vector size = 100; (b) vector size = 500; (c) vector size = 5000.

Table 5
Summary of lowest MSE, RMSE, and MAE values derived for ODANN, with data assimilation component, using different NLP features extraction methods.

Rolling time-window	NLP features extraction method	Batch size	Vector size	Error scores on validation dataset			Error scores on testing dataset		
				MSE	RMSE	MAE	MSE	RMSE	MAE
3 days	TfidfVectorizer	2	94 237	0.000221	0.0149	0.00697	0.0000360	0.00603	0.00444
	CBOW	2	100	0.00195	0.0442	0.0195	0.0000180	0.00425	0.00314
	CBOW	6	500	0.00150	0.0387	0.0190	0.0000250	0.00497	0.00411
	CBOW	4	5000	0.00117	0.0343	0.0176	0.0000210	0.00454	0.00358
	Skip-gram	**6**	**100**	**0.00124**	**0.0352**	**0.0177**	**0.0000130**	**0.00361**	**0.002972**
	Skip-gram	4	500	0.00127	0.0356	0.0181	0.0000190	0.00434	0.00369
	Skip-gram	6	5000	0.00112	0.0334	0.0166	0.0000250	0.00500	0.00395
5 days	TfidfVectorizer	8	94 237	0.000237	0.0154	0.00728	0.0000610	0.00781	0.00620
	CBOW	2	100	0.000814	0.0285	0.0153	0.00000829	0.00288	0.00225
	CBOW	**2**	**500**	**0.000621**	**0.0249**	**0.0111**	**0.0000080**	**0.00282**	**0.00214**
	CBOW	8	5000	0.000588	0.0242	0.01118	0.0000130	0.003552	0.00295
	Skip-gram	2	100	0.00104	0.0322	0.0150	0.0000150	0.00387	0.00287
	Skip-gram	16	500	0.00155	0.0394	0.0218	0.0000210	0.00460	0.00325
	Skip-gram	4	5000	0.000613	0.0248	0.0127	0.0000110	0.00327	0.00273
7 days	TfidfVectorizer	16	94 237	0.000527	0.0230	0.00983	0.0000310	0.00559	0.00476
	CBOW	**4**	**100**	**0.000401**	**0.0200**	**0.0122**	**0.00000918**	**0.00303**	**0.00247**
	CBOW	8	500	0.00100	0.0317	0.0188	0.0000220	0.00470	0.00373
	CBOW	6	5000	0.000781	0.0279	0.0145	0.0000210	0.00458	0.00386
	Skip-gram	4	100	0.00117	0.0342	0.0170	0.0000200	0.00447	0.00363
	Skip-gram	6	500	0.00102	0.0320	0.0147	0.0000230	0.00483	0.00412
	Skip-gram	12	5000	0.00130	0.0361	0.0189	0.0000210	0.00453	0.00374

Note: The values in bold indicate the best prediction model with the specific rolling time-window.

Table 6
Summary of cross-validating ODANN with varying sizes of training, validation, and testing datasets from original data pool ranging between 25 Jan 2020 and 11 May 2020.

Training & Validation data size	Testing data size	Error scores on validation dataset			Error scores on testing dataset		
		MSE	RMSE	MAE	MSE	RMSE	MAE
60%	40%	0.00122	0.0349	0.0218	0.0000157	0.00396	0.000305
70%	30%	0.00101	0.0318	0.0180	0.0000130	0.00361	0.000279
80%	20%	0.000856	0.0293	0.0153	0.0000110	0.00332	0.000256
85%	15%	0.000621	0.0249	0.0111	0.0000080	0.00282	0.000214

consists of the following sequential steps: **(Data Hydration)** extractions of COVID-19 Twitter text data from a pool of Tweets IDs collated from the relevant historical day coupled with the inherent 1 day lead-time (e.g. 29 April 2020); **(Data Pre-Processing)** encoding of the extracted text data from the relevant historical day into numerically useful input features layer having an optimal vector size (e.g. input layer of 500 neurons in size); **(Trained Model Restoration)** calling upon the pre-trained ODANN model which takes in the encoded input features layer while also assimilating with historical G values, based on the pre-defined rolling

Table 7
Summary of average computational runtime for each key step in proposed method in near real-time context.

Step	Average runtime
Data hydration (∼1 million of COVID-19 Twitter text data daily)	6 h
Data pre-processing to derive input feature layer	20 min
Trained model restoration	20 s
Model predictions using input feature layer, coupled with data assimilation	10 s

time-window size (e.g. 5 days), to forecast the G-value on the given day in near real-time. Table 7 summarizes the average computational runtime for each of the key steps in the near real-time context. The listed runtimes in Table 7 show that data hydration requires the longest time to be completed, where the remaining steps can be accomplished with a good efficiency. At this stage, we have not explored any forms of data parallelism to accelerate the computational runtime, especially for the data hydration. This remains to be explored in our continual experiments in future studies. Overall, the total runtime to perform all steps listed in Table 7 takes an average of 6.4 h maximum on any given day in near real-time, hence the imposition of the lead-time of 1 day is more than sufficient to ensure that predictions of G-value on the given day can be performed with both efficiency and accuracy.

5. Discussions

5.1. Comparison of ODANN with alternative time-series models

To further validate the effectiveness of ODANN with its data assimilation component, several comparative experiments were performed with classical time-series prediction models or ML algorithms to forecast the same G parameter over time. In addition, the robustness of ODANN was further tested by setting different percentages of random missing quantities in the datasets for the combined model training, validation, and testing steps. In the following, we first discuss on the comparison analysis of ODANN with the other time-series models.

On a whole, the proposed ODANN model yielded the most satisfactory results in forecasting the G values over time, with a defined lead-time of 1 day, under the evaluation metrics of MSE, RMSE, and MAE. For an extensive comparison, several candidates, including ARIMA, AutoARIMA, Prophet, Random Forest (RF), Support Vector Machine (SVM), and Long Short-Term Memory (LSTM) algorithms [41] were also leveraged to perform the same combined training, validation, and testing phases, i.e. 85% of the total data quantity for model training and validation, and 15% for model testing, to directly model the dynamic behaviour of the proposed G parameter over time. However, without considering any Twitter data as representative of the general community's emotional responses towards the current pandemic, they still maintaining the same lead-time of 1 day to forecast the G value on any given day between 25 January 2020 and 11 May 2020. The resulting prediction performances derived from the alternative time-series models on the testing dataset are summarized in Fig. 16 and Table 8, while Tables 9a and 9b summarize the statistical comparison results between ODANN and the alternative time-series models on the same testing dataset.

The statistical analyses are namely, **Type 1** paired T-test, Kruskal–Wallis test, and Wilcoxon Signed Rank test, which mainly involve the *p*-value computations, as summarized in Table 9a, to estimate the probability of obtaining the results that are at least as extreme as the results actually observed, under the assumption that the respective null hypothesis is correct while assuming a significance value of 0.05; **Type 2:** Pearson correlation coefficient, Spearman correlation coefficient, and Kendall's tau, which focus on estimating different coefficient values, ranging between 0 and 1 as summarized in Table 9b, which quantify the level of association between 2 sets of time-series predictions. In summary, the statistical analyses in both Tables 9a and 9b indicate the following:

- **Paired T-test**: The null hypothesis is that the predictions from ODANN, on the testing dataset, have identical average, i.e. expected, values with that of the other alternative time-series models. The summarized p-values in Table 9a for the t-test indicate that the average values of the predictions from RF, SVM, ARIMA, and AutoARIMA model were most similar to that of ODANN's since their p-values are greater than 0.05, while the average values of the predictions from LSTM and Prophet differed significantly in their average values as the null hypothesis can be rejected with p-values less than 0.05.
- **Kruskal–Wallis test**: The null hypothesis is that the median value of the predictions from ODANN, on the testing dataset, was equal to that of the other alternative time-series models. The summarized p-values in Table 9a for the Kruskal–Wallis test indicate that the median value of the predictions from ARIMA, AutoARIMA, RF, SVM and LSTM models were most similar to that of ODANN's since their p-values are greater than 0.05, while the median value of the predictions from Prophet differed significantly as the null hypothesis can be rejected with p-values less than 0.05.
- **Wilcoxon Signed Rank test**: The null hypothesis is that the predictions from ODANN, on the testing dataset, have the same data distribution from that of the other alternative time-series models. The summarized p-values in Table 9a for the Wilcoxon Signed Rank test indicate that the predictions from RF, SVM, ARIMA and AutoARIMA had the most similar distribution to that of ODANN's since their p-values are greater than 0.05, while the predictions from the remaining models differed significantly in their data distribution as the null hypothesis can be rejected with p-values less than 0.05. Specifically, the respective distributions for the time-series differences between the predictions from LSTM or Prophet, with that of ODANN were not symmetric about the zero-value point.
- **Pearson correlation coefficient**: Measures the level of linear relationship between the predictions from ODANN, on the testing dataset, and that of the other alternative time-series models. The summarized coefficient values in Table 9b for the Pearson correlation analysis indicate that the predictions values derived via the LSTM method had the closest linear relationship, however not necessarily identical, with that of ODANN due to its highest 0.839 coefficient value. The other coefficient values suggest that the predictions from the remaining models generally do not correlate linearly, to a significant extent, with that of ODANN's.
- **Spearman correlation coefficient**: Measures the monotonicity of the relationship between two datasets. Similar conclusion from the preceding Pearson coefficient can also be made when using the Spearman correlation coefficient (see Table 9b), where the predictions from the LSTM model had the closest positive monotonic relationship with that of ODANN's when using the testing dataset. Overall, the predictions from Prophet had the lowest monotonic correlation with that of ODANN's on the same testing dataset. Additional discussion on Prophet's result will be made subsequently.
- **Kendall's tau**: The Kendall's tau value measures the correspondence between ODANN's predictions, using the testing

dataset, and the corresponding predictions from the other time-series models. Overall, the computed tau values indicate that no significant strong agreement exists between the predictions made by ODANN and the other time-series models on the same testing dataset. Relatively, only the predictions from LSTM had the closest agreement with that of ODANN.

- Overall, the above-discussed statistical analyses indicate that there is no universal statistical analysis test capable of pre-determining which time-series model can generate test predictions to best match with that of ODANN. Each test serves a different objective, as depending on the context of the problem. Using the RMSE and MAE scores summarized in Table 8, the test predictions from SVM can best match with the predictive accuracy of ODANN, and the computed scores for Paired T-test, Kruskal–Wallis test, and Wilcoxon Signed Rank test (Table 9a) indicate that SVM's predictions were consistently similar with that of ODANN's in terms of the average (expected), median, and data distribution parameters. However, we note that the latter model still outperformed SVM in the final prediction accuracy, i.e., minimizing the RMSE and MAE scores, when using the same testing dataset.

As illustrated in Fig. 16, the trend of the prediction curves deriving from the alternative time-series models, with the exception of Prophet, was reasonably consistent with the measured G values (ground truth), which suggests that the alternative models can be reasonably useful in forecasting the growth rate in the number of confirmed COVID-19 cases on a global scale. The compared results listed in Table 8 reveal the following ranked prediction performance of the 6 alternative time-series models: ODANN > SVM > ARIMA > LSTM > RF > AutoARIMA > Prophet. With the present dataset, ODANN outperformed the other traditional models with at least 0.00100 reductions in the average RMSE and MAE values, respectively, hence indicating ODANN's capability to better encapsulate the growth trajectories of the analysed G parameter over time. The results also highlighted the likelihood that the community's emotional responses towards the pandemic do affect, to an extent, the temporal variations of the G parameter since its inception, especially when coupled with the data assimilation component in ODANN. Therefore, there are valuable knowledge embedded in the extracted high-level features using the different NLP features extraction methods, which can potentially offer useful guidance to the different stakeholders to obtain more accurate predictions pertaining to the proposed G parameter.

On the other hand, we note that Prophet generally performed less satisfactorily as compared to the other models, inclusive of ODANN, due to the need for the training dataset to have some level of seasonality component, whether is it daily, weekly, monthly or yearly in nature. At this stage, it is obvious that the time-series profile for G parameter (see Fig. 3) do not quite follow any seasonal trends due to the relatively small dataset being analysed at this stage, hence it is difficult for Prophet algorithm to capture any observable seasonal trends during its training phase. As discussed, the rate of increase in the G values occurred in the approximate periods of Day 0 to Day 5 and Day 40 to Day 60, while the corresponding rate of decline occurred approximately during Day 10 to Day 30 and Day 60 to Day 85, hence there was no strong seasonality or trend components in the variations of the G parameter over time with the present dataset. We would, however, expect that the Prophet model can improve on its present predictive accuracy with a larger data pool for the combined model training, validation, and testing steps in any future works as more data is being made available. Finally, it is worth noting that the same results observations have also been reported by recent studies [42,43] where the authors' Prophet model performed less ideally as compared to that of the other time-series models such as ARIMA, TBAT, etc., in modelling and forecasting the transmission rate of COVID-19.

Table 8
Evaluation of predicted results from ODANN, ARIMA, AutoARIMA, RF, SVM, LSTM, and Prophet time-series models using the testing dataset.

Model	Parameter	RMSE	MAE
ARIMA	Order (1, 1, 1)	0.00412	0.00336
AutoARIMA	Order (0,1,1)(3,1,1) Daily seasonality	0.00639	0.00468
RF	Number of trees = 8 Maximum depth of the tree = 4	0.00447	0.00358

SVM	Kernel = 'rbf'
	Regularization parameter = 10
	Epsilon = 0.001
		0.00388	0.00315

LSTM	Number of hidden layers = 1 Number of neurons = 10	0.00430	0.00356
Prophet	Width of the uncertainty intervals = 0.95 No. of simulated draws for uncertainty intervals = 35 Growth = linear Daily seasonality	0.0300	0.0294
ODANN	Input features layer (via CBOW) = 500 Epochs = 50 Batch size = 2 Rolling time-window size (data assimilation) = 5 days	0.00282	0.00214

ODANN consistently achieved the highest predictive accuracy from its testing step, even when random missing values were added into the dataset. Missing data commonly occurs during data collection/collation, which can generally reduce the overall representation of the samples and even cause biased estimations. Therefore, it is important to investigate how the proposed ODANN model, coupled with data assimilation component, and the other alternative time-series models respond to different percentages of missing data in their respective prediction task. To set the missing data condition, we randomly dropped 10%, 20%, and 30% of the G values data from the original dataset, and then simply filled out the corresponding missing values using its available previous day of observed G value. For the computational experiments, we randomly generate 30 combinations for each of the above-listed missing percentages of the data points for the G value. After the required data imputations for each of the combination, we maintained the same respective optimal model configurations, as listed in Table 8, for each of the time-series model, inclusive of ODANN, to forecast the same G parameter over time with the same lead-time of 1 day. Note that ODANN still leveraged on the optimal batch size of 4, input features layer size of 500 for CBOW features extraction method, and a rolling time-window size of 5 days for assimilating the historical time records for the G parameter to perform the prediction step.

Tables 10 and 11 summarizes the average RMSE and MAE scores, coupled with their standard deviation values (also see Figs. 17 and 18), computed for all time-series models from the 30 random combinations for each of the missing data percentages as shown. As expected, the accuracy gradually reduced with an increasing amount of missing data. Under the condition of 10% missing data, ODANN remained more accurate in its predictions on the testing dataset, with RMSE and MAE scores of 0.00315 and 0.00246 respectively, as compared to that of the other models. Both computed scores were generally lower than the 2nd lowest corresponding values attained from the SVM model (the next best prediction model). As the percentage of missing data increased, the comparative differences for the RMSE and MAE between our proposed model and SVM, however, reduced gradually as shown in Tables 10 and 11. Notwithstanding the missing data

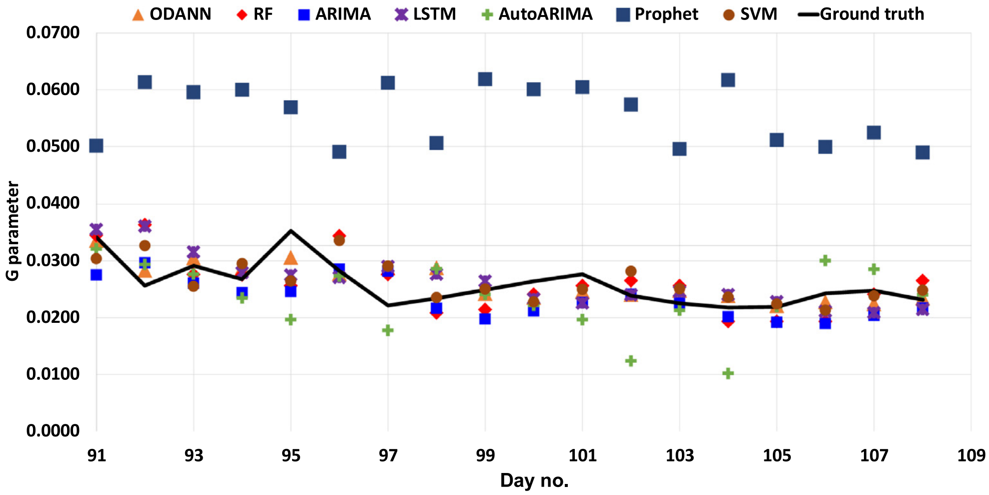


Fig. 16. Prediction of G value on the testing dataset using ODANN, ARIMA, AutoARIMA, RF, SVM, LSTM, and Prophet time-series models.

Table 9a
Summary of p-values to compare ODANN with other time-series models for predictions on testing dataset by assuming significance value of 0.05.

Statistical analysis method	RF	SVM	ARIMA	LSTM	AutoARIMA	Prophet
Paired T-test	1.36E−01	5.20E−02	5.83E−02	2.33E−02	1.68E−01	6.95E−17
Kruskal–Wallis test	4.97E−01	1.51E−01	1.41E−01	5.80E−01	3.33E−01	2.89E−08
Wilcoxon signed rank test	1.79E−01	9.58E−02	8.88E−02	3.19E−02	3.38E−01	9.54E−07

Table 9b
Summary of correlation coefficient values to compare ODANN with other time-series models for predictions on testing dataset.

Statistical analysis method	RF	SVM	ARIMA	LSTM	AutoARIMA	Prophet
Pearson correlation coefficient	6.77E−01	5.89E−01	6.74E−01	8.48E−01	3.71E−01	2.36E−01
Spearman correlation coefficient	5.76E−01	5.26E−01	6.01E−01	7.22E−01	4.04E−01	1.29E−01
Kendall’s tau	4.20E−01	3.61E−01	4.23E−01	5.54E−01	3.04E−01	1.06E−01

quantity for the random combinations, ODANN appeared to provide the highest accuracy capability with the lowest RMSE and MAE average values after the 30 experiments for each of the missing data percentages investigated. In all missing data scenarios experimented, there was no obvious change in the resulting accuracy of ARIMA and SVM, hence indicating the convergence in the predictive accuracy of the models. On the contrary, it can be observed from their corresponding length of the plot boxes (Figs. 17 and 18) and computed standard deviation values that RF, LSTM, AutoARIMA, Prophet, and ODANN generally experienced more fluctuations, hence their prediction results may encompass greater level of uncertainty. This, however, can be appropriately addressed by running more random combinations of the missing data percentages in the future studies.

5.2. Novelty of ODANN model

We further demonstrate the novelty of ODANN model by considering the proposed model’s capability to assimilate additional data features of any pre-defined characteristics into the optimal locations of the available hidden layers within the DNN model. Previously, we have shown the effectiveness of assimilating 5 days of historical time-records, i.e. $(t-2, t-3, t-4, t-5, t-6)$, for the G parameter with the inherent lead-time of 1 day, where we can best minimize the RMSE and MAE scores as compared to the other time-series models as summarized in Table 8. To further improve on ODANN’s predictive accuracy to forecast the G parameter on a given day, we consider the scenario where we assimilate ODANN’s hidden layers with relevant socioeconomic factors and restrictive government policies pertaining to COVID-19 for the same modelling step. The additional factors to be considered are listed in Table 12 for the same period between 25 January 2020 and 11 May 2020. For all factors, except for the government stringency index, they are typically represented as non-discrete values, i.e. not time-series continuous values, as shown in Table 12. Hence, we normalize the classes into discrete values via Equation (5) which considers the data distribution among the different classes.

$$\overline{X} = \frac{X-\mu}{\sigma} \tag{5}$$

where $\overline{X}$ represents the normalized class discrete value, X is the class score (e.g. Class 1, 2, 3, etc.), μ the mean class value, and σ the standard deviation of the class scores for the respective factor/policy. We note that the government stringency indexes are also normalized via Equation (5) to ensure similar scaling values for all considered input data features.

Fig. 19 illustrates the design of ODANN to concurrently assimilate the historical time-series records for the G parameter and the listed socioeconomic factors and restrictive government policies, based upon the defined rolling time-window size, to model and forecast the G value on any given day with a lead-time of 1 day. We again maintain the same optimal model configuration (batch size = 2, input features layer size = 500, CBOW features extraction method, rolling time-window size = 5 days) to perform the new scenario of assimilating the additional data features from Table 12 for the modelling step. For example, by adhering to the same rolling time-window size of 5 days, the total number of new neurons for assimilating the additional data features into the selected hidden layer of ODANN equates to 55 as shown in Fig. 19. The same model training (and validation), and testing datasets of 85% and 15% respectively, are also maintained for the analysis. By including the additional data features, we can further reduce

Table 10
Comparison of RMSE under different percentage of missing data.

Method	RMSE under different percentages of missing data		
	10%	20%	30%
ARIMA	4.19E−03 (±3.37E−05)	4.21E−03 (±3.71E−05)	4.23E−03 (±4.88E−05)
AutoARIMA	7.96E−03 (±2.67E−03)	8.05E−03 (±3.62E−03)	9.35E−03 (±3.89E−03)
RF	4.69E−03 (±5.88E−04)	4.76E−03 (±5.51E−04)	4.99E−03 (±6.15E−04)
SVM	4.11E−03 (±1.57E−04)	4.13E−03 (±1.82E−04)	4.21E−03 (±2.33E−04)
LSTM	4.22E−03 (±6.30E−04)	4.46E−03 (±7.17E−04)	4.48E−03 (±7.75E−04)
Prophet	3.16E−02 (±3.66E−03)	3.21E−02 (±6.17E−03)	3.32E−02 (±7.25E−03)
ODANN	3.15E−03 (±1.60E−03)	4.10E−03 (±2.17E−03)	4.29E−03 (±6.13E−04)

Table 11
Comparison of MAE under different percentage of missing data.

Method	MAE under different percentages of missing data		
	10%	20%	30%
ARIMA	3.44E−03 (±3.98E−05)	3.45E−03 (±2.19E−05)	3.46E−03 (±6.23E−05)
AutoARIMA	6.66E−03 (±2.20E−03)	6.93E−03 (±3.47E−03)	8.09E−03 (±3.44E−03)
RF	3.53E−03 (±3.82E−04)	3.58E−03 (±3.87E−04)	3.64E−03 (±4.08E−04)
SVM	3.32E−03 (±1.32E−04)	3.33E−03 (±1.71E−04)	3.36E−03 (±2.06E−04)
LSTM	3.50E−03 (±6.34E−04)	3.61E−03 (±7.76E−04)	3.69E−03 (±7.73E−04)
Prophet	3.07E−02 (±3.76E−03)	3.13E−02 (±6.40E−03)	3.22E−02 (±7.46E−03)
ODANN	2.46E−03 (±1.32E−03)	3.30E−03 (±2.08E−03)	3.34E−03 (±5.79E−04)

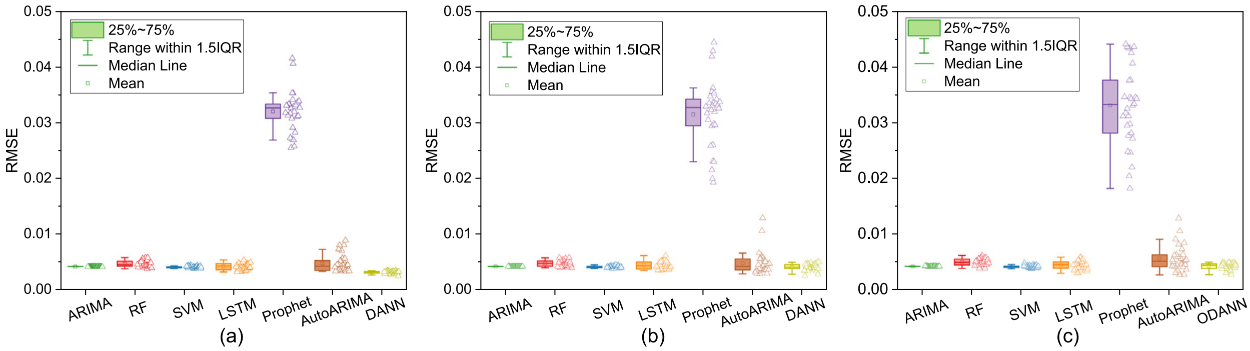


Fig. 17. Boxplot of RMSE under different percentages of missing data: (a) 10%; (b) 20%; (c) 30%.

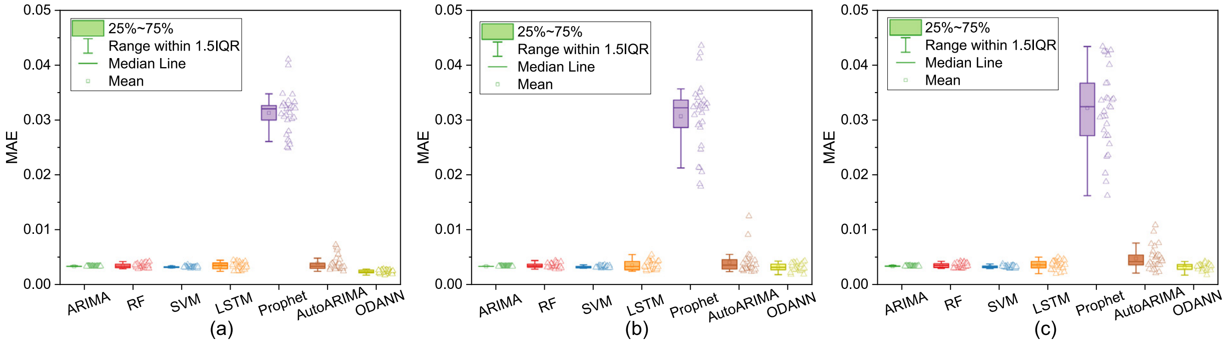


Fig. 18. Boxplot of MAE under different percentages of missing data: (a) 10%; (b) 20%; (c) 30%.

the original RMSE and MAE scores (from Table 8) for ODANN to around 0.002000 and 0.00154 respectively, hence improving the model's predictive accuracy for the near real-time predictions. Overall, the novelty of our proposed ODANN can be summarized as follows:

- The current model design of ODANN is built to enable diverse data components to be fused systematically and effectively, as part our data assimilation/fusion step, hence preventing any data component to outweigh the others. For example, we avoid directly assimilating the encoded semantic word representations derived from the Twitter data, having the defined vector size, into the same hidden layer as that of the historical time-records for the G parameter due to the differing scales in their respective 1D array sizes. The vector size for the encoded Twitter data and assimilated historical time-records for the G parameter are in the ordering scales of $O\left(10^1 - 10^3\right)$ and $O\left(10^0\right)$, hence the differing scales are likely to exert more weights on the encoded Twitter data during the training phase of ODANN when both are fused directly together. Therefore, the derived word representation (1D array) is instead leveraged as the input features layer for ODANN which enables the systematic aggregation of the original word vectors into higher-level useful features of smaller vector sizes (as shown in Figs. 5 and 19), along the depth of the ODANN model, having the

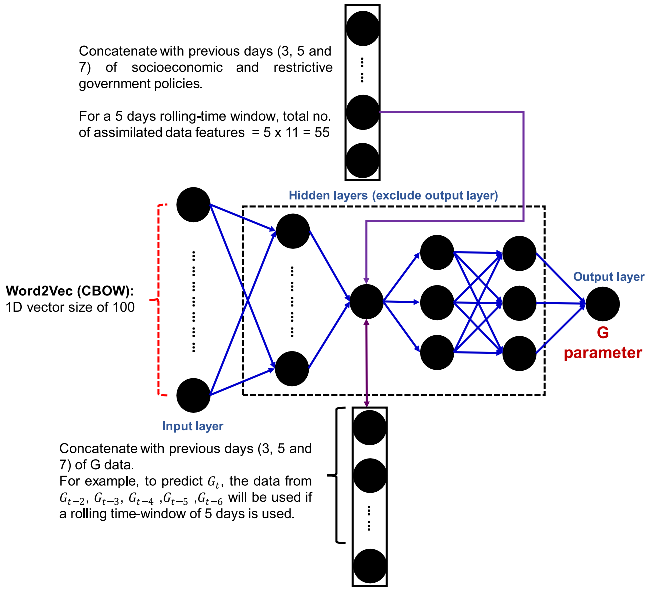

Concatenate with previous days (3, 5 and 7) of socioeconomic and restrictive government policies.
For a 5 days rolling-time window, total no. of assimilated data features = 5 x 11 = 55
Hidden layers (exclude output layer)
Output layer
Word2Vec (CBOW):
1D vector size of 100
G parameter
Input layer
Concatenate with previous days (3, 5 and 7) of G data.
For example, to predict G_t, the data from G_{t-2}, G_{t-3}, G_{t-4} ,G_{t-5} ,G_{t-6} will be used if a rolling time-window of 5 days is used.

Fig. 19. Illustration of ODANN to assimilate historical time-series records for G parameter and other socioeconomic factors and restrictive governmental policies (from Table 11) for any given rolling time-window size.

same scaling as that of the assimilated data features at the selected hidden layer of ODANN in Fig. 19. By doing so, we have shown that we can maximize the predictive accuracy of ODANN on the testing dataset for forecasting the G parameter with a lead-time of 1 day.

- In this latest scenario, we demonstrate the flexibility of ODANN to assimilate other important data features (from Table 12) which can be useful to enhance its predictive accuracy, while ensuring a well-balanced weights distribution among the different types of data features. While the number of assimilated neurons for the additional socioeconomic factors and restrictive government policies may exceed the number of assigned neurons for the aggregated word vectors and the historical time-series records for the G parameter, we highlight that each data feature in Table 12 is actually a unique parameter by itself hence there is still no one singular parameter being placed more significance than the others. Going forward, as more data is being available to the community which include new data features (vaccination rates, environmental factors, etc.), the present design of ODANN can easily assimilate those new features to perform the same modelling analysis with equal efficiency.
- While the design of ODANN may be relatively straightforward from neural network research, the better agreement achieved from ODANN as compared to other time-series models, as summarized in Tables 8, 10 and 11, clearly underline the effectiveness of our proposed workflow/approach by having a singular end-to-end network model to concomitantly process complex semantic word vectors, as representative of the society's emotional responses towards the pandemic, and a multitude of socioeconomic and governmental factors in a single-shot learning and validation process. Accuracy and computational efficiency have both been achieved with ODANN, and we are hopeful that the same workflow and model design can be extended to other domain problems which models a target objective as function of multiple data features of varying characteristics.

5.3. Comparison of ODANN with previous research studies

Besides comparing ODANN with other time-series prediction models as explained qualitatively and quantitatively in the preceding sub-section, we further validate the predictive accuracy of ODANN with recent similar notable studies [42–46], which too focused on forecasting the transmission rate of COVID-19, in terms of the number of confirmed COVID-19 cases, since the virus' inception. In the following, we outlined the key methodologies and reported results by the previous studies [42–46], for comparison with our proposed ODANN model and its prediction results.

Kumar and Susan [42] too leveraged on ARIMA and Prophet time-series prediction models to model the temporal data of COVID-19 spread worldwide, and for several countries in the different continents, for the period between 22 January 2020 and 20 May 2020. Overall, the authors demonstrated that ARIMA model was generally more effective for forecasting the prevalence rate of COVID-19, which too aligned with our present study where we have shown that ARIMA/AutoARIMA had resulting error scores (RMSE and MAE) which were a magnitude smaller than that of Prophet as previous summarized in Table 8. A more similar study in using social media to forecast the virus outbreak with neural ordinary differential equations (ODEs) was performed by Núñez et al. [44]. The authors' data comprised of a massive amount of online surveys, regarding COVID-19 symptoms, via Facebook to train and validate the authors' personalized neural ODE, followed by using the trained neural ODE to forecast the virus' outbreak rate in different US states for up to sixty days. No error metric scores were reported by the authors in their published paper, however, a visual inspection of their temporal plots for comparing their model's predictions and the respective monitored data during the extrapolation phase, i.e. model testing, indicates some level of differences between both sets of data. In addition, the authors did not extend their prediction model for the forecasting on the global scale. A more aggregated analysis was carried out by Yousefinaghani et al. [45] to detect spikes/waves in the number of confirmed cases in the United States and Canada using social media data and Google searches online. Their adopted lead-time adopted ranged between 1 and 2 weeks which can be useful to decision-makers to early detect any possible spikes, however, the authors did not specifically investigated or reported on the number of false positive (FPs) forecasted hence the exact precision of their method is still not known at this stage. Overall, there are currently limited studies reported in the current literature which leverages on the big-data availability on social media (Facebook, Twitter, etc.) to model and forecast the actual number of reported/confirmed COVID-19 cases globally, as presented in our present study.

The remaining 2 previous studies [43,46] provided a more comprehensive comparison with our present results attained from ODANN. We first note that each of the studies, inclusive of ours, models the growth rate in the number of confirmed COVID-19 cases globally via different means. Papastefanopoulos et al. [43] modelled the number of active cases per unit population size in the top 10 countries having the largest number of reported COVID-19 cases as of 4 May 2020. The countries include the United States (US), Spain, Italy, United Kingdom (UK), France, Germany, Russia, Turkey, Brazil, and Iran. Their study is selected for comparison since the above-listed countries encompass the most significant portion (>70%) of the confirmed COVID-19 cases globally, which has also been underlined by the authors in their paper. They reported the respective predictive performance using different time-series prediction models (ARIMA, Prophet, HWAAS, NBEATS, Gluonts, and TBAT) for the 10 listed countries by computing the RMSE score (same of Eq. (3)) based on the predictions

Table 12
List of socioeconomic factors and restrictive government policies for additional data features assimilation into ODANN.

Type	Data features	Value range
Socioeconomic	Level of income support	**3 classes** (no support, support >50% loss salary, support <50% loss salary)
	Level of debt relief	**3 classes** (no relief, moderate relief, large relief)
	Face covering	**5 classes** (no policy, recommended, required in some public places, required in all public places, always required outside of homes)
	Cancellation of public events and gatherings	**3 classes** (no cancellations, recommended cancellations, compulsory cancellations)
	Testing and contact tracing	**4 classes** (no testing, testing for those with symptoms and belonging to higher-risk groups, testing for anyone with symptoms, open public testing)
	Public information campaigns	**3 classes** (none, public officials urging caution, coordinated information campaigns)
Restrictive government policies	Government stringency index	**0–100** (100 being the highest stringency score)
	Schools and workplaces closures	**4 classes** (no closures, recommended, recommended at some levels, required at all levels)
	Stay-at-home restrictions	**4 classes** (no measures, recommended, required except for running essentials, required with few exceptions)
	Controls for domestic travels	**3 classes** (no measures, recommended movement restrictions, restricted movement)
	Controls for international travels	**5 classes** (no measures, screening, quarantine from high-risk countries, ban on high-risk countries, total border closure)

made from their model testing step using data instances between 28 April 2020 and 4 May 2020. Table 13 summarizes the RMSE scores from each of their method used, where the respective RMSE score, as shown in the table, represented the average of all 10 countries combined from the authors' model testing step. Our updated RMSE scores in Table 13 were based upon same the same working principles as that of [43] where we re-normalized the reported and forecasted confirmed number of COVID-19 cases, from our prediction step using ODANN, by the total world population ($\approx$ 7.67 billion people as of 2020/2021) for the same period analysed by the authors. We note that prior de-normalization of the predicted and reported G values were performed by using the inverse version of Eq. (1). Also, the de-normalization step was carried out after the predictions were made on the proposed G parameter using our ODANN model, as previously summarized in the preceding section(s). By performing the appropriate estimations, we can then better compare and evaluate our model performance from ODANN's with that of [43] in the same ordering scale. The listed RMSE scores in Table 13 show that our proposed ODANN, with the optimized model configuration (5 days rolling time-window size, 500 neurons for input layer, etc.) is more likely to perform better than the other time-series models from the previous study, and also in our current analysis from Table 8 as demonstrated earlier, when modelling the contributory influences of large-scale social media data as the input features layer to ODANN, and also assimilating with the historical records for the G parameter and/or other important socioeconomic and governmental factors/data if necessary.

Table 13
Comparative analysis of ODANN's predictive accuracy with previous studies.

Method	RMSE	MAPE
ARIMA [43]	1.75E−02	**-Nil-**
Prophet [43]	3.00E−02	
HWAAS [43]	2.40E−02	
BNEATS [43]	2.11E−02	
Gluonts [43]	4.45E−02	
TBAT [43]	6.99E−03	
Simple-time series model [46]	**-Nil-**	0.200%
ODANN w/o SEG[a] factors (optimized configuration)	3.45E−03	0.205%
ODANN with SEG[a] factors (optimized configuration)	2.63E−03	0.154%

[a] SEG factors represent socioeconomic and governmental restrictive policies/factors.

On the other hand, Petropoulos et al. [46] adopted a simpler time-series model, namely the non-seasonal multiplicative error and multiplicative trend exponential smoothing model (ETS-MMN), by modelling and forecasting the global numbers of confirmed COVID-19 cases and deaths with different number of horizons, i.e. lead-times. We focused on their results derived from the 1-day horizon to match with our present analysis due to the inherent lead-time of 1 day adopted to perform the required predictions. The authors leveraged on the mean absolute percentage error (MAPE) score, as defined in Eq. (6), to evaluate their forecasting results for the period between 10 February 2020 and 30 May 2020 using different time horizons. For example, a 10-days horizon indicates that the predictions were made on 10 February 2020, 20 February 2020, etc., with a 10 days interval. By far, their MAPE score for the period of May 2020, which aligned with the dates for the testing phase of ODANN in our present study, was reported to be approximately 0.200% with the 1-day time horizon. To better compare and evaluate ODANN's predictions with that of [46], we again de-normalized the reported and predictions values made from ODANN by again using the inverse version of Eq. (1), followed by computing the appropriate MAPE scores using Eq. (6). Again, we note that the de-normalization step was carried out after the predictions were made on the proposed G parameter using ODANN. Overall, we can observe that the predictions derived from the simpler time-series model as proposed by Petropoulos et al. can match almost exactly with that of our ODANN model with the data assimilation component, where their computed MAPE scores were almost identical as shown in Table 13. However, by fusing with data features pertaining to the socioeconomic and governmental factors, the resulting MAPE score from ODANN can be reduced further by around 0.045%, hence improving the forecasting step.

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{\left|G_{p,i} - G_{m,i}\right|}{G_{p,i}} \times 100\% \tag{6}$$

6. Conclusions and future works

In this paper, we have developed a hybrid deep learning model, termed as ODANN, which effectively combines features extraction methods, via natural language processing (NLP), and data assimilation concept to accurately predict the daily growth rate in the number of confirmed COVID-19 cases globally, via a proposed G parameter, with a lead-time of 1 day. NLP features extraction methods were leveraged to pre-process large volumes of Twitter data (100 million in exceedance) to derive high-level semantic word vectors to quantify the general community's emotional responses towards the current pandemic. Coupled with data assimilation, we demonstrated that ODANN can outperform traditional time-series models, including ARIMA, RF, SVM, LSTM, AutoARIMA, and Prophet, in forecasting the G parameter ranging between 25 January 2020 and 11 May 2020. Specifically, by learning from the historical time-series records for the G parameter using a rolling time-window size of 5 days for the data assimilation component, and fusing with the aggregated word vectors derived from large volumes of Twitter data at specific hidden layer(s) of the deep learning model which ensured a well-balanced weights distributions of the data features, ODANN can forecast the G parameter, with a lead-time of 1 day, having average RMSE and MAE scores of 0.00282 and 0.00214 respectively.

The novel contributions from the study mainly consist of the following aspects, namely: (a) combining NLP features extraction methods with neural network prediction model to forecast the global spread of COVID-19 over time. By considering the community's daily aggregated emotional responses towards the pandemic, our proposed ODANN model shows superiority in both accuracy and robustness, even towards missing data conditions. Information from Twitter offers valuable insights into people's emotional responses towards COVID-19, which has been proven to be useful to maximize the accuracy performance of our predictive model when coupled with data assimilation at the selected hidden layer(s) of the deep learning model. Moreover, ODANN can still maintain relatively low error scores, even when 30% of random missing values were artificially introduced into the original dataset; and (b) ability of ODANN to easily assimilate or fuse with other socioeconomic and governmental factors of varying characteristics, which can further enhance the model's predictive accuracy by ensuring a well-balanced weightage distributions among the assigned neurons in the deep learning model. We note that this model design or framework for assimilating different types of data features, while balancing the resulting weightage distributions for the different types of data features within a single-shot learning process, has not been explored by far for COVID-19 related predictions.

In our future works, we intend to assimilate contemporary data after 11 May 2020 to perform near real-time predictions. Since the sentiments towards COVID-19 change rapidly, we can conduct semantic analysis on informative texts from other popular social media platforms (i.e., Twitter, Facebook, Weibo, etc.). The proposed approach has great potential to reveal the psychological responses of the public towards COVID-19, hence identifying possible social concerns to forecast the global emotional evolution towards the current pandemic. Besides, social network analysis (SNA) based upon graph theory is another method to monitor the spread of information on social media [47]. For example, we can perform SNA on COVID-19 related Twitter data to examine the dynamic and complex patterns of public health information to assist governments to control the volume of false or inaccurate information pertaining to COVID-19 on the internet.

CRediT authorship contribution statement

Alvin Wei Ze Chew: Conceptualization, Data curation, Resources, Methodology, Software, Formal Analysis, Validation, Investigation, Writing - Original Draft, Writing - review & editing. **Yue Pan:** Methodology, Software, Formal analysis, Validation, Investigation, Writing – original draft. **Ying Wang:** Data curation, Resources, Software, Formal analysis, Validation. **Limao Zhang:** Conceptualization, Investigation, Resources, Writing – review & editing.

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 study was supported in part by Microsoft Corporation for the AI for Health COVID-19 Azure Compute Grant of ID:00011000272 and the Start-Up Grant at Nanyang Technological University, Singapore (No. 04INS000423C120).

References

[1] C. Huang, et al., Clinical features of patients infected with 2019 novel coronavirus in wuhan, China, Lancet 395 (10223) (2020) 497–506, http://dx.doi.org/10.1016/S0140-6736(20)30183-5.

[2] I. Ali, O.M.L. Alharbi, COVID-19: Disease, management, treatment, and social impact, Sci. Total Environ. 728 (2020) 138861, http://dx.doi.org/10.1016/j.scitotenv.2020.138861.

[3] R. Aldaco, et al., Food waste management during the COVID-19 outbreak: a holistic climate, economic and nutritional approach, Sci. Total Environ. 742 (2020) 140524, http://dx.doi.org/10.1016/j.scitotenv.2020.140524.

[4] A. Chew, Y. Wang, l. Zhang, Correlating dynamic climate conditions and socioeconomic-governmental factors to spatiotemporal spread of COVID-19 via semantic segmentation deep learning analysis, Sustain. Cities Soc. 75 (2021) 103231, http://dx.doi.org/10.1016/j.scs.2021.103231.

[5] T. Liu, et al., Time-varying transmission dynamics of novel coronavirus pneumonia in China, BioRxiv (2020) http://dx.doi.org/10.1101/2020.01.25.919787, 2020.01.25.919787.

[6] Y. Pan, L. Zhang, Z. Yan, M.J. Lwin M.O., Discovering optimal strategies for mitigating COVID-19 spread using machine learning: Experience from Asia, Sustain. Cities Soc. 75 (2021) 103254–103306, http://dx.doi.org/10.1016/j.scs.2021.103254.

[7] S. Unkel, C.P. Farrington, P.H. Garthwaite, C. Robertson, N. Andrews, Statistical methods for the prospective detection of infectious disease outbreaks: a review, J. R. Stat. Soc. Ser. A (Stat. Soc.) 175 (1) (2012) 49–82, http://dx.doi.org/10.1111/j.1467-985X.2011.00714.x.

[8] E.B. Postnikov, Estimation of COVID-19 dynamics 'on a back-of-envelope': Does the simplest SIR model provide quantitative parameters and predictions? Chaos Solitons Fractals 135 (2020) 109841, http://dx.doi.org/10.1016/j.chaos.2020.109841.

[9] H.G. Hong, Y. Li, Estimation of time-varying reproduction numbers underlying epidemiological processes: A new statistical tool for the COVID-19 pandemic, PLoS One 15 (7) (2020) e0236464, [Online]. Available: https://doi.org/10.1371/journal.pone.0236464.

[10] L. Browning, et al., Digital pathology and artificial intelligence will be key to supporting clinical and academic cellular pathology through COVID-19 and future crises: the pathlake consortium perspective., J. Clin. Pathol. 74 (7) (2021) 443–447, http://dx.doi.org/10.1136/jclinpath-2020-206854.

[11] K. Hou, T. Hou, L. Cai, Public attention about COVID-19 on social media: An investigation based on data mining and text analysis., Pers. Individ. Differ. 175 (2021) 110701, http://dx.doi.org/10.1016/j.paid.2021.110701.

[12] S.-F. Tsao, H. Chen, T. Tisseverasinghe, Y. Yang, L. Li, Z.A. Butt, What social media told us in the time of COVID-19: a scoping review, Lancet Digit. Health 3 (3) (2021) e175–e194, http://dx.doi.org/10.1016/S2589-7500(20)30315-0.

[13] M. Haman, The use of Twitter by state leaders and its impact on the public during the COVID-19 pandemic, Heliyon 6 (11) (2020) e05540, http://dx.doi.org/10.1016/j.heliyon.2020.e05540.

[14] D.E. O'Leary, Twitter Mining for discovery, prediction and causality: applications and methodologies, Intell. Syst. Account. Financ. Manag. 22 (3) (2015) 227–247, http://dx.doi.org/10.1002/isaf.1376.

[15] F.E. Ayo, O. Folorunso, F.T. Ibharalu, I.A. Osinuga, Machine learning techniques for hate speech classification of twitter data: State-of-the-art, future challenges and research directions, Comput. Sci. Rev. 38 (2020) 100311, http://dx.doi.org/10.1016/j.cosrev.2020.100311.

[16] A. Signorini, A.M. Segre, P.M. Polgreen, The use of Twitter to track levels of disease activity and public concern in the U.S. during the influenza a H1n1 pandemic, PLoS One 6 (5) (2011) e19467, [Online]. Available: https://doi.org/10.1371/journal.pone.0019467.

[17] H. Hirose, L. Wang, Prediction of infectious disease spread using Twitter: A case of influenza, in: 2012 Fifth International Symposium on Parallel Architectures, Algorithms and Programming, 2012, pp. 100–105, http://dx.doi.org/10.1109/PAAP.2012.23.

[18] J.C. Santos, S. Matos, Analysing Twitter and web queries for flu trend prediction, Theor. Biol. Med. Model. 11 Suppl 1 (Suppl 1) (2014) S6, http://dx.doi.org/10.1186/1742-4682-11-S1-S6.

[19] L. Sinnenberg, A.M. Buttenheim, K. Padrez, C. Mancheno, L. Ungar, R.M. Merchant, Twitter As a tool for health research: A systematic review, Am. J. Public Health 107 (1) (2017) e1–e8, http://dx.doi.org/10.2105/AJPH.2016.303512.

[20] R. Agerri, X. Artola, Z. Beloki, G. Rigau, A. Soroa, Big data for natural language processing: A streaming approach, Knowl.-Based Syst. 79 (2015) 36–42, http://dx.doi.org/10.1016/j.knosys.2014.11.007.

[21] T. Davenport, R. Kalakota, The potential for artificial intelligence in healthcare, Future Healthc. J. 6 (2) (2019) 94–98, http://dx.doi.org/10.7861/futurehosp.6-2-94.

[22] B.R. Beck, B. Shin, Y. Choi, S. Park, K. Kang, Predicting commercially available antiviral drugs that may act on the novel coronavirus (SARS-CoV-2) through a drug-target interaction deep learning model., Comput. Struct. Biotechnol. J. 18 (2020) 784–790, http://dx.doi.org/10.1016/j.csbj.2020.03.025.

[23] C. Li, Y. Yang, H. Liang, B. Wu, Transfer learning for establishment of recognition of COVID-19 on CT imaging using small-sized training datasets, Knowl.-Based Syst. 218 (2021) 106849, http://dx.doi.org/10.1016/j.knosys.2021.106849.

[24] F. Shi, et al., Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for COVID-19, IEEE Rev. Biomed. Eng. 14 (2021) 4–15, http://dx.doi.org/10.1109/RBME.2020.2987975.

[25] J. Chen, K. Li, Z. Zhang, K. Li, P.S. Yu, A survey on applications of artificial intelligence in fighting against COVID-19, 2020, pp. 1–37, [Online]. Available: http://arxiv.org/abs/2007.02202.

[26] F. Rustam, et al., COVID-19 future forecasting using supervised machine learning models, IEEE Access 8 (2020) 101489–101499, http://dx.doi.org/10.1109/ACCESS.2020.2997311.

[27] C.M. Yeşilkanat, Spatio-temporal estimation of the daily cases of COVID-19 in worldwide using random forest machine learning algorithm, Chaos Solitons Fractals 140 (2020) 110210, http://dx.doi.org/10.1016/j.chaos.2020.110210.

[28] P. Arora, H. Kumar, B.K. Panigrahi, Prediction and analysis of COVID-19 positive cases using deep learning models: A descriptive case study of India, Chaos Solitons Fractals 139 (2020) 110017, http://dx.doi.org/10.1016/j.chaos.2020.110017.

[29] V.K.R. Chimmula, L. Zhang, Time series forecasting of COVID-19 transmission in Canada using LSTM networks, Chaos Solitons Fractals 135 (2020) 109864, http://dx.doi.org/10.1016/j.chaos.2020.109864.

[30] T. Tian, Y. Jiang, Y. Zhang, Z. Li, X. Wang, H. Zhang, COVID-Net: A deep learning based and interpretable predication model for the county-wise trajectories of COVID-19 in the United States, MedRxiv (2020) http://dx.doi.org/10.1101/2020.05.26.20113787, 2020.05.26.20113787.

[31] E. Chen, K. Lerman, E. Ferrara, Tracking social media discourse about the COVID-19 pandemic: Development of a public coronavirus Twitter data set, JMIR Public Health Surveill. 6 (2) (2020) http://dx.doi.org/10.2196/19273.

[32] Z. Zengin Alp, Ş. Gündüz Öğüdücü, Identifying topical influencers on twitter based on user behavior and network topology, Knowl.-Based Syst. 141 (2018) 211–221, http://dx.doi.org/10.1016/j.knosys.2017.11.021.

[33] M.O. Lwin, et al., Global sentiments surrounding the COVID-19 pandemic on Twitter: Analysis of Twitter trends, JMIR Public Health Surveill. 6 (2) (2020) e19447, http://dx.doi.org/10.2196/19447.

[34] H.W. Park, S. Park, M. Chong, Conversations and medical news frames on Twitter: Infodemiological study on COVID-19 in South Korea, J. Med. Internet Res. 22 (5) (2020) e18897, http://dx.doi.org/10.2196/18897.

[35] A.S.M. Venigalla, S. Chimalakonda, D. Vagavolu, Mood of India during Covid-19 - an interactive web portal based on emotion analysis of Twitter data, in: Conference Companion Publication of the 2020 on Computer Supported Cooperative Work and Social Computing, 2020, pp. 65–68, http://dx.doi.org/10.1145/3406865.3418567.

[36] N. Zheng, et al., Predicting COVID-19 in China using hybrid AI model, IEEE Trans. Cybern. 50 (7) (2020) 2891–2904, http://dx.doi.org/10.1109/TCYB.2020.2990162.

[37] B.B. Hazarika, D. Gupta, Modelling and forecasting of COVID-19 spread using wavelet-coupled random vector functional link networks, Appl. Soft Comput. 96 (2020) 106626, http://dx.doi.org/10.1016/j.asoc.2020.106626.

[38] F. Pedregosa, et al., Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (Null) (2011) 2825–2830.

[39] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, Distributed representations ofwords and phrases and their compositionality, Adv. Neural Inf. Process. Syst. (2013) 1–9.

[40] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: 1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc., 2013, pp. 1–12.

[41] L. Zhang, Y. Pan, X. Wu, M.J. Skibniewski, Artificial intelligence in construction engineering and management, Springer, ISBN: 978-981-16-2842-9, 2021, pp. 1–256.

[42] N. Kumar, S. Susan, COVID-19 pandemic prediction using time series forecasting models, in: 2020 11th Int. Conf. Comput. Commun. Netw. Technol. ICCCNT 2020, 2020, http://dx.doi.org/10.1109/ICCCNT49239.2020.9225319.

[43] V. Papastefanopoulos, P. Linardatos, S. Kotsiantis, COVID-19: A comparison of time series methods to forecast percentage of active cases per population, Appl. Sci. 10 (11) (2020) 1–15, http://dx.doi.org/10.3390/app10113880.

[44] M. Núñez, N.L. Barreiro, R.A. Barrio, C. Rackauckas, Forecasting virus outbreaks with social media data via neural ordinary differential equations, MedRxiv (2021) [Online]. Available: http://medrxiv.org/content/early/2021/01/31/2021.01.27.21250642.abstract, 2021.01.27.21250642.

[45] S. Yousefinaghani, R. Dara, S. Mubareka, S. Sharif, Prediction of COVID-19 waves using social media and google search: A case study of the US and Canada, Front. Public Health 9 (April) (2021) 1–11, http://dx.doi.org/10.3389/fpubh.2021.656635.

[46] F. Petropoulos, S. Makridakis, N. Stylianou, COVID-19: Forecasting confirmed cases and deaths with a simple time-series model, Int. J. Forecast. (2020) http://dx.doi.org/10.1016/j.ijforecast.2020.11.010, doi: 10.1016/j.ijforecast.2020.11.010.

[47] Y. Pan, Li Zhang, A BIM-data mining integrated digital twin framework for advanced project management, Autom. Constr. 124 (2021) 103564, http://dx.doi.org/10.1016/j.autcon.2021.103564.