Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 19 No. 14 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v19i14.41291

PAPER

A Boosted Evolutionary Neural Architecture Search for Time Series Forecasting with Application to South African COVID-19 Cases

Solomon Oluwole Akinola[1](✉), Qing-Guo Wang[1,2], Peter Olukanmi[1], Tshilidzi Marwala[1]

[1]Institute for Intelligent Systems, University of Johannesburg, Johannesburg, South Africa

[2]BNU-HKBU United International College, Zhuhai, China

oluwolea@uj.ac.za

ABSTRACT

In recent years, there has been an increase in studies on time-series forecasting for the future occurrence of disease incidents. Improvements in deep learning approaches offer techniques for modelling long-term temporal relationships. Nonetheless, this design practice is rigorously painstaking, prone to errors, and requires human expertise. The advent of feature enrichment with automatic architecture search typically optimises the discovery of new neural architectures applicable in domains such as time-series modelling. The main methodological contribution of this study is an approach for time-series forecasting using feature-enriched filters and an evolutionary neural architecture search with sequence-to-sequence gated recurrent units (GRU-Seq2Seq). This is applied to the prediction of daily cases of coronavirus disease in South Africa. The highly pathogenic coronavirus pandemic incident data was modelled with filters, optimised hyper-parameter search trials and an evolutional neural algorithm. The proposed model was benchmarked against ARIMA and SARIMA. The model predicted trends for 30, 60 and 90-day horizons and evaluated them for 7, 14 and 31 days. Simulation results demonstrate that observed daily case counts with added filters and evolutionary search optimisation for forecasting improve performance accuracy. Generally, the proposed bFilter+GRU-Seq2Seq with optimal search configuration outperformed ARIMA and SARIMA with lower error scores and higher performance metrics, with an R^2 score of 7.48E–01 for a 30-day forecast horizon.

KEYWORDS

time series, sequence-to-sequence, gated-recurrent-units, neural architecture search

1 INTRODUCTION

Deep learning architectures are developed by human experts, which is time-consuming and error-prone due to inadequate features from contributing variables. Feature enrichment with filters, such as the Hodrick-Prescott (HP) filter [1]

Akinola, S.O., Wang, Q.-G., Olukanmi, P., Marwala, T. (2023). A Boosted Evolutionary Neural Architecture Search for Time Series Forecasting with Application to South African COVID-19 Cases. *International Journal of Online and Biomedical Engineering (iJOE)*, 19(14), pp. 107–130. https://doi.org/10.3991/ijoe.v19i14.41291

Article submitted 2023-05-11. Revision uploaded 2023-06-17. Final acceptance 2023-06-25.

and the Christiano-Fitzgerald (CF) trend filter [2], focus on long-run time-series data. A boosted HP filter [3] elevates the HP filter [1] and is well suited for machine- and deep-learning data-agnostic processes. The boosted HP filter satisfactorily accommodates trends that are much more general to the unit root processes. Empirical experiments further ascertain robustness. Combining a boosted HP filter [3] with neural architecture search (NAS) can potentially motivate the discovery of new architectures. NAS has shown promising results in different domains, including sequence modelling [4]. This study's primary contribution is to provide an empirical and analytical simulation of a boosted HP filter with NAS in time series parameter optimisation. Precisely, the power of predictive models was explored with automated NAS, feature enrichment, and gated recurrent unit (GRU) architecture for the emerging daily case count of COVID-19 data in South Africa.

Infectious diseases such as COVID-19 [5–7] have shown rhythmic patterns relating to factors that allow future outbreak modelling. Prior knowledge of incidence time-series data supports model selection for intertwining relationships. The evolving COVID-19 daily case counts demonstrated robust time-series forecasting (TSF) predictability. Using the data-driven autoregressive integrated moving average (ARIMA) [8], SARIMA and hybrid filters with recurrent neural network GRU hybrid sequence-to-sequence (Seq2Seq), the future rise in daily case counts for COVID-19 in South Africa long-term and short-term spatial predictions was explored.

ARIMA is an established predictive model for TSF. Prior investigations implemented classical ARIMA and variations of hybrid ARIMA predictive models for various epidemiological analyses [9–12]. In more recent predictive models for the COVID-19 pandemic, ARIMA model investigations have been conducted [13–15]. ARIMA models are reliable for short-time forecasts [16], but the long TSF horizon does not fit well with ARIMA models [15] [17]. The seasonality component of the ARIMA model also describes the features of univariate seasonality and non-seasonal ARIMA components.

Deep learning algorithms can comprehensively employ TSF for epidemiological problems. A detailed review of machine learning approaches has been investigated [14], [18]. The COVID-19 pandemic generated interest in machine and deep-learning approaches for TSF. Zhang et al.'s [6] study focused on the performance improvement and interpretability of COVID-19 using a novel autoregressive (AR) and long-short-term memory neural networks (LSTM) hybrid model. In another recent investigation, Chakraborty et al. [7] investigated transfer learning with the gated recurrent units (GRU) model from an ensemble of four countries trained to predict cases of COVID-19 confirmed, deaths, and recovered. The model was fine-tuned on India's COVID-19 dataset. Mahajan et al. [19] investigated six forecasting techniques to predict new case counts for COVID-19 in ten Indian states. In [20], six deep learning approaches were investigated in two countries, Australia and Iran, to forecast new cases and deaths for short and long-time horizons. Ramchandani and Mostafavi [21] demonstrated DeepCOVIDNet for COVID-19 case counts in the US with the most influential characteristics for predicting cumulative infection. The investigation in [22] explored a spatiotemporal anomaly detection model for COVID-19 for early detection based on the reconstruction error. In [23], a deep learning long-short-term memory (LSTM) model forecasted two-week accrued daily case counts. The study in [24] featured convolutional neural networks (CNN) and temporal CNN (TCN) for COVID-19 daily case count forecasting in France.

Diverse experimental studies during and after the COVID-19 outbreak have inspired research interest in novel learning methodologies and research outputs. Bernátová et al. [25] study considered models for bridging distance learning during

critical periods such as COVID-19 lockdown using a pedagogical experiment for learning resources. In another investigation, Ali et al. [26] investigated the impact of in-hospital contact, primarily from a cloud-based Internet of Things (IoT) health monitoring system with biosensors that form multi-vital signs. The considerable impact was ameliorating healthcare costs through real-time monitoring with comparable results. Yet again, Kiflee et al. [27] investigated digital technology themes on COVID-19 in Malaysia and publication indexing counts and demonstrated huge interest in advancing research trends.

Injecting feature enrichment [28] into the training information can help improve the accuracy of COVID-19 case counts for forecasting future horizons. Feature enrichment with trend and cycle filters transforms the univariate TSF into a multivariate TSF. A stepwise [29] search for the ARIMA and SARIMA models and advanced tuning, optimisation, and NAS techniques on a hybrid model with two GRU networks in a Seq2Seq arrangement for TSF of COVID-19 daily case counts were applied. We executed pre-processing with normalisation, trend decomposition, and differencing in the ARIMA search. We investigated a hybrid network using NAS neural network intelligence (NNI) with a boosted filter [3] on a GRU-Seq2Seq architecture. With careful selection of parameters such as search space, trial, and tuner for the hybrid model, we obtained optimal model performance from the COVID-19 dataset cases in South Africa. We incorporated filters into the training and validation sets to harness the GRU-Seq2Seq potential for feature representation. One of the weaknesses of deep learning is that substantial data is required to extract hidden dependencies in the training set. Furthermore, we implemented pre-processing with a sliding window, minimax scaling, teacher forcing ratio, boosted HP cycle filter [1], [3], and CF trend filter [2]. We observed that the boosted cycle filter [3] and CF trend filter [2] were crucial in defining the top 5% of the hyper-parameter dependencies. We benchmarked the results of the best bFilter+GRU-Seq2Seq model with those of the ARIMA and SARIMA models.

The rest of the paper is organised as follows: Section II analyses the technique and compares it with baseline models, Section III provides the experimental results and discussion, and Section IV presents the conclusion and future works.

2 METHOD

This section discusses the approach with the implementation of ARIMA, SARIMA, GRU architecture, Seq2Seq, NAS, filters, normalisation, dynamic teacher forcing, sliding window and evaluation metrics in the study.

2.1 Dataset

We sourced our date for the experiment from Our-World-in-Data, a publicly available repository on COVID-19 daily case counts in South Africa [30]. South Africa's daily case counts from its inception on the February 7, 2020, to January 30, 2022 were analysed in our experiment. The South African daily case count provides aggregated cases from the nine provinces. The National Institute for Communicable Diseases (NICD) provides surveillance information to the public regarding COVID-19 in South Africa [31]. The peak daily number of new cases culminated in selected incidences with awareness of new COVID-19 strains and reduced lockdown measures.

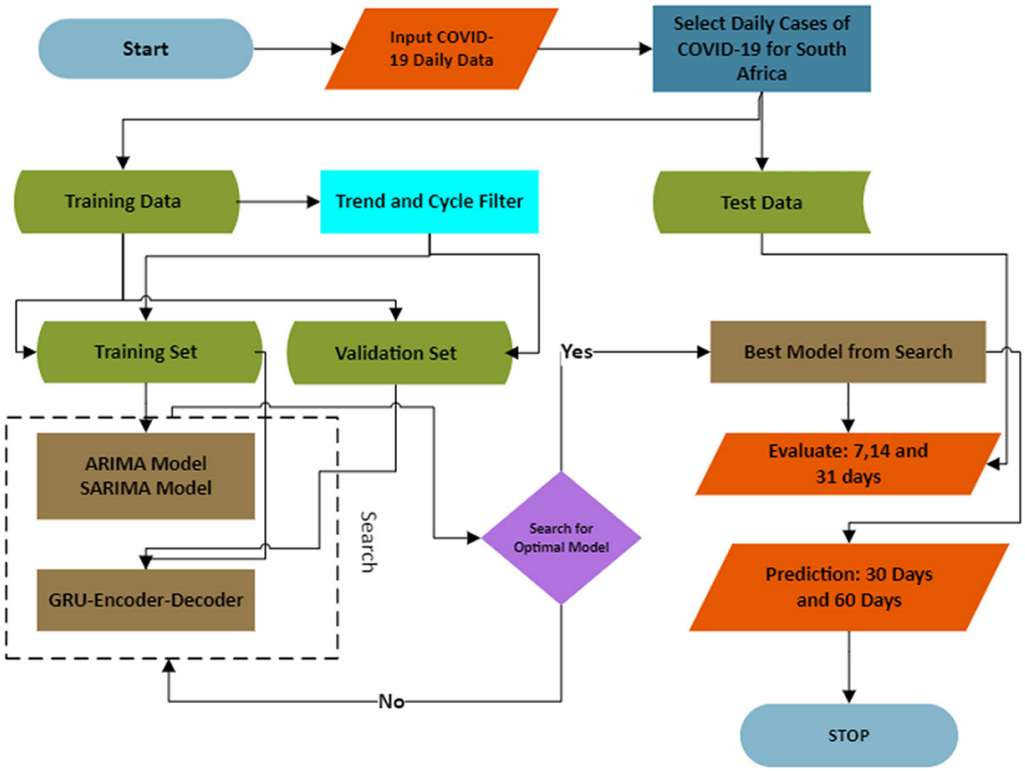

Fig. 1. Workflow of the COVID-19 models for ARIMA, SARIMA and bFilter+GRU-Seq2Seq (Encoder-Decoder)

South Africa is the fifth-most populous country in Africa, with a total area of 2,798 km on the South Atlantic and Indian Ocean coastlines. South Africa is bordered by Namibia, Botswana and Zimbabwe to the north, Mozambique and Eswatini to the east, and encircles the country of Lesotho. It has the second-largest economy in Africa as of 2021, and 28.3% of the population is younger than 15 years. South Africa is an ideal candidate as a test bed for COVID-19 case counts in Africa because it is considered one of the most prevalent COVID-19 hotspots for daily case counts as of December 2022 [32], high adult mortality due to the impact of HIV and AIDS, and declining life expectancy [33]. The African continent ranked lowest owing to the limited reporting of cases, testing, and vaccination [34]. However, the South African approach justifies a case study from Africa.

2.2 Workflow

A general workflow of COVID-19 daily case counts in South Africa TSF is presented in Figure 1. Input from Our-World-in-Data COVID-19 for South Africa case counts were split into training, validation and test sets. Null observations were converted to zero to regularise the missing values. A min-max normalisation was applied to rescale values between 0 and 1. Feature enrichment was incorporated with trend and cycle filters for short-term fluctuations resulting from COVID-19 strain outbreaks. A sliding window was applied to the transformation to prepare the time-series dataset for a supervised learning problem. The data was split into train, validate, and test sets. The test-train ratio was 4% for the test and 96% for the train set. 20% validation was performed on the train set. The training data comprised 694 observations, and the test data had 31 observations. The train data case counts were from inception on the 7th of February 2020 to the 31st of December 2021. The test data forecast is

from the 1st of January 2022 to the 30th of January 2022. The test forecast horizons were estimated with one-step-ahead forecasts for 31, 60, and 90 days and evaluation for 7 days, 14 days, and 31 days, respectively.

The ARIMA, SARIMA and GRU-Seq2Seq derivations and the proposed derivation follows. The neural architecture search algorithm resulted in the discovery optimal hyper parameter. Analysis of prediction, evaluation, and results, further justify the approach.

2.3 ARIMA and SARIMA

ARIMA combines AR and moving average (MA) with automatic differencing of non-stationary time series. In an AR model, the forecasting of a variable employs a linear combination of past values. An $AR(p)$ model is a multiple regression model with p lagged observations as predictors.

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t. \tag{1}$$

The $MA(q)$ model is a multiple regression with q lags as the forecast errors, and $MA(q)$ is expressed in (2) as

$$y_t = c + \varepsilon_t + \theta_1 + \cdots + \theta_q \varepsilon_{t-q}, \tag{2}$$

Where, ε_t is the white noise, y_t is the weighted moving average of the previous forecast error. Replacing the parameters ϕ_1, ..., ϕ_p and θ_1, ..., θ_q for orders p $AR(p)$ and $MA(q)$ led to distinct time-series patterns. An autoregressive moving average ARMA(p, q) model associates several regressions with p lags as observations and q lag errors as predictors. The insertion of differencing in the ARMA model results in the ARIMA(p, d, q) model with d level of differencing, as described in (3).

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t, \tag{3}$$

Box and Jenkin first proposed the ARIMA model [8], with norms for its application. Assumptions proposed are articulated in [35] as: (1) no anomalies in the timeseries dataset; (2) a univariate dataset; (3) stationary data (constant mean and variance) and (4) constant model parameters and error terms. Whereas ARIMA is applicable for TSF with non-seasonal components, seasonal ARIMA (SARIMA) is a time series with seasonal components $(P, D, Q)_m$. P, D and Q are seasonal components for $AR(P)$ order, $MA(Q)$ order and level of differencing D, respectively. The m parameter is the seasonality over time.

$$\Phi_p(B^m)\phi_p(B)(1-B^m)^D(1-B^m)^d y_t = \Theta_Q(B^m)\theta_q(B)_{w_t}, \tag{4}$$

$$B^k y_t = y_{t-k}, \tag{5}$$

In (4), stationary time-series, w_t represents the Gaussian white noise, $\varphi(B)$ is the non-seasonal AR, and $\theta(B)$ is the non-seasonal MA components. Achieving stationarity in the data with seasonal differencing D, requires an order of one or greater. The seasonal AR component is $\Phi_P(B^m)$, and the seasonal MA component is $\Theta_Q(B^m)$. In (5), B is the lag operator for shifting the y_t data back by the period k. In (6)–(9), the terms for the non-seasonal AR, non-seasonal MA, seasonal relationships for seasonal AR, and seasonal MA models are independently stated.

$$\varphi(B) = 1 - \varphi_1 B - \varphi_2 B^2 - \cdots - \varphi_p B^p, \tag{6}$$

$$\theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \cdots + \theta_q B^q, \tag{7}$$

$$\Phi_p(B^m) = 1 - \Phi_1 B^m - \Phi_2 B^{2m} + \cdots + \Phi_p B^{Pm}, \tag{8}$$

and

$$\Theta_Q(B^m) = 1 - \Theta_1 B^m - \Theta_2 B^{2m} + \cdots + \Theta_p B^{Qs}. \tag{9}$$

Expressing lag values using standard time-series metrics is essential for investigating the autocorrelation on multiple lag observations. The autocorrelation function (ACF), partial autocorrelation function (PACF), Akaike information criterion (AIC) and Bayesian information criterion (BIC) are widely applicable for these tasks. Time-series autocorrelation is the correlation of present observations with preceding lags. The ACF for time-series y_t in (10) determines the linear relationship between lags in a time-series dataset. The ACF check determines whether differencing is required or not. Visualising the time-series plot for autocorrelation vs. lags illustrates the direct correlation between observations at time t and observations at a previous time ($t - k$).

$$ACF(y_t, y_{t-k}) = \frac{cov(y_t, y_{t-k})}{var(y_t)}, \tag{10}$$

Here, k is the lag, which changes between y_t and observations, which are k periods separated, $cov(\cdot)$ is the covariance factor, and $var(\cdot)$ is the variance factor. On the other hand, the PACF in (11) for a given time step defines a partial correlation of the time-series with observations between points. The PACF considers the correlation between transitional observations while determining the correlation between two observations in different periods. For example, the time-series from lag 1, ..., k can be expressed as shown in (11). The PACF consists of observations for consecutive time steps. ACF and PACF can explain the choice of the best model parameters. The ACF indicates which autocorrelations to retain, and the PACF identifies an autoregressive model's order.

$$PACF(y_t, y_{t-k}) = \frac{cov(y_t, y_{t-k} | y_{t-1}, \ldots, y_{t-k})}{\sqrt{var(y_t | y_{t-1}, \ldots, y_{t-k+1})}\sqrt{var(y_{t-k} | y_{t-1}, \ldots, y_{t-k+1})}}, \tag{11}$$

The PACF demonstrates a coherent association with the lag and a trailing-off correlation from the lag onwards. There is no strict recommendation for interpreting the correlation coefficient because data collection in some disciplines is complex.

The AIC and BIC are popular criteria for assessing model quality using in-sample statistical measures. Refer to (12) and (13) for expressions. AIC and BIC incorporate penalised-likelihood criteria by crediting models with minor errors while applying penalties for models with more parameters. Lower residual errors of AIC and BIC are a good indication of the significance of the model. The goal is to decrease the complexity and increase the likelihood (goodness-of-fit) of the model. We mathematically illustrate AIC as:

$$AIC = -2logL(\hat{\theta}) + 2k, \tag{12}$$

Where, $logL(\hat{\theta})$ represents the likelihood function, L is the maximum likelihood, and K is the model parametrisation. Equally, in the BIC model, the criterion is imposed

with a reduced consequence on the parameters compared with AIC. In both AIC and BIC, a lower value best describes the times series data from many candidate models. We illustrate BIC in mathematical notation as:

$$BIC = -2logL(\hat{\theta}) + KlogN, \tag{13}$$

Where, N represents the number of observations.

2.4 GRU-RNN architecture

Inputs are fixed-length sequences in fully connected neural networks (FCNN). These peculiar behaviours fail when characterising sequential patterns as vectors of an ordered sequence. It is inferred that the FCNN is not suitable for TSF as it is not capable of a long-term relationship. The RNN architecture is a more effective approach [36]. The RNNs can preserve information over a sequence with a recurrent relation applied at every timestamp. RNNs can identify salient patterns in sequences and generate accurate forecasts. RNNs are effective in the TSF of weather [37], stock prices [38], electrocardiogram (ECG) [39], and deoxyribonucleic acid (DNA) sequencing [40]. As a result of computational complexity, RNN suffers from vanishing gradients [41] and exploding gradient problems [42]. The RNN also updates the hidden state after every iteration, resulting in difficulties recalling long-term dependencies. Two major RNN architectures address these difficulties with a memory unit for storing long-term information: GRU and long short-term memory (LSTM). To elaborate further, we will describe the formulation of the GRU architecture for the experiment. In (14)–(17), we describe the formulation for the GRU architecture as:

$$r_t = \sigma(x_t W_{ir} + b_{ir} + h_{t-1} W_{hr} + b_{hr}), \tag{14}$$

$$z_t = \sigma(x_t W_{iz} + b_{iz} + h_{t-1} W_{hz} + b_{hz}), \tag{15}$$

$$n_t = tanh(x_t W_{in} + b_{in} + r_t \odot (h_{t-1} W_{hn} + b_{hn})), \tag{16}$$

$$h_t = (1 - z_t) \odot n_t + z_t \odot h_{t-1}, \tag{17}$$

Where, hidden state h_t at time t, input x_t at time t, previous layer hidden state h_{t-1} at time $t-1$ also known as the initial hidden state at time 0, reset gate r_t, update gate z_t, and new gate n_t, at time t respectively. σ is the sigmoid function, and $\odot$ is the Hadamard product. The GRU cell improves on gaps of the RNN with the updated iteration limiting the hidden state to compensate for long-term information or dependencies. We depict the computational graph in the first level of Figure 2, as in (14). The reset gate r_t in time step t is the direct result of a sigmoid activation σ with summation of input x_t in time step t with its connection weight w_{ir} and the previously hidden state h_{t-1} in time step $t-1$ with the connection weight w_{hr}. Bias b_{ir} in time step t and bias b_{hr} in time step t are added to the input and hidden product prior to the activation function σ. The reset gate r_t is responsible for forgetting the unimportant part in the hidden state of GRU. The following variable is the update gate z_t in (14). The update gate is responsible for the proportion of the new state that represents the previous state. GRU selects long-term features r_t and forms a hidden state with long-term memory n_t and mixes it with the previously hidden state h_{t-1} with z_t ratio. In (15), the new gate n_t, permits or resets the previous values by providing

long-term information storage. The tanh activation function directly results from the output r_t with the Hadaman (elementwise) product of the previous hidden state added to the input. Incorporating the respective bias b_{in} and b_{hn} with weights W_{in} and W_{hn} multiplied by the input and previous hidden state, respectively. In (16), the new hidden state h_t is a simple linear combination of two hidden states: h_{t-1} previous hidden state and n_t – new gate (candidate hidden state) with long-term memory. The GRU simplifies the complexity of LSTM while preserving a high accuracy.

2.5 Sequence-to-sequence

The Seq2Seq [44] network is a two-layer RNN, GRU or LSTM network with the first layer (encoder) establishing typical input sequence characteristics of time step data and the second layer (decoder) mapping the hidden sequence from the encoder output as target time steps for the desired output. The architecture is closely related to [45] and has been widely implemented in mapping audio, sentence sequences [46], neural machine translation and TSF. A typical Seq2Seq architecture is illustrated in Figure 3. The hidden state h_n from the encoder inputs $x_1, x_2, ..., x_m$ returns a prediction sequence denoted as $y_1, y_2, ..., y_n$. The input and output states are recursive. The evaluation of the first input x_1 is performed using the recurrent input cell. The output from x_1 is the input of the recurrent hidden cell state h_1. The next phase combines the hidden state h_1 with the subsequent input x_2. This process is repeated until the last input x_m handles a hidden state h_m and ends the encoder layer. The h_m is the decoder layer recurrent cell input and generates the first output y_1. The y_1 value is fed recursively as input to the decoder layer recurrent cell to generate output $y_2, ..., y_n$. The equation assumes that each input to the decoder layer recurrent cell is a result of the preceding input.

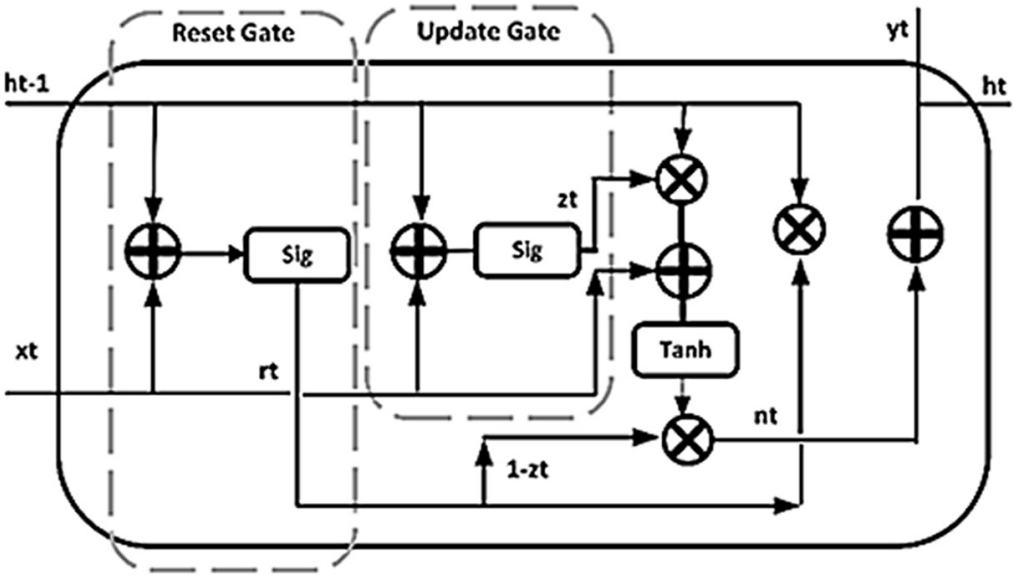

Fig. 2. The figure illustrates the schematic representation of the gated recurrent unit (GRU) cell with sigmoid and hyperbolic tangent activation functions [43]

2.6 Neural architecture search

The NAS is an evolutionary technique used for optimal model discovery [47]. Human experts developing deep learning models have successfully created model

architectures using a combination of heuristics and the requisite skills. Human experts who develop deep learning models require considerable time and effort. The process mimics the human experts' approach through an automatic network search for the best-performing model for task selection. This algorithm is simple and has a high level of accuracy. NAS is highly customizable with various open-source libraries. The application of NAS can significantly increase the variability and likelihood of obtaining highly useful architectures. A significant drawback is the duration of the training process. In some cases, NAS requires considerable computational resources and may not provide the best model from a trial session search. Although these drawbacks exist, NAS offers a robust algorithm for fast optimisation.

In [48] [49], a survey of the different techniques for NAS was reported. Our experiment focused on the NAS NNI toolkit [50]. The NNI is an open-source toolkit provided by Microsoft Inc. The hyper-parameters are: (1) number of hidden layers; (2) activation function; (3) dropout layer ratio; (4) number of training epochs; (5) optimisation; and (6) tuning learning rate for best model performance [51]. The NAS-NNI tuner aims to determine the top combination of hyper-parameters in a search space by reducing the number of trials. Each of the hyper-parameters can drastically affect the optimisation during the model search. Three key components constitute the hyper-parameter learning process: trial, search space and turner. A trial conducted training, validation and testing on a selected hyper-parameter trial metric. The search space combines all sets of hyper-parameters, and the tuner is a strategic model performance optimisation on a search space. There are several tuners, each with an algorithmic strategy [52]. We employed an evolutionary algorithm [47], known as the naïve evolutional or genetic algorithm tuner from the NAS-NNI toolkit, for search space reduction. The naïve evolutionary tuner applies intuitive mutation operators, traversing through large search spaces. The best combinations were selected for each generation, followed by a hyper-parameter mutation for the next generation in the search space. The naïve evolutional tuner has a simple development process with different features.

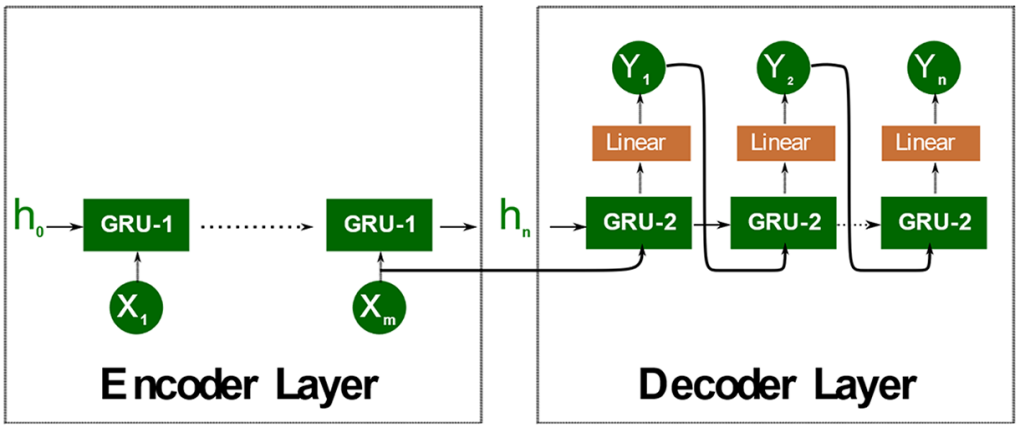

Fig. 3. Illustration of the schematic representation of the gated recurrent unit (GRU) cell for Seq2Seq implementation

2.7 Hodrick-Prescott, Christiano-Fitzgerald and booster filters

A range of filter smoothing techniques from HP [1], the band-pass technique developed by Christiano and Fitzgerald [2], and the boosting technique developed by Phillips and Shi [3]—trends and cycle filters—are features that are incorporated

as model input for experimentation. Hodrick and Prescott [1] proposed the HP filter technique for characterising time series as the sum of smoothly varying trends and cyclical components. The trend is a long time series in the upward or downward direction. The cyclical component represents the fluctuation around the trend variation for the selected periods. The HP filter has been widely applied to several other filters with commonly shared characteristics. CF filter [2] generalises the Baxter-King approximate band-pass filter.[1] The CF is a weighted moving average asymmetric for the period using the entire time series. The CF estimates the cyclical and trend components for a time series employing several band-pass estimate schemes. A study [53] criticised the HP filter for (1) producing series with false relations from undelaying data; (2) characterising different point values by false generation; (3) generating a smoothing parameter that lacks correlation with standard practice; and (4) an alternative technique with detrending. In [3], Philips and Shi proposed repeated analysis with an HP filter to generate a more innovative smoothing approach, called the boosted HP (bHP) filter. The bHP is an L2-loss with a boosting machine learning approach. The bHP filter recovers trend procedures from a function involving combined techniques, deterministic drifts, and structural breaks, which are widely applicable in standard modelling methodologies. In our experiment, we applied filters to discover the optimal model and to remove trends based on the nature of the COVID-19 dataset.

2.8 Normalisation

Normalisation is an effective process for transforming a dataset into a mutual measure. Normalisations avoid bias with dataset features and enhance the model learning. In our investigation, we applied min-max normalisation to the feature values. Min-max normalisation does not alter the original distribution of the data. In (18), $\hat{x}$ returns a value between 0 and 1 for x as a dataset feature value; $min(x)$ is the smallest value in the dataset, and $max(x)$ is the most significant value in the dataset.

$$\hat{x} = \frac{x - min(x)}{max(x) - min(x)}, \tag{18}$$

2.9 Dynamic teacher forcing

The teacher forcing ratio enhances learning by using the ground truth from the previous time steps as the input. This approach is a substitute for back propagation through time, which is widely applied in training RNNs. Various deep learning architectures use the teacher forcing ratio [54]. In our investigation, dynamic teacher forcing improved performance by aiding the decoder with the expected input value at a mixed ratio [51].

2.10 Sliding window

A sliding window uses previous time steps to predict subsequent periods [55] and is essential for preparing data for a deep learning approach. We applied the sliding window to pre-process the COVID-19 daily case counts for the TSF.

[1] https://www.statsmodels.org/dev/examples/notebooks/generated/tsa_filters.html

2.11 Evaluation metric

We employed multiple metric evaluations for COVID-19 daily case counts to address the difficulties inherent in evaluating model performance. These are root mean square error (RMSE), mean absolute percentage error (MAPE), mean absolute error (MAE) and coefficient of variance (R^2) score. The mathematical derivation for the error estimations is given in (19) to (21).

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}\frac{\left|y_t - \hat{y}_t\right|}{y_t}\%, \tag{19}$$

The *MAPE* in (19) is also known as the mean absolute percentage deviation (MAPD). The *MAPE* takes the absolute value from the forecast difference values $\hat{y}_t$ and the actual values, y_t, dividing by the actual values y_t, the percentage applied and averaging across the dataset.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_t - \hat{y}_t\right|, \tag{20}$$

In (20), the *MAE* is also known as the L1 loss. To estimate *MAE*, we take the absolute value of the difference between the forecast values $\hat{y}_t$ and the actual values, y_t. The average was obtained across the dataset. *MAE* is not sensitive to outliers when compared to the *RMSE*.

$$RMSE = \frac{1}{n}\sum_{i=1}^{n}\sqrt{(y_t - \hat{y}_t)^2}, \tag{21}$$

The *RMSE* in (21) is an estimate of L2 loss. The *RMSE* estimates the squared values between the forecast values $\hat{y}_t$ and the actual values, y_t. The root values are the estimates across the dataset. *RMSE* is most useful when significant errors exist and influence the model performance.

$$R^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(y_t - \hat{y}_t)^2}{(y_t - \bar{y}_t)^2}, \tag{22}$$

The R^2 score in (22) is a performance metric and not an error metric, as explained for the previous metrics in equations (19)–(21). R^2 estimates the ratio between the squared values from the difference in forecast values $\hat{y}_t$ and the actual values, y_t. The squared value between the difference of the mean values $\bar{y}_t$ and the actual values y_t and then the average of the sum, $\bar{y}_t$ is

$$\bar{y}_t = \frac{1}{n}\sum_{i=1}^{n} y_t. \tag{23}$$

3 RESULTS AND DISCUSSION

This section discusses exploratory analysis, forecast strategies and simulation with ARIMA, SARIMA and bFilter+GRU-Seq2Seq models using components, optimisation, prediction, and evaluation. The primary purpose of this investigation was to

determine forecast horizons for the COVID-19 daily cases in South Africa using evolutionary search optimisation and a feature-enriched boosted filter [3] with a GRU-Seq2Seq architecture. We demonstrate below that the resulting feature-enriched boosted filter with a GRU-Seq2Seq architecture is more accurate and termed the technique bFilter+GRU-Seq2Seq.

3.1 Exploratory data analysis

Figure 4 details the split for train and test sets, and Figure 5 shows the emerging cyclic trend and peak period as daily COVID-19 case counts progress in South Africa. We observed three significant peaks on the 24th of July 2020 (13.944K), the 8th of January 2021 (21.98K) and the 3rd of July 2021 (26.485K), respectively. After each peak period, there was a downward trend towards near-trivial daily confirmed cases. With the emergence of the Omicron strain, confirmed cases rose to an all-time high, signifying a shorter trough compared with previous trends. We detail the simulation for ARIMA, SARIMA, and bFilter+GRU-Seq2Seq in the subsequent subsections.

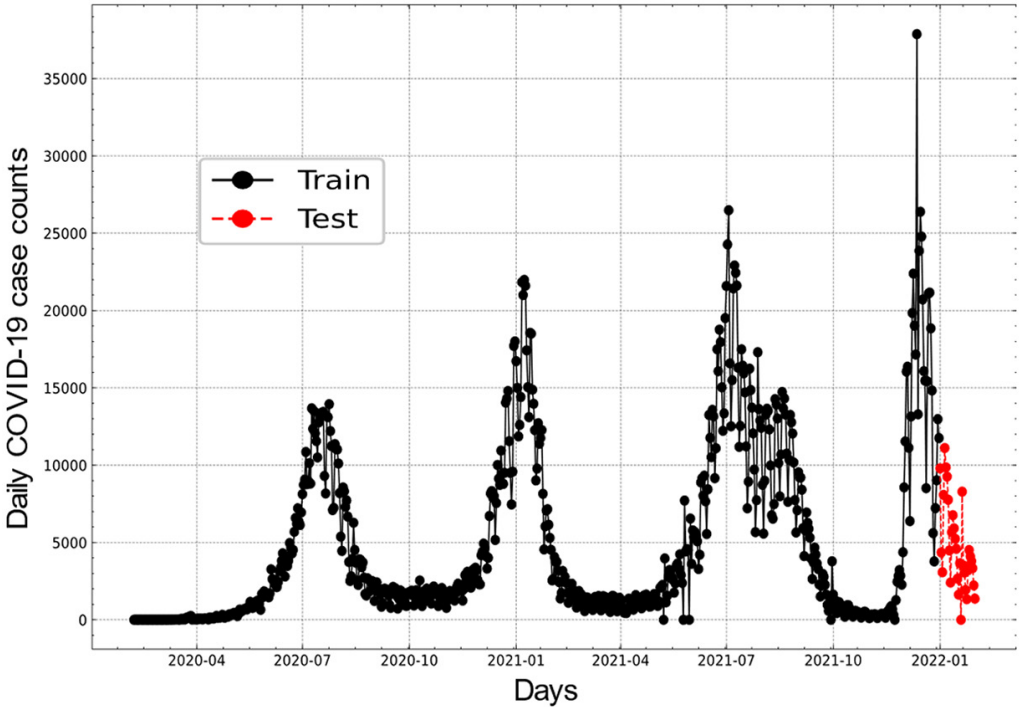

Fig. 4. Data distribution for training from the inception of case counts to the 31st of December 2021 and for the test set in the month of January 2022

3.2 ARIMA and SARIMA simulation

We developed models for ARIMA and SARIMA using the Python Statsmodel ARIMA library installed on a Windows 11 PC with 16 GB RAM and an Intel i7 9th generation processor. The average fit time for training the ARIMA and SARIMA models on the local PC were 67.659s and 230.873s, respectively. The parameters of $ARIMA_{(p,d,q)}$ and $SARIMA_{(p,d,q)(P,\ D,\ Q)m}$ were optimized using Python AUTO_ARIMA from the Pmdarima module and Statsmodel libraries. We investigate autocorrelation on

multiple lags as indicated by correlation plots in Figure 6a and 6b. In Figure 6a, the diagram shows the lag values on the x-axis and the correlation coefficient between –1 and 1 on the y-axis. The ACF line plot in Figure 6a shows that the first 22 lags are outside the shaded area when autocorrelation values are higher than 0.35. In Figure 6b, an obvious pattern is the decay from the PACF plot and not a clear pattern in an MA that would confirm a choice for the order of lags. There are lag values on the negative and positive sides of the PACF. ACF and PACF plots were set at a 95% confidence interval. We considered a more pragmatic approach because our goal was to select optimal lags for the best predictive model performance.

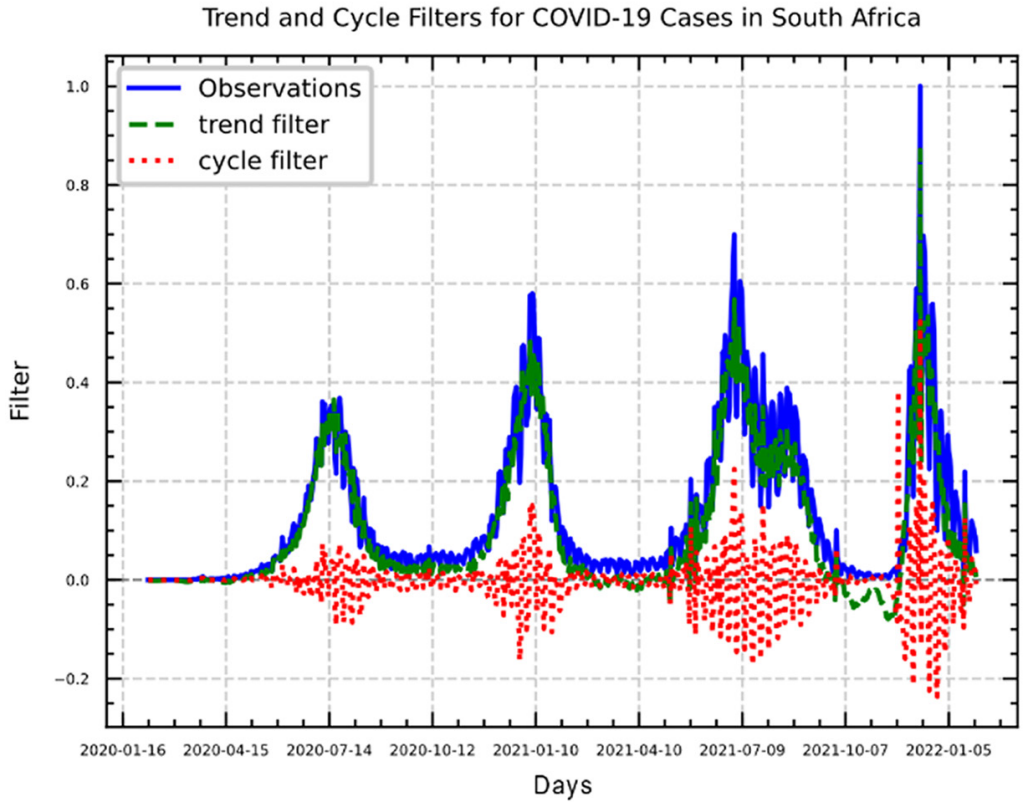

Fig. 5. The HP filter for trend and cyclical fluctuations from the original time-series of COVID-19 cases as indicated from the inception of COVID-19 in South Africa to the end of January 2022

We applied first-order differencing to the daily case counts of COVID-19 in South Africa. The Augmented Dickey-Fuller (ADF) test [56] and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test [57] for unit root tests were used for the time-series to determine the differencing, AR, or MA terms required to fit the model. We selected the best parameter from the AIC as the evaluation metric for the best ARIMA(p,d,q) and SARIMA(p,d,q)(P,D,Q)m models from the AUTO_ARIMA optimization. The ARIMA and SARIMA models were built using a stepwise configuration approach to detect the optimal combination by adjusting the seasonality feature to "false" for ARIMA models and "True" for the SARIMA model. The parameter search space was assigned the values listed in Table 1. Our data analysis showed that the best ARIMA and SARIMA model were identified using the AIC and BIC [58] information criteria. The $ARIMA_{(10,1,3)}$ implies an AR impact of order 10 and a MA effect of order 3, and a first-order differential is required. The evaluation in Table 1 lists the metrics for the values of $ARIMA_{(10,1,3)}$ and $SARIMA_{(0,1,1)(4,0,0)[7]}$ for the best COVID-19 case counts train data, respectively.

Figure 7a–c illustrate the visualized evaluation from the $ARIMA_{(10,1,3)}$ for 30, 14, and 7 days forecast horizons respectively. The train data were from the last 120 observations, starting the 3rd of September 2021 to the 31st of December 2021. The red separator line indicates the beginning of the test data observations from the 1st of January 2022 to the 30th of January 2022. The predicted result is the green line plot. In the $ARIMA_{(10,1,3)}$ model, the AIC and BIC were 1.23E+04 and 1.24E+04 respectively. The $ARIMA_{(10,1,3)}$ model evaluation errors for 30 days for RMSE, MSE, MAE, and MAPE are 3.13E+03, 2.73E+03, 2.73E+03, and 3.94E+17 respectively. The performance R^2 score was –2.38E–01. The $ARIMA_{(10,1,3)}$ model evaluation errors for 14 days for the RMSE, MSE, MAE, and MAPE were given as 3.53E+03, 1.25E+07, 3.05E+03, and 5.66E–01 respectively. The performance R^2 score was –8.53E–01. In the 7 days evaluation, RMSE, MSE, MAE, and MAPE are given as 3.56E+03, 1.27E+07, 3.04E+03, 4.56E–01 respectively. The performance R^2 score was –5.99E–01.

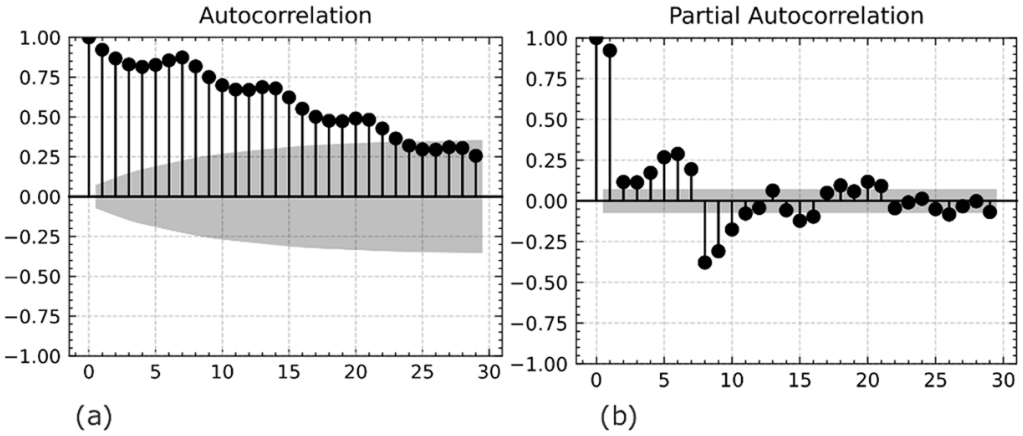

Fig. 6. In (a) Autocorrelation plot of daily COVID-19 case counts in South Africa ending on the 31st of December 2021 and in (b) Partial autocorrelation plot of daily COVID-19 case counts in South Africa ending on the 31st of December 2021

Table 1. ARIMA and SARIMA model configurations

Component	Model-Period	Min	Max
Non-Seasonality	AR(p)	0	10
	MA(q)		10
	I(d)		10
Seasonality	AR(P)		7
	MA(Q)		7
	I(D)		7
Periods	Daily	7	
	Monthly	12	
	Weekly	52	
	Quarterly	4	
	Annual	1	

Notes: Non-seasonal components for the ARIMA model with a starting value of 0 and maximum search at 10. The seasonal and periodic components for the SARIMA model with a starting value of 0 and maximum search at 7.

Figure 7e–f illustrate the visualized evaluation from the $SARIMA_{(0,1,1)(4,0,0)[7]}$ for 31, 14, and 7 days forecast horizons respectively. The training data were from the last 120 observations beginning the 3rd of September 2021 to the 31st of December 2021. The red separator line indicates the beginning of test data observations from the 1st of January 2022 to the 30th of January 2022. The predicted result is the green-line plot. In the $SARIMA_{(0,1,1)(4,0,0)[7]}$ model, AIC and BIC were 12326.537 and 12353.783 respectively. The $SARIMA_{(0,1,1)(4,0,0)[7]}$ best R^2 score was 5.85E–01 for 7 days evaluation horizon. The short-term forecast indicates better predictability than the long-term forecast.

3.3 bFilter+GRU-SEQ2SEQ simulation

We developed the bFilter+GRU-Seq2Seq model using the PyTorch package (version 1.10.1+cu113) on Python (version 3.9) programming language. The simulation was performed on a compute unified device architecture (CUDA) enabled computer. The NAS was tuned with Microsoft NNI [50] for the trend filter, cycle filter, number of hidden layer nodes, hidden layer size, learning rate and teacher forcing ratio (TFR). The statistical HP filter separates trend and cyclical fluctuations from the original time-series COVID-19 cases, as indicated in Figure 5. Trend and cycle filters were optimized using the default configuration for bHP, HP and CF filters, and no filter, respectively. The number of hidden nodes (8,12,16,24,32,64), hidden layer size (4,6,8,12), learning rate (.001,.005,.01) and teacher forcing rate (0.1, 0.2, 0.3, 0.4, 0.5) parameters were configured for sequential combinations to achieve the best prediction metric. Sequences in parenthesis are a range of values for the NAS optimization. NAS optimisation search creates suitable models for the COVID-19 daily case count task. The deep learning model proposed for South Africa was separated using a filter cycle, trend, and normalised historical data component. The number of features were three entries from the cycle the filter, trend filter, and normalised daily case counts. Given this feature representation, a multivariate input with a multi-step forecast emerged. For this task, a sliding window and random sample validation with a probability of 2% on the training set were applied to the model quality metric. The NAS optimisation durations were 1h, 11min, and 12s. The number of trials was 100 and 0 failed running four concurrent configurations. The training platform was on a local computer and the NAS tuner algorithm employed was the naïve evolution approach. The shortest duration for a model run was 1min, 23s, and the longest duration was 1min, 27s. The performance across the NAS-NNI trials was similar, as shown in Table 2. The model's best hyper-parameter was obtained from the NAS-NNI training set data from the 7th of February 2020 to the 31st of December 2021. The bHP cycle and CF trend filters showed the lowest metric error for the hyper-parameter configuration.

The COVID-19 new case counts hyper-parameters without trend or cycle filters were not ranked within the top 10 configurations. All the top metrics are close, and the RMSE ranges from 2.44E–03 and 5.87E–03. All learning rates were 0.01 except for the 10th ranked configuration with a learning rate of 0.005. The top 10 ranked combinations applied different TFRs ranging from 0.1–0.4. The complete top-ranked hyper-parameter combinations are presented in Table 2. In Table 2, the first, third, fifth and seventh-ranked model configurations applied TFR of 0.2. The second and fourth configurations applied a TFR of 0.4. The last 3 in the top ten ranked configurations applied a TFR of 0.3, and the seventh ranked with only a TFR of 0.1.

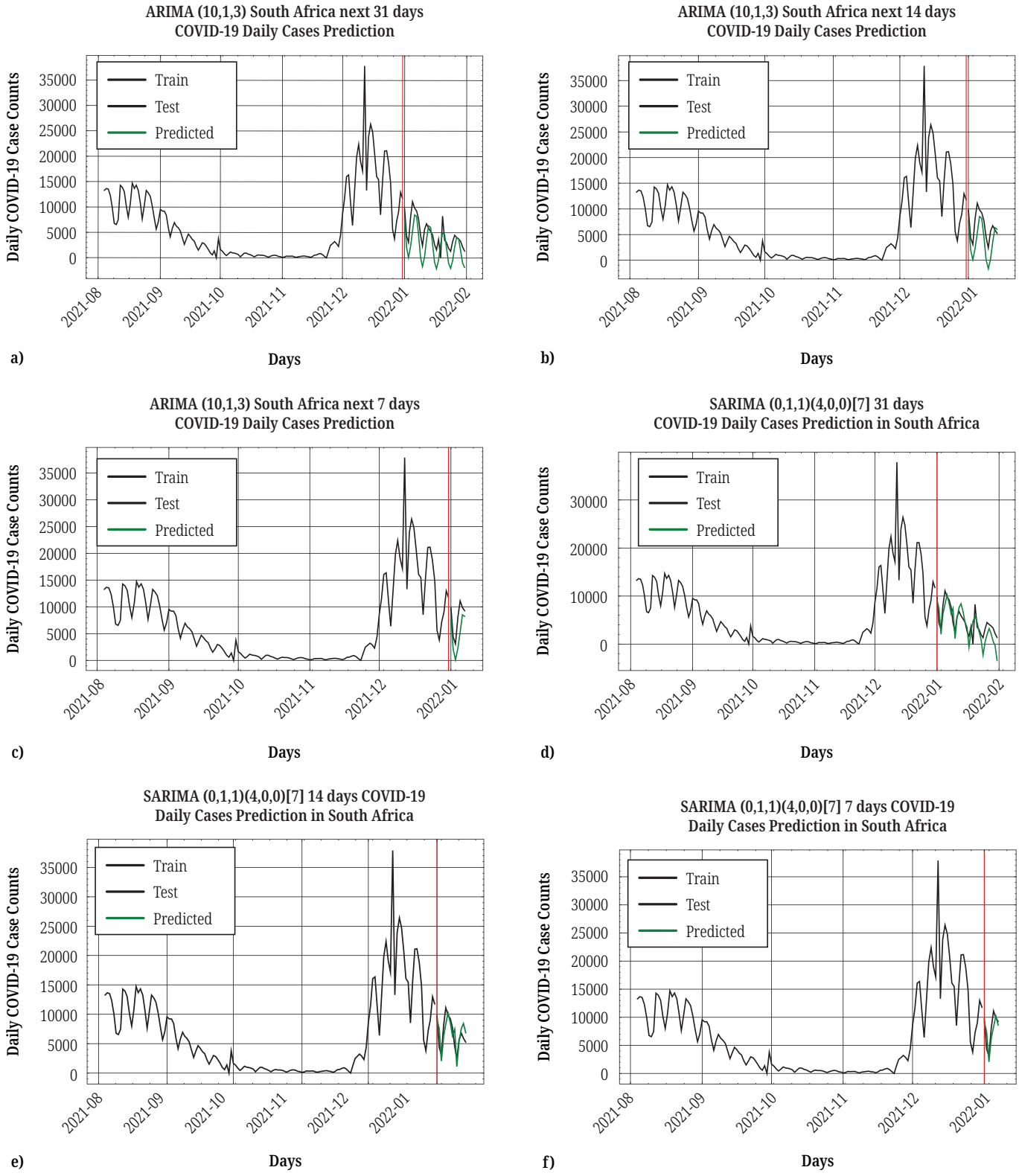

Fig. 7. ARIMA 31 days daily COVID-19 daily case counts: (a) 1st of January 2022 to the 31th of January 2022, (b) 1st of January 2022 to the 14th of January 2022, (c) 1st of January 2022 to the 7th of January 2022, SARIMA 31 days daily COVID-19 case counts from: (d) 1st of January 2022 to the 31th of January 2022, (e) 1st of January 2022s to the 14th of January 2022, and (f) 1st of January 2022 to the 7th of January 2022

The model was evaluated for 31, 14 and 7 days TSF (Table 3). Table 3 shows that the combined daily new cases for South Africa were trained with the hyper-parameter

configuration, the prediction and evaluation metrics were reported. In South Africa, the COVID-19 pandemic's first three waves were below 30,000 daily case counts until the fourth wave, culminating in the emergence of the Omicron strain with the highest daily case count of 37,875, and cumulative daily cases at 3,167,497 on the 12th of December 2021. The cumulative daily case counts were highest in Africa as of the end of 2021. The best RMSE performance for model valuation of the NAS configuration was 2.74E–03. The 31 days TSF horizon had lower RMSE, MSE, and MAE errors of 1.19E+03, 1.42E+06 and 9.85E+02, respectively, and the best R^2 score at 7.48E–01, followed by 14 days TSF horizon with the best MAPE score. Figure 8a–c illustrate the evaluation for 7, 14 and 31 days horizons.

Table 2. Top 10 hyper-parameter combinations from NAS result metric

Rank	Period	Metric	Trend	Cycle	Node	Layer	LR	TFR
1	**1m 23s**	**0.002444**	**CF**	**bHP**	**64**	**6**	**0.01**	**0.2**
2	1m 31s	0.002589	bHP	bHP	64	6	0.01	0.4
3	1m 25s	0.002703	CF	HP	32	8	0.01	0.2
4	1m 27s	0.002927	NONE	bHP	64	6	0.01	0.4
5	1m 27s	0.003622	HP	bHP	32	12	0.01	0.2
6	1m 23s	0.004463	NONE	HP	12	12	0.01	0.1
7	1m 24s	0.004739	HP	bHP	24	12	0.01	0.2
8	1m 24s	0.005346	CF	bHP	64	12	0.01	0.3
9	1m 25s	0.005629	CF	bHP	64	12	0.01	0.3
10	1m 26s	0.005873	NONE	NONE	64	6	0.005	0.3

Notes: The trend and cycle configurations are bHP, HP, and CF for booted Hodrick–Prescott, Hodrick–Prescott, and Christiano-Fitzgerald filters in that order. The top hidden layer node (NODE), hidden layer size (LAYER), learning rate (LR), and teacher forcing ratio (TFR) configurations are depicted.

Figure 8d and 8f provide details of the bFilter+GRU-Seq2Seq model forecast horizons for 30, 60 and 90 days, respectively. The bFilter+GRU-Seq2Seq model was evaluated at 31, 14 and 7 days. The training data were from the last 120 observations, from the 3rd of September 2021 to the 31st of December 2021. The red separator line indicates the beginning of the test data observations from the 1st of January 2022 to the 30th of January 2022. The predicted results are plotted as green lines. 31 days evaluation errors were lower on the test set with 1.19E+03, 1.42E+06, 9.85E+02 and 7.48E–01 for the RMSE, MSE, MAE and R^2 scores. Only on the 14th day result was the MAPE lower at 2.01E–01 when compared with 31 days MAPE and higher with 7.37E+00 and 4.92E+00 for MAE, respectively. The model prediction for 31 days horizon from the 1st of January 2022 to the 30th of January 2022 shows a decreasing number of cases and daily case counts not exceeding 3500. In the case of 60 days horizon ending the 1st of March 2022, the number of daily case counts continued to drop and did not exceed 2500 daily case counts. In the case of 90 days horizon ending on the 31st of March 2022, the daily number of new cases from the model reached 6000 daily case counts.

3.4 Comparative analysis

The investigation results identified the apparent justification for the bHP filter [3] with GRU-Seq2Seq resulting in improved accuracy and lower evaluation errors. Prior investigations have distinguished the importance of feature enrichment in

deep learning tasks [28]. The present paper determined the effect of incorporating feature enrichment with a bHP filter [3] and GRU-Seq2Seq for TSF task. The most prominent finding from the analysis is that bHP filters are essential for salient features in lagged observations in the training set. Contrary to some expectations in [53], this study supports the evidence from previous observations e.g., [28].

Furthermore, the bHP filter improved features in seven of the top 10 model configurations. Based on our experiment. We did not find a varied difference in cycle filters and learning rate configurations. However, the model configuration varies depending on the trend filter, the number of nodes, layers and TFR.

Our overall best model (bFilter+GRU-Seq2Seq model) was 28% higher in performance accuracy when compared to Ogundare and Van Zyl [45] GRU model for R^2 score, while Ogundare and Van Zyl [45] ARIMA model had lower errors on equivalent evaluation metrics than the $ARIMA_{(10,1,3)}$. The short-term forecast indicated better predictability than the long-term forecast, as expected when comparing the bFilter+GRU-Seq2Seq model to the SARIMA model. The SARIMA-based models are effective for short-term TSF, in line with previous studies [15] [34].

Table 3. Performance comparison of ARIMA, SARIMA and filters+GRU-Seq2Seq models for the test dataset

RMSE	*MSE*	*MAE*	*MAPE*	R^2_Score	Horizon	Model
3.13E+03	**2.73E+03**	2.73E+03	3.94E+17	–2.38E–01	31	ARIMA (10,1,3)
3.53E+03	1.25E+07	3.05E+03	5.66E–01	–8.53E–01	14	
3.56E+03	1.27E+07	3.04E+03	4.56E–01	–5.99E–01	7	
2.20E+03	4.83E+06	1.83E+03	6.90E+17	3.90E–01	31	SARIMA(0,1,1)(4,0,0)[7]
1.80E+03	3.22E+06	1.53E+03	2.86E–01	5.22E–01	14	
1.81E+03	3.29E+06	1.50E+03	2.44E–01	5.85E–01	7	
1.19E+03	1.42E+06	**9.85E+02**	**2.62E–01**	**7.48E–01**	31	Filter+GRU-Seq2Seq
1.35E+03	**1.82E+06**	**1.19E+03**	**2.01E–01**	**6.11E–01**	14	
1.70E+03	**2.89E+06**	**1.61E+03**	**2.40E–01**	**3.43E–01**	7	

Notes: 30 days, 14 days, and 7 days horizons (HO) for ARIMA (top 3 rows), SARIMA (mid 3 rows), and GRU-Seq2Seq (bottom 3 rows) Models result for COVID-19 daily new cases data in South Africa using RMSE MSE, MAE MAPE, and R^2 Score metrics.

In the LSTM model proposed by Elsheikh et al. [48], our best R^2 score was 24% lower in comparable metrics. In the interpretable hybrid AR-LSTM model [6], our MAPE is considered reasonable when compared. Similarly, Chakraborty et al. [7] Pretrained a transfer weighted ensemble with GRU in the daily confirmed cases and demonstrated improvement in RMSE results. A possible explanation for these results may be the number and period of observations, feature engineering, training observations, statistical, machine learning or deep learning architecture deployed for the model approach. Our findings were restricted to COVID-19 daily case counts in South Africa. However, such associations are likely in countries with similar COVID-19 observations and daily case counts. Again, it is possible to hypothesise that similar results with our proposed configuration are generalisable.

The present investigation raises the possibility that feature enrichment with bHP filters and GRU-Seq2Seq supports the conceptual premise of TSF. The current results are significant in at least two major aspects: (1) feature enrichment in TSF tasks offers robust models for TSF, and (2) filters are relevant in extrapolating salient features for TSF tasks. Further work is required to establish the viability of feature enrichment with filters and deep learning architectures in relevant domains with TSF tasks.

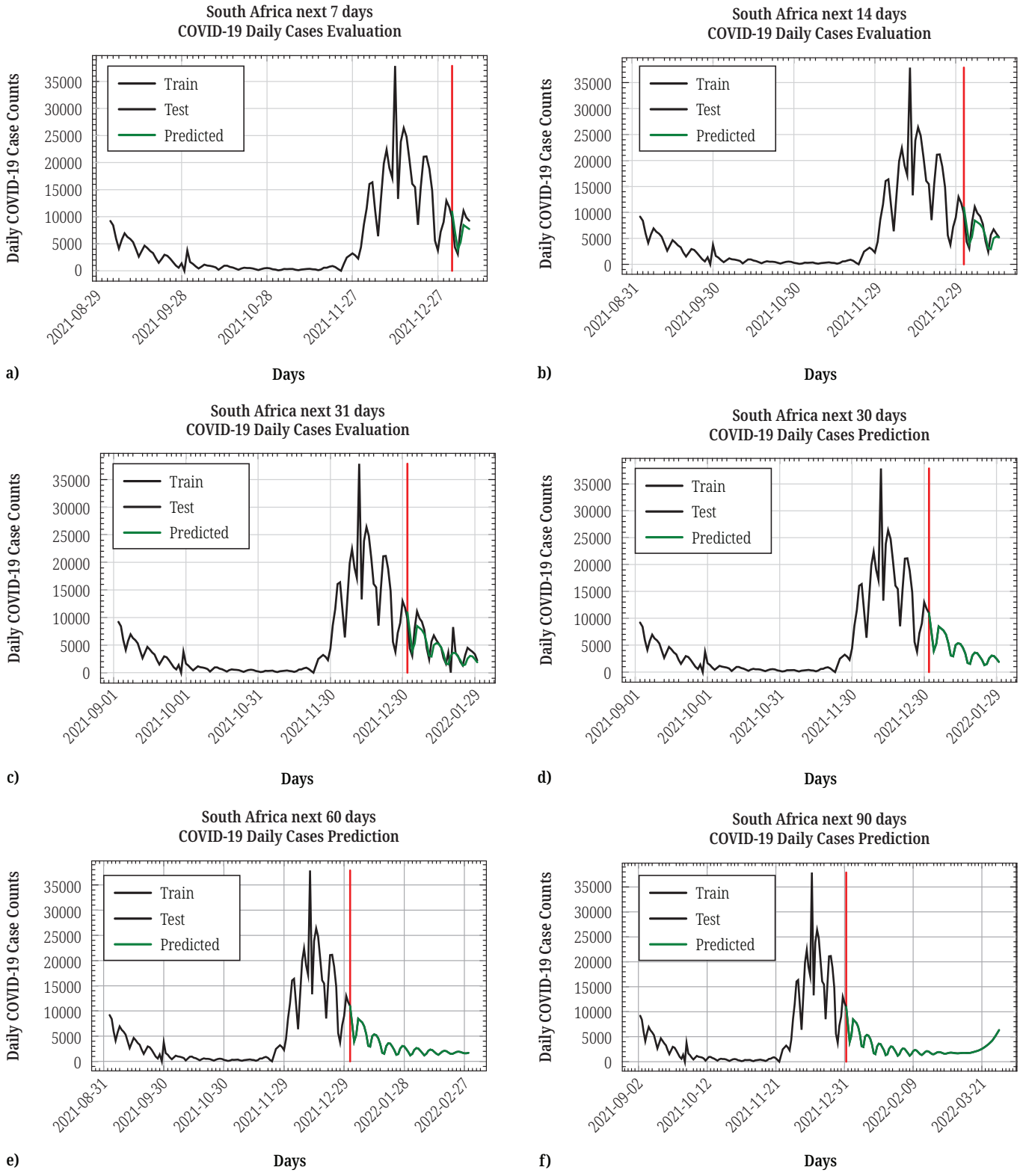

Fig. 8. bFilter+GRU-Seq2Seq evaluation plot of COVID-19 daily case counts in South Africa: (a) 1st of January 2022 to the 7th of January 2022, (b) 1st of January 2022 to the 14th of January 2022, (c) 1st of January 2022 to the 31st of March 2022, GRU-Seq2Seq prediction plot of COVID-19 daily case counts in South Africa from (d) 1st of January 2022 to the 30th of January 2022, (e) 1st of January 2022 to the 1st of March 2022, and (f) 1st of January 2022 to the 31st of March 2022

4 CONCLUSION

We have presented a time-series forecasting approach which combines feature-enriched filters and evolutionary neural architecture search with bFilter+GRU-Seq2Seq. Our approach is applied for predicting daily case counts of COVID-19 disease in South Africa. Experiment shows that our proposed technique bFilter+improved on the traditional COVID-19 South Africa cases. We benchmarked ARIMA and SARIMA against the bFilters+GRU-Seq2Seq model for daily cases of COVID-19 TSF in South Africa. Using historical data from the Our-World-in-Data. COVID-19 daily case counts were trained from inception to the 31st of December 2021. Experimental results predict a steady decrease in daily case counts within 60 days. Daily case counts will rise as we approach the 90 days prediction horizon. A comparative practical result with ARIMA and SARIMA shows sufficient predictive accuracy in evaluating short-term forecasts compared with the bFilter+GRU-Seq2Seq forecast. bFilter+GRU-Seq2Seq demonstrates sufficient predictive accuracy over the ARIMA and SARIMA models in long-term forecasting. The future goal is to leverage the approach for African countries to model daily case counts as indicators. Again, surge in publications during and after the COVID-19 pandemic is testament to huge interest in advancing active research trends in multidisciplinary studies [25–27] and lessons from the pandemic's impact.

5 ACKNOWLEDGMENT

You may mention here granted financial support or acknowledge the help you got from others during your research work. Simply delete this section if it doesn't apply.

6 REFERENCES

[1] R. J. Hodrick and E. C. Prescott, "Postwar US business cycles: An empirical investigation," *J Money Credit Bank*, vol. 29, no. 1, pp. 1–16, 1997. https://doi.org/10.2307/2953682

[2] L. J. Christiano and T. J. Fitzgerald, "The band pass filter," *Int. Econ. Rev. (Philadelphia)*, vol. 44, no. 2, pp. 435–465, 2003. https://doi.org/10.1111/1468-2354.t01-1-00076

[3] P. C. B. B. Phillips and Z. Shi, "Boosting: Why you can use the HP filter," *Int. Econ. Rev. (Philadelphia)*, vol. 62, no. 2, pp. 521–570, 2020. https://doi.org/10.1111/iere.12495

[4] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8697–8710. https://doi.org/10.1109/CVPR.2018.00907

[5] W. Ding, Q.-G. Wang, and J.-X. Zhang, "Analysis and prediction of COVID-19 epidemic in South Africa," *ISA Trans.*, vol. 124, pp. 182–190, 2022. https://doi.org/10.1016/j.isatra.2021.01.050

[6] Y. Zhang, S. Tang, and G. Yu, "An interpretable hybrid predictive model of COVID-19 cases using autoregressive model and LSTM," *Sci. Rep.*, vol. 13, no. 1, p. 6708, 2023. https://doi.org/10.1038/s41598-023-33685-z

[7] D. Chakraborty, D. Goswami, S. Ghosh, A. Ghosh, J. H. Chan, and L. Wang, "Transfer-recursive-ensemble learning for multi-day COVID-19 prediction in India using recurrent neural networks," *Sci. Rep.*, vol. 13, no. 1, p. 6795, 2023. https://doi.org/10.1038/s41598-023-31737-y

[8] G. E. Box and G. M. Jenkins, *Time Series Analysis: Forecasting and Control,* Revised ed., San Francisco: Holden-Day, 1976.

[9] A. Earnest, M. I. Chen, D. Ng, and Y. S. Leo, "Using autoregressive integrated moving average (ARIMA) models to predict and monitor the number of beds occupied during a SARS outbreak in a tertiary hospital in Singapore," *BMC Health Serv. Res.*, vol. 5, no. 36, 2005. https://doi.org/10.1186/1472-6963-5-36

[10] M. Ture and I. Kurt, "Comparison of four different time series methods to forecast hepatitis A virus infection," *Expert Syst. Appl.*, vol. 31, no. 1, pp. 41–46, 2006. https://doi.org/10.1016/j.eswa.2005.09.002

[11] H. Li and Q. Tang, "Analyzing mortality bond indexes via hierarchical forecast reconciliation," *ASTIN Bulletin*, vol. 49, no. 3, pp. 823–846, 2019. https://doi.org/10.1017/asb.2019.19

[12] B. Xu, J. Li, and M. Wang, "Epidemiological and time series analysis on the incidence and death of AIDS and HIV in China," *BMC Public Health*, vol. 20, no. 1, p. 1906, 2020. https://doi.org/10.1186/s12889-020-09977-8

[13] A. K. Sahai, N. Rath, V. Sood, and M. P. Singh, "ARIMA modelling & forecasting of COVID-19 in top five affected countries," *Diabetes and Metabolic Syndrome: Clinical Research and Reviews*, vol. 14, no. 5, pp. 1419–1427, 2020. https://doi.org/10.1016/j.dsx.2020.07.042

[14] F. N. Khan, A. A. Khanam, A. Ramlal, and S. Ahmad, "A review on predictive systems and data models for COVID-19," in *Studies in Computational Intelligence*, K. Raza, Ed., vol. 923. Singapore: Springer Singapore, 2021, pp. 123–164. https://doi.org/10.1007/978-981-15-8534-0_7

[15] R. R. Sharma, M. Kumar, S. Maheshwari, and K. P. Ray, "EVDHM-ARIMA-Based time series forecasting model and its application for COVID-19 Cases," *IEEE Trans. Instrum. Meas.*, vol. 70, 2021. https://doi.org/10.1109/TIM.2020.3041833

[16] R. H. Shumway and D. S. Stoffer, *Time Series Analysis and Its Applications.* New York: Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-52452-8

[17] A. Barthwal and S. Avikal, "Time series analysis of COVID-19 confirmed cases in select countries," in *Recent Advances in Time Series Forecasting*, CRC Press, 2021, pp. 37–50. https://doi.org/10.1201/9781003102281-3

[18] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: A review," *Data Min Knowl Discov*, vol. 33, no. 4, pp. 917–963, 2019. https://doi.org/10.1007/s10618-019-00619-1

[19] V. Mahajan, R. Shah, D. Bhatt, D. Patel, P. Agrawal, and L. Tak, "Comparative study of time series forecasting models for COVID-19 cases in India," in *Recent Advances in Time Series Forecasting*, CRC Press, 2021, pp. 111–138. https://doi.org/10.1201/9781003102281-8

[20] N. Ayoobi *et al.*, "Time series forecasting of new cases and new deaths rate for COVID-19 using deep learning methods," *Results Phys*, vol. 27, p. 104495, 2021. https://doi.org/10.1016/j.rinp.2021.104495

[21] A. Ramchandani, C. Fan, and A. Mostafavi, "DeepCOVIDNet: An interpretable deep learning model for predictive surveillance of COVID-19 Using heterogeneous features and their interactions," *IEEE Access*, vol. 8, pp. 159915–159930, 2020. https://doi.org/10.1109/ACCESS.2020.3019989

[22] Y. Karadayi, M. N. Aydin, and A. S. Öğrenci, "Unsupervised anomaly detection in multivariate spatio-temporal data using deep learning: Early detection of covid-19 outbreak in Italy," *IEEE Access*, vol. 8, pp. 164155–164177, 2020. https://doi.org/10.1109/ACCESS.2020.3022366

[23] M. D. Hssayeni *et al.*, "The forecast of COVID-19 spread risk at the county level," *J. Big Data*, vol. 8, no. 1, pp. 1–16, 2021. https://doi.org/10.1186/s40537-021-00491-1

[24] L. Mohimont, A. Chemchem, F. Alin, M. Krajecki, and L. A. Steffenel, "Convolutional neural networks and temporal CNNs for COVID-19 forecasting in France," *Applied Intelligence*, vol. 51, pp. 8784–8809, 2021. https://doi.org/10.1007/s10489-021-02359-6

[25] R. Bernátová, M. Bernát, J. Poráčová, L. Rudolf, and A. Kľučarová, "On the teaching of natural science- and electrotechnologically-oriented university course for future teachers in the period of Covid 19 pandemic," *International Journal of Emerging Technologies in Learning (iJET)*, vol. 18, no. 8, pp. 97–117, 2023. https://doi.org/10.3991/ijet.v18i08.37365

[26] F. I. Ali, T. E. Ali, and Z. T. Al_dahan, "Private backend server software-based telehealthcare tracking and monitoring system," *International Journal of Online and Biomedical Engineering (iJOE)*, vol. 19, no. 1, pp. 119–134, 2023. https://doi.org/10.3991/ijoe.v19i01.32433

[27] A. K. R. Kiflee, N. N. Hasbullah, S. N. Ahmad, and S. H. Mastor, "Research progress of digital technology and emerging themes during Covid 19 in Malaysia: Bibliometric analysis approach," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 6, pp. 98–115, 2023. https://doi.org/10.3991/ijim.v17i06.35551

[28] Q. Xie, S. Tu, G. Wang, Y. Lian, and L. Xu, "Feature enrichment based convolutional neural network for heartbeat classification from electrocardiogram," *IEEE Access*, vol. 7, pp. 153751–153760, 2019. https://doi.org/10.1109/ACCESS.2019.2948857

[29] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *J. Stat. Softw*, vol. 27, no. 3, p. 22, 2008. https://doi.org/10.18637/jss.v027.i03

[30] Edouard Mathieu *et al.*, "Coronavirus pandemic (COVID-19)," *Our World in Data*, 2020. [Online]. Available: https://ourworldindata.org/coronavirus. [Accessed Jun. 5, 2023].

[31] SACOVID-19 portal, "COVID-19 South African coronavirus news and information," *COVID-19 Corona Virus South African Resource Portal*, 2021. https://sacoronavirus.co.za/. [Accessed May 23, 2022].

[32] H. Tegally *et al.*, "Sixteen novel lineages of SARS-CoV-2 in South Africa," *Nat. Med.*, vol. 27, no. 3, pp. 440–446, 2021. https://doi.org/10.1038/s41591-021-01255-3

[33] Statistics South Africa, "COVID-19 epidemic reduces life expectancy in 2021 | Statistics South Africa." https://www.statssa.gov.za/?p=14519. [Accessed Jul. 5, 2022].

[34] P. Cihan, "Forecasting fully vaccinated people against COVID-19 and examining future vaccination rate for herd immunity in the US, Asia, Europe, Africa, South America, and the World," *Appl. Soft Comput.*, vol. 111, p. 107708, 2021. https://doi.org/10.1016/j.asoc.2021.107708

[35] K. E. Arunkumar, D. V. Kalaga, C. Mohan, S. Kumar, and G. Chilkoor, "Forecasting the dynamics of cumulative COVID-19 cases (confirmed, recovered and deaths) for top-16 countries using statistical machine learning models: Auto-Regressive Integrated Moving Average (ARIMA) and Seasonal Auto-Regressive Integrated Moving Average (SARIMA)," *Applied Soft Computing Journal*, vol. 103, p. 107161, 2021. https://doi.org/10.1016/j.asoc.2021.107161

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. http://dx.doi.org/10.1162/neco.1997.9.8.1735

[37] S. Narejo and E. Pasero, "A hybrid approach for time series forecasting using deep learning and nonlinear autoregressive neural network," in *INTELLI 2016: The Fifth International Conference on Intelligent Systems and Applications (includes InManEnt 2016)*, 2016, pp. 69–75.

[38] R. Goebel, Y. Tanaka, and W. Wahlster, "PRICAI 2014: Trends in artificial intelligence," in *13th Pacific Rim International Conference on Artificial Intelligence*, D.-N. Pham and S.-B. Park, Eds., Springer, 2014. https://doi.org/10.978-3-319-13560-1

[39] F. Zhu, F. Ye, Y. Fu, Q. Liu, and B. Shen, "Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, 2019. https://doi.org/10.1038/s41598-019-42516-z

[40] J. X. Zhang *et al.*, "A deep learning model for predicting next-generation sequencing depth from DNA sequence," *Nat. Commun.*, vol. 12, no. 1, pp. 1–10, 2021. https://doi.org/10.1038/s41467-021-24497-8

[41] S. O. Olukanmi, F. V. Nelwamondo, and N. I. Nwulu, "Utilizing google search data with deep learning, machine learning and time series modeling to forecast influenza-like illnesses in South Africa," *IEEE Access*, vol. 9, pp. 126822–126836, 2021. https://doi.org/10.1109/ACCESS.2021.3110972

[42] S. Shastri, K. Singh, M. Deswal, S. Kumar, and V. Mansotra, "CoBiD-net: A tailored deep learning ensemble model for time series forecasting of covid-19," *Spatial Information Research*, vol. 30, pp. 9–22, 2022. https://doi.org/10.1007/s41324-021-00408-3

[43] K. E. ArunKumar, D. V. Kalaga, Ch. Mohan Sai Kumar, M. Kawaji, and T. M. Brenza, "Comparative analysis of Gated Recurrent Units (GRU), long short-term memory (LSTM) cells, autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA) for forecasting COVID-19 trends," *Alexandria Engineering Journal*, vol. 61, no. 10, pp. 7585–7603, 2022. https://doi.org/10.1016/j.aej.2022.01.011

[44] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," *Advances in Neural Information Processing Systems (NIPS),* pp. 3104–3112, 2014. https://doi.org/10.1007/s10107-014-0839-0

[45] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing,* October 2013, pp. 1700–1709.

[46] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179

[47] E. Real *et al.,* "Large-scale evolution of image classifiers," in *34th International Conference on Machine Learning (ICML),* 2017, pp. 4429–4446.

[48] T. Elsken, J. H. Metzen, and F. Hutter, "Correction to: Neural architecture search," in *Automated Machine Learning,* Hutter, F., Kotthoff, L., Vanschoren, J., Eds., The Springer Series on Challenges in Machine Learning. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-05318-5_11

[49] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen, and K. C. Tan, "A survey on evolutionary neural architecture search," *IEEE Trans Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 550–570, 2023. https://doi.org/10.1109/TNNLS.2021.3100554

[50] C. Ying, A. Klein, E. Real, E. Christiansen, K. Murphy, and F. Hutter, "NAS-BENCH-101: Towards reproducible neural architecture search," in *36th International Conference on Machine Learning (ICML)*, 2019, pp. 12334–12348.

[51] I. Gridin, *Time Series Forecasting using Deep Learning: Combining PyTorch, RNN, TCN, and Deep Neural Network Models to Provide Production-Ready Prediction Solutions.* Noida, India: BPB Publications, 2021.

[52] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nat. Mach. Intell.*, vol. 1, pp. 24–35, 2019. https://doi.org/10.1038/s42256-018-0006-z

[53] J. D. Hamilton, "Why you should never use the Hodrick-Prescott," *Review of Economics and Statistics*, vol. 100, no. 5, pp. 831–843, 2018. https://doi.org/10.1162/rest_a_00706

[54] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989. https://doi.org/10.1162/neco.1989.1.2.270

[55] J. Ortiz Laguna, A. G. Olaya, and D. Borrajo, "A dynamic sliding window approach for activity recognition," in *International Conference on User Modeling, Adaptation, and Personalization,* 2011, pp. 219–230. https://doi.org/10.1007/978-3-642-22362-4_19

[56] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root," *J. Am. Stat. Assoc.*, vol. 74, no. 366, p. 427, 1979. https://doi.org/10.2307/2286348

[57] Y. Shin and P. Schmidt, "The KPSS stationarity test as a unit root test," *Economics Letters*, vol. 38, no. 4, pp. 387–392, 1992. https://doi.org/10.1016/0165-1765(92)90023-R

[58] G. Schwarz, "Estimating the dimension of a model," *The Annals of Statistics,* vol. 6, no. 2, pp. 461–464, 1978. https://doi.org/10.1214/aos/1176344136

7 AUTHORS

Solomon Oluwole Akinola received a Bachelor of Computer Engineering degree from the Ladoke Akintola University of Technology. He received his master's degree from the University of Ibadan. He is currently pursuing his PhD at the University of Johannesburg. His research interests include applying machine (deep) learning to natural language processing, intelligent user interfaces, and the Internet of Things (E-mail: oluwolea@uj.ac.za).

Qing-Guo Wang received a B.Eng. in Chemical Engineering in 1982, M. Eng. in 1984 and a PhD in 1987, both in Industrial Automation, all from Zhejiang University, PR China. He held the Alexander-von-Humboldt Research Fellowship in Germany from 1990 to 1992. From 1992 to 2015, he was with the Department of Electrical and Computer Engineering at the National University of Singapore, where he became a full-time professor in 2004. He is a Distinguished Professor at the Institute for Intelligent Systems, University of Johannesburg, South Africa. He holds an A rating from the National Research Foundation of South Africa (NRF). He is a member of the Academy of Sciences in South Africa. His current research interests are mainly in modelling, estimating, predicting, controlling, optimising, and automating complex systems, including but not limited to industrial and environmental processes, new energy devices, defence systems, medical engineering, and financial markets. He has published nearly 300 international journal papers and 7 research monographs. He received approximately 15000 citations with an h-index of 65. He is currently the Deputy Editor-in-Chief of the ISA Transactions (USA) (E-mail: wangq@uj.ac.za).

Peter Olukanmi obtained a PhD from the University of Johannesburg and an MSc in Computer Science from the University of KwaZulu-Natal (UKZN) BSc in Systems Engineering from the University of Lagos. He won two IEEE conference awards in soft computing and machine intelligence. His research interests include fundamental and applied data science and mathematical modelling (E-mail: polukanmi@uj.ac.za).

Tshilidzi Marwala is a South African mechanical engineer and a computer scientist. He became a Professor at the University of the Witwatersrand in 2003 and a chairperson of System and Control Engineering in South Africa. He had previously worked at the CSIR and South African Breweries. Marwala's research interests include the theory and application of artificial intelligence in engineering, computer science, finance, economics, social science, and medicine. Marwala has made fundamental contributions to engineering science, including developing the concept of pseudo-modal energies and proposing the theory of rational counterfactual thinking, reasonable opportunity cost, and flexibly bounded rationality (E-mail: tmarwala@uj.ac.za).