

Article

# The Forecasting of a Leading Country's Government Expenditure Using a Recurrent Neural Network with a Gated Recurrent Unit

Cheng-Hong Yang <sup>1,2,3,4,5</sup>, Tshimologo Molefyane <sup>2</sup>  and Yu-Da Lin <sup>6,\*</sup> 

<sup>1</sup> Department of Information Management, Tainan University of Technology, Tainan 710302, Taiwan; chyang@nkust.edu.tw

<sup>2</sup> Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan; f109152190@nkust.edu.tw

<sup>3</sup> Ph.D. Program in Biomedical Engineering, Kaohsiung Medical University, Kaohsiung 80708, Taiwan

<sup>4</sup> School of Dentistry, Kaohsiung Medical University, Kaohsiung 80708, Taiwan

<sup>5</sup> Drug Development and Value Creation Research Center, Kaohsiung Medical University, Kaohsiung 80708, Taiwan

<sup>6</sup> Department of Computer Science and Information Engineering, National Penghu University of Science and Technology, Magong 880011, Taiwan

\* Correspondence: yudalinemail@gms.npu.edu.tw

**Abstract:** Economic forecasting is crucial in determining a country's economic growth or decline. Productivity and the labor force must be increased to achieve economic growth, which leads to the growth of gross domestic product (GDP) and income. Machine learning has been used to provide accurate economic forecasts, which are essential to sound economic policy. This study formulated a gated recurrent unit (GRU) neural network model to predict government expenditure, an essential component of gross domestic product. The GRU model was evaluated against autoregressive integrated moving average, support vector regression, exponential smoothing, extreme gradient boosting, convolutional neural network, and long short-term memory models using World Bank data regarding government expenditure from 1990 to 2020. The mean absolute error, root mean square error, and mean absolute percentage error were used as performance metrics. The GRU model demonstrates superior performance compared to all other models in terms of MAE, RMSE, and MAPE (with an average MAPE of 2.774%) when forecasting government spending using data from the world's 15 largest economies from 1990 to 2020. The results indicate that the GRU can be used to provide accurate economic forecasts.

**Keywords:** machine learning; economic forecasting; gated recurrent unit; neural network

**MSC:** 46M15



**Citation:** Yang, C.-H.; Molefyane, T.; Lin, Y.-D. The Forecasting of a Leading Country's Government Expenditure Using a Recurrent Neural Network with a Gated Recurrent Unit. *Mathematics* **2023**, *11*, 3085. <https://doi.org/10.3390/math11143085>

Academic Editor: Basilis Boutsinas

Received: 15 June 2023

Revised: 3 July 2023

Accepted: 11 July 2023

Published: 12 July 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Government budgets for education, health care, defense, and welfare are large [1], and they cannot be accurately predicted using traditional econometric models because of the complex nonlinear relationships existing between variables [2]. Researchers have recently applied artificial intelligence to help governments better allocate resources, deliver services, and coordinate large-scale operations [3]. A country's economic conditions and priorities are reflected in various factors that affect government spending, such as population size, which significantly impacts government spending [4]. A large population often requires increased investment in public services, healthcare, education, and social security programs. For example, in 2020 (World Development Indicators, 2021), countries with larger populations allocated a substantial portion of their budgets to social welfare, with spending on social welfare accounting for 30% to 45% of the total budget. The demand

for social services varies across country, encompassing areas such as poverty alleviation, healthcare, education, and unemployment benefits, thereby influencing government spending decisions. Gross domestic product (GDP) indicates a country's economic performance and total output. Government spending and GDP are closely related. During periods of GDP growth, governments often allocate more resources to infrastructure development and public services in order to support further economic growth. Conversely, during economic downturns or recessions, governments may implement austerity measures and reduce spending to manage budget deficits [5]. Government spending is also influenced by other macroeconomic variables, such as inflation, unemployment rates, and interest rates. Rising inflation rates can lead to increased spending on subsidies and welfare programs to mitigate the impact of price increases on citizens' purchasing power [6]. Through measures such as job creation programs or unemployment benefits, unemployment rates can affect government spending. In addition, interest rates play a crucial role in determining the cost of government borrowing, which affects spending decisions. By taking into account these factors related to government spending, policymakers can make informed decisions about budget allocation, revenue generation, and overall economic management with the economy to promote sustainable development and meet societal needs [7].

According to the Organisation for Economic Co-operation and Development (OECD), government investments in education and health care contribute to long-term economic growth; specifically, every 1% increase in education spending was reported to be correlated with a 0.5% increase in GDP per capita over the long term (source: OECD. Education at a Glance 2020: OECD Indicators). However, the relationship between government spending and GDP is complex; it depends on several factors, such as the state of the economy, the type and composition of government spending, and the effectiveness of government policies [8,9]. Therefore, accurate forecasts of government spending are crucial for improving resource allocation and economic planning. Accurate forecasts help governments budget more effectively [1] and with less risk [2]. Jeong et al. demonstrated that neural networks can be used to improve traditional time series models, such as seasonal autoregressive moving average (ARIMA) models, when applied to budget-related prediction [10]. Neural networks are a powerful means of capturing complex, nonlinear relationships between variables [11]. However, neural networks can accurately predict government spending only if a suitable set of factors that affect spending are selected; however, this is difficult to achieve. For example, Palmer et al. found that neural networks accurately predict economic tourism demand only if the appropriate input variables and network architecture are selected [12]. Despite these challenges, neural networks are promising as a means to predict government spending, even with complex time series data.

Machine learning models have produced accurate forecasts on the basis of time series data. For example, Lago et al. demonstrated that a gated recurrent unit (GRU) model (a deep learning model) outperformed its traditional counterparts in accurately predicting spot electricity prices. The GRU model had the highest prediction accuracy with a symmetric mean absolute percentage error (symmetric MAPE) of 13.04%, outperforming a deep neural network (the second best model with a symmetric MAPE of 12.34%), a long short-term memory (LSTM) model, a convolutional neural network (CNN) model, and even several mixed models [13]. Another study formulated a hybrid recurrent neural network (RNN) model based on GRU and LSTM that predicts daily and hourly multilevel wind power in Germany from data on wind speed at different heights [14]. The proposed model outperformed ARIMA and support vector regression (SVR) models in terms of gradient stability, training speed, mean absolute error (MAE), root mean square error (RMSE), and *t*-test results on long sequences. Li et al. proposed a novel ensemble decision method based on deep reinforcement learning to predict GDP. In this method, predictions from GRU, temporal convolutional network, and deep belief network models were taken as input to train three GDP prediction models, and the deep Q network algorithm was used to optimize the integration weight coefficients. This proposed method outperformed

18 competing methods in evaluation experiments, achieving MAPE values below 4.2% in all tests [15].

Government spending in the economy takes many forms. The composition of GDP among different types of spending is often of interest to economists. Government spending is one of four components of GDP, including consumption, investment, government purchases, and net exports. According to the International Monetary Fund (<https://www.imf.org/en/Home>), a 1% increase in government spending as a share of GDP is correlated with a 0.4 to 1.2 percentage point increase in GDP growth, depending on how government spending is allocated. Conversely, decreased government spending can exert contractionary effects and even lead to recession. Government spending is also related to other macroeconomic variables. For example, infrastructure spending boosts investment and productivity, and education and healthcare spending enhances the productivity and health of the workforce in particular and the population in general; these factors promote long-term economic growth.

In this study, we reviewed existing forecasting methods and developed a GRU model that predicts government spending on the basis of financial GDP indicators. We used a historical dataset to improve model generalizability. Our model outperformed six competing methods in terms of accuracy and reliability in an extensive battery of experiments involving time series datasets on the GDPs of 15 of the largest economies in the world. Our contributions are as follows.

1. Our GRU model accurately predicts government spending on the basis of financial GDP indicators.
2. Our GRU model outperformed ARIMA, exponential smoothing (ETS), extreme gradient boosting (XGBoost), SVR, CNN, and LSTM models in evaluation experiments.
3. The strengths and weaknesses of neural network models in predicting government expenditure were explored.
4. The aforementioned methods can capture complex nonlinear relationships between different economic factors to generate accurate predictions.

This paper is organized as follows. Section 2 reviews existing prediction approaches, namely, ARIMA, ETS, support vector machine (SVM), XGBoost, CNN, LSTM, and GRU, and presents our proposed approach. Section 3 describes the setup and results of our evaluation experiments. Finally, Section 4 concludes the paper.

## 2. Methods

Statistical models, especially ARIMA and exponential smoothing, are widely used in time series forecasting because they can capture patterns and dependencies in time series data [16]. Machine learning models are an improvement on such statistical methods and are becoming increasingly popular. These models are summarized as follows.

### 2.1. Autoregressive Integrated Moving Average

An ARIMA model describes the temporal evolution of the dependence between observations using the time lag between observations. It can be used to model stationary time series data wherein the mean and variance of the data remain constant over time. An ARIMA model combines autoregressive, differenced, and moving average components to model time series data [17] and has three main components: the number of lagged observations ( $p$ ), the difference in nonseasonal observations ( $d$ ), and the size of the moving average window ( $q$ ). These three components must be applied in the order expressed in the tuple  $(p, d, q)$ . In our experiment, the Python library *pmdarima* was used to implement ARIMA modeling. Subsequently, the *auto.arima* function in this package was used to create a process to automatically select the best parameters for the ARIMA model on the basis of the data. This helped the model quickly generate predictions.

## 2.2. Exponential Smoothing

ETS is used to model patterns and dependencies over time, primarily in nonstationary time series data. ETS models output predictions by assigning exponentially decaying weights to past observations [18]. The Holt linear exponential smoothing method is commonly used, and predictions can be obtained using the following equations:

$$\hat{y}_t = \alpha y_t + (1 - \alpha)(\hat{y}_{t-1} + b_{t-1}) \quad (1)$$

$$b_t = \gamma(\hat{y}_t - \hat{y}_{t-1}) + (1 - \gamma) b_{t-1} \quad (2)$$

$$\hat{y}_{t+m} = \hat{y}_t - b_t m \quad (3)$$

where  $\alpha$  and  $\gamma$  are smoothing constants ranging from 0 to 1. These standard equations are used to obtain the exponential trend detailed in [19]. In ETS, these techniques are used to simulate fundamental trends and patterns, primarily for nonstationary time series data. ETS models are popular because they are easy to implement and can rapidly provide accurate predictions [20,21].

## 2.3. Support Vector Regression

Introduced by Vapnik et al., SVM is a machine learning algorithm that can be used for classification and regression problems. SVR is a variant of SVM and is used to solve regression problems to make predictions on continuous target variables. SVR can be used when the relationship between the independent and dependent variables is nonlinear [22]. In regression, an SVR model identifies the best-fitting hyperplane with the maximum number of points. For a hyperplane  $Y = wx + b$ , the decision boundaries are  $wx + b = +a$  and  $wx + b = -a$ . Therefore, any hyperplane that satisfies SVR must lie in the bounds  $-a < (Y - wx + b) < +a$ . Thus, the goal is to identify a function that satisfies the decision boundary. SVR has been used to forecast, for example, the demand for wind and solar energy [23], the exchange rate of the Euro [24], the state of the climate [25], and the volume of airport freight [26]. In previous studies, SVR or hybrid methods with SVR have performed the best in evaluation experiments.

## 2.4. Extreme Gradient Boosting

A versatile and increasingly popular method designed to balance between performance and computational cost, XGBoost is an optimized version of the gradient boosting algorithm, and uses ensemble learning with multiple decision trees for prediction [27]. XGBoost has been applied in various fields, such as health care [28], computer vision [29], and missing data imputation [30].

## 2.5. Convolution Neural Network

CNNs consist of multiple layers and can be trained using a back-propagation algorithm. CNNs contain three layers: convolution, pool, and fully connected. The convolutional layer is responsible for learning the input's feature representation. It utilizes multiple convolutional kernels to compute different feature maps. In particular, the neurons in each feature map are connected to a set of neighboring neurons in the previous layer, and this set of neighboring neurons is called the receptive field of the neuron in the previous layer. To obtain a new feature map, the input is convolved with the trained convolution kernel. The resulting convolution is then subjected to an element-by-element nonlinear activation function. The activation function introduces nonlinear properties, which are advantageous for multilayer networks to detect nonlinear features. Commonly used activation functions include Sigmoid, tanh [31], and ReLU [Ref] the pooling layer is typically described as being located between two convolutional layers, and is responsible for achieving translation invariance by reducing the resolution of the feature maps. Each feature map in the pooling layer is connected to the corresponding feature map in the previous layer. In the initial layer,

the convolution kernel is designed to detect low-level features, such as edges and curves. Higher-level convolution kernels learn to encode more abstract features. By progressively stacking multiple convolutional and pooling layers, we can gradually extract higher-level feature representations. One or more fully connected layers designed for higher-level inference [30] may follow the multiple convolutional and pooling layers. The fully connected layer establishes connections between all neurons in the preceding layer and each neuron in the current layer, allowing for the generation of global semantic information.

### 2.6. Long Short-Term Memory

LSTM is a type of RNN specifically designed to overcome the vanishing gradient problem in traditional RNNs. It was introduced by Hochreiter and Schmidhuber in 1997 and has since been widely applied in various fields, including time series forecasting [32]. In LSTM, time series data are formulated as a sequence, wherein each observation corresponds to a time step. LSTM neural networks are then trained using previous observations to predict the next value in the sequence. Long-term dependencies in the data can be captured when the model is configured to use multiple previous time steps for each prediction. In recent studies, LSTM has had promising results in time series prediction. For example, LSTMs have produced more accurate forecasts of electricity load [33] and stock market trends [34] relative to competing methods. LSTM has also been used to predict wind power output [35], solar radiation [36], and agricultural production [37] from time series data. The LSTM architecture consists of a collection of recurrently connected memory blocks, which are subnetworks that maintain their state over time and regulate the flow of information through nonlinear gates. The structure of a single LSTM cell is shown in Figure 1A. These gates govern the interactions between different cells and regulate the flow of information. The input gate controls the process of updating the memory state. The output gate determines whether the output flow can influence the memory state of other cells. The forgetting gate determines whether the prior states should be remembered or forgotten. The LSTM is implemented using the following composite functions:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (4)$$

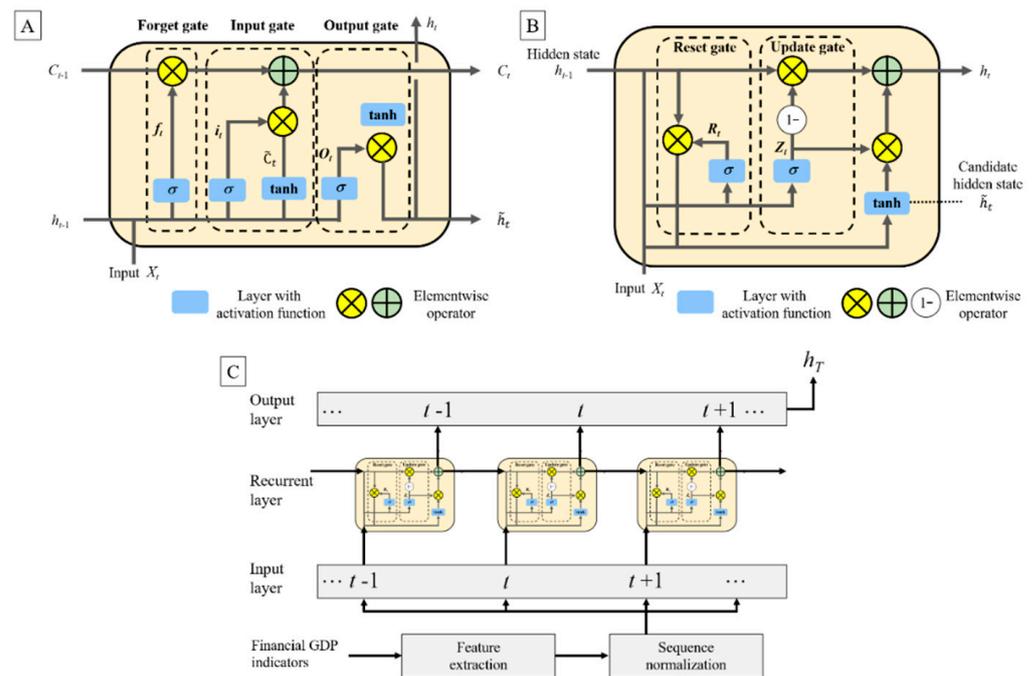
$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)$$

$$c_t = f_{t-1} + i_t \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (7)$$

$$h_t = o_t \tanh(c_t) \quad (8)$$

where  $\sigma$  represents the logistic sigmoid function,  $i$ ,  $f$ ,  $o$ , and  $c$  represent the input gate, forget gate, output gate, and cell input activation vectors, respectively.  $h$  represents the hidden vector. The subscripts of the weight matrix have intuitive meanings. For example,  $W_{hi}$  represents the hidden input gate matrix, and so on.



**Figure 1.** Structure of the GRU neural network. (A) LSTM cell. (B) GRU cell. (C) Framework of proposed government expenditure forecasting approach.

2.7. Gated Recurrent Unit

GRUs models are efficient, have fast training speed, and can address vanishing gradient problems in traditional RNNs, making them suited to applications involving large-scale time series prediction [38]. GRU methods have performed well in various applications, such as in the prediction of gas concentrations in mines [39] and the prediction of COVID-19 mortality rates [40].

Cho et al. (2014) proposed an architecture wherein GRU gate mechanisms, including a reset gate  $r_t$  and update gate  $Z_t$ , are used to efficiently handle vanishing gradient problems (Figure 1A) [33]. The reset gate is essential for the detection of short-term dependencies within sequences, allowing a network to selectively forget information from the previous time step and focus on the information that is most relevant to the current task. The update gate is responsible for capturing long-term dependencies in sequential data and determining how much information from the previous time step should be passed to the current time step; these determinations allow the network to adapt to changes in the input data and learn complex patterns over a long period.

During training, a GRU model learns the optimal values of the gates and other parameters by using the input data  $X_t$  and the expected output. A trained GRU model then outputs predictions based on input sequences by using the learned parameters. The learning process is encapsulated in Equations (9)–(12), which describe the units that capture the respective dependencies within the GRU network [41]:

$$Z_t = \sigma(h_{t-1}W_{xz}x_t + W_{hz}h_{t-1}) \tag{9}$$

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1}) \tag{10}$$

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \tilde{h}_t \tag{11}$$

$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \otimes h_{t-1})) \tag{12}$$

In these equations, two fully connected layers with a sigmoid activation function  $\sigma$  produce the output of two gates.  $W_{xz}$ ,  $W_{xr}$ ,  $W_{hz}$ , and  $W_{hr}$  denote the weight parameters of the reset gate and the update gate. The  $\otimes$  operator refers to element-wise multiplication, and  $\oplus$  refers to addition. When  $r_t$  is integrated and updated, the cell uses the hyperbolic tangent activation function  $\tanh$  at time step  $t$  to reach the candidate state defined in Equation (7). Subsequently, for the final memory update to be determined, Equation (6) is integrated with  $Z_t$ , and the hidden state  $h_t$  that matches the hidden state at time step  $t - 1$  is identified. Finally, if the new candidate state of the hidden layer conforms to the description given in Equation (7), the final memory state  $h_t$  is reached.

2.8. GDP Indicators Forecasting Framework

Figure 2 summarizes the flow of the experiment in which models were evaluated in terms of their performance in predicting government expenditure. Our proposed GRU model was designed to have a reset gate  $r_t$  and update gate  $Z_t$  (Figures 1A,B and 3; Algorithm 1) to capture dependencies between different periods. Specifically, the data were obtained and subjected to preprocessing (e.g., missing value evaluation and feature normalization). These data were then divided into training and testing data. Subsequently, the performances of trained models using various methods were evaluated in terms of MAE, RMSE, and mean absolute percentage error (MAPE). The results are visualized graphically. The framework detail of GRU algorithm is represented in Figure 3.

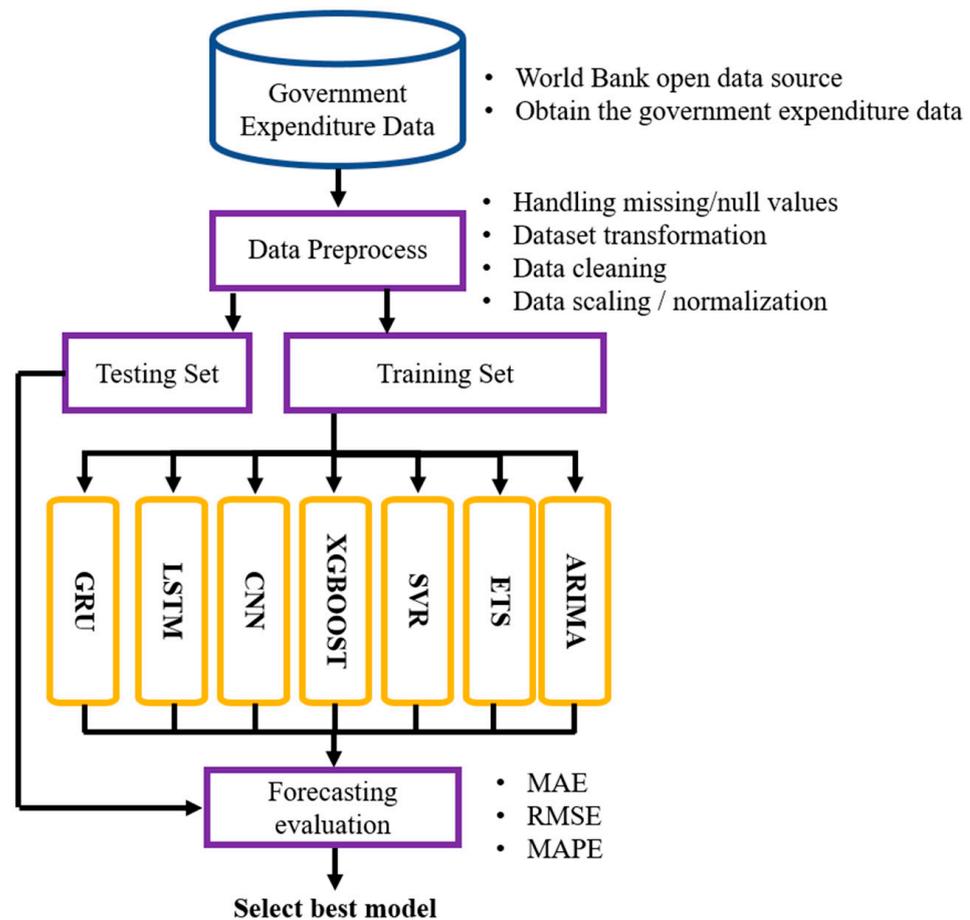


Figure 2. Framework of GDP indicators’ forecasting.

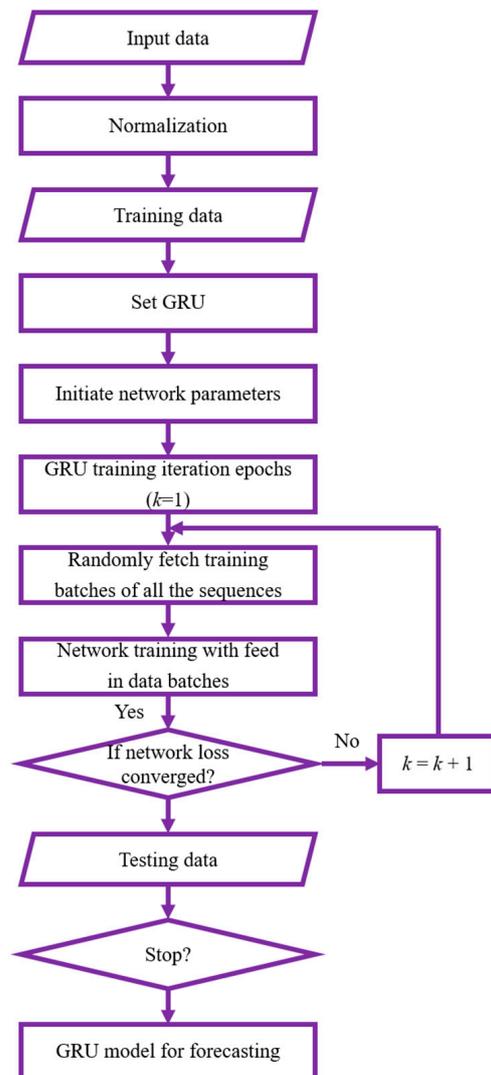
**Algorithm 1** GRU for Government Expenditure Forecasting

Let  $X$  be the dataset.

*Desired Output:* Prediction of best model  $F(x)$

1. Load time-series dataset, reconstruct, standardize, and preprocess the data
2. Split the data into training and testing sets, defined as  $X_{train}$  and  $X_{test}$ , respectively
3. Define GRU model structure and parameters
4. Update gate  $Z_t$  is obtained using (4)
5. Reset gate  $r_t$  is obtained using (5)
6. Candidate memory state  $\tilde{h}_t$  is obtained using (7)
7. Weighted variables,  $W$  are initialized
8. Final memory state  $h_t$  is computed using (6)
9. Calculate the loss function according to  $\operatorname{argmin}_W L(W) = \sum_{i=1}^N \operatorname{loss}(y_i, \hat{y}_i)$ , where  

$$\operatorname{loss}(y_i, \hat{y}_i) = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$
10. Use  $X_{train}$  to train the model
11. Evaluate model  $F(x)$  predictions using  $X_{test}$
12. Select the best model using performance metrics MAE, RMSE, and MAPE
13. Visualize and analyze the forecasting of the best model  $F(x)$



**Figure 3.** GRU model for forecasting.

### 2.9. Evaluation Criteria

MAE, MAPE, and RMSE are widely used metrics of rolling forecasting performance. They describe the deviation between actual and predicted values, and are defined as follows:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (13)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100\% \quad (14)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (15)$$

where  $y_i$  is the actual value,  $\hat{y}_i$  is the predicted value, and  $N$  is the sample size. The MAE is defined as the mean of the absolute differences between the actual and predicted values, and ranges from 0 to positive infinity. The MAPE is a relative indicator that is invariant to the units or magnitude of the actual and predicted values. Finally, the RMSE is the square root of the quotient that is the average deviation between actual and predicted values divided by the number of observations. The RMSE is sensitive to minor errors in model predictions, and is thus an effective metric of accuracy. For the three metrics, lower values (i.e., values closer to 0) indicate a lower error and higher prediction accuracy. These metrics are designed to comprehensively assess our forecasts' accuracy, precision, and reliability. Since MAE calculates the absolute difference between predicted and actual values, it provides an unbiased average accuracy metric. This provides a visual interpretation of the error and insight into the degree of deviation from the true value. On the other hand, RMSE emphasizes the effect of larger errors by considering squared differences. It provides a measure of the dispersion and magnitude of the error, and can assess the accuracy and reliability of the prediction. Finally, MAPE reflects the relative error between the predicted and actual values, providing insight into the proportional accuracy of the forecast. Each metric captures a different aspect of prediction performance, enabling a comprehensive understanding of the model's strengths and weaknesses. These metrics form a robust framework for assessing the forecasting power of the model and the applicability of GDP-based government spending forecasts. Using these performance metrics allows for a broad assessment of forecasting methods. The choice of these metrics depends on the specific situation and the nature of the forecasting task. They provide information on different aspects that help to understand the model's accuracy, precision, and reliability. By considering these metrics together, a comprehensive assessment of model performance can be obtained, and the strengths and weaknesses of the model can be identified.

## 3. Results and Discussion

### 3.1. Data Source

To ensure a comprehensive analysis, we will focus on data from 1990 to 2020 sourced from the World Bank (<https://data.worldbank.org/>). This time horizon captures a period of significant economic fluctuations, including major events such as financial crises, recessions, and economic growth. By considering a broad time horizon, we aim to understand government spending behavior over time comprehensively. In selecting countries, we base our selection on their respective GDP rankings. Specifically, we selected the 15 countries with the largest GDP to ensure representation of the major global economies. These countries are the United States, China, Japan, Germany, the United Kingdom, India, France, Italy, Canada, South Korea, Russia, Brazil, Australia, Spain, and Mexico. By including these countries, we aim to capture the various economic systems, levels of development, and geopolitical factors that may influence government spending patterns. The dataset was downloaded as a .csv file from the World Bank website and then transformed into a data frame in Python. The dataset obtained from the World Bank was of high quality, and did not

contain any missing values. The descriptive statistics (specifically, mean, median, standard deviation, sample size, first quartile [Q1], third quartile [Q3], and interquartile range [IQR]) for each of these 15 countries are presented in Table 1. We standardized the data using the integrated Python function `min_max_scaler`; this function rescales a data point  $x$  to between 0 and 1 using the following equation:  $(x - x_{\min}) / (x_{\max} - x_{\min})$ , where  $x$  is the value of the data point,  $x_{\min}$  is the minimum value, and  $x_{\max}$  is the maximum value.

**Table 1.** Descriptive statistics of government expenditure.

Country	Count	Min	Max	Mean	Med	SD	Q1	Q3	IQR
Australia	31.0	−3.610	6.986	2.829	3.091	1.849	1.780	3.714	1.934
Brazil	31.0	0.253	5.034	2.660	3.042	1.398	1.744	3.711	1.968
Canada	31.0	0.141	9.171	2.785	2.261	1.992	1.531	3.736	2.205
China	31.0	0.966	6.187	3.311	3.487	1.388	2.376	4.418	2.043
France	31.0	0.203	3.875	1.897	1.569	0.960	1.277	2.549	1.271
Germany	31.0	−0.725	12.732	2.018	1.843	2.350	0.563	2.608	2.045
India	31.0	0.027	3.621	1.254	1.056	0.863	0.612	1.862	1.251
Italy	31.0	−1.167	2.981	0.854	0.724	0.878	0.284	1.312	1.028
Japan	31.0	−0.052	1.221	0.238	0.129	0.288	0.046	0.350	0.304
Korea	31.0	0.212	2.156	0.855	0.780	0.495	0.496	1.033	0.537
Mexico	31.0	0.877	3.988	2.491	2.564	0.751	2.181	2.892	0.710
Russia	31.0	0.175	4.503	1.649	1.201	1.243	0.583	2.577	1.994
Spain	31.0	0.640	6.770	2.789	2.405	1.343	1.869	3.442	1.572
UK	31.0	−0.864	11.929	3.812	2.280	3.238	1.735	5.837	4.102
USA	31.0	0.465	3.406	1.593	1.473	0.762	1.034	2.058	1.024

Min: minimum value; Max: maximum value; Mean: average value; Med: median.

### 3.2. GRU Architecture Results and Sensitivity Analysis

We employed three optimizer tools, namely Adam, RMSprop, and Adagrad, to detect the optimal architecture for each neural network. The CNN model contains a 1D convolutional layer (CONV1D) with 200 filters and kernel size 1, followed by a MaxPooling layer. The LSTM model has two layers of 200 units each, followed by a one-cell dense layer. The GRU model has two layers of 200 units each, followed by a separate dense layer of 1 unit. The Adam optimizer and the mean absolute error loss function were used to create these neural network models. To capture the historical dependencies, the training data were prepared using a backward window with 1 and 1000 periods for each model. Table 2 presents a comparison of the parameter settings and the individual MAPE (%) results as a measure of precision. The adjusted period values, the number of neural network layers, and the activation function are shown. For the RMSprop optimizer, we increased the number of parameters while decreasing the number of periods. The overall MAPE value is slightly higher than the value in Table 2. For the Adagrad optimizer, we reduced the parameter and period values. It is important to note that the period values should not be excessively high, as they can lead to overfitting. The LSTM shows slightly better values with the parameter settings of the Adagrad optimizer. The CNN model shows a MAPE value of 86.798%, indicating a much worse performance. After a training procedure based on the optimizer, Adam optimizer produced an optimal GRU architecture, and the architectural detail is shown in Table 2.

We performed a sensitivity analysis, including experiments with parameter settings for different scenarios. The number of units in the GRU layer increased and decreased slightly. In addition, we adjusted the values of epochs to different lower values to observe the model's performance over time, since too many epochs can lead to overfitting. Table 3 presents the results of our sensitivity analysis. From Table 3, we can observe that lower epoch values, in combination with a higher number of GRU layers, tend to produce better prediction results for all three evaluated metrics. Notably, the differences between the accuracy results from the sensitivity analysis are not substantial. This indicates that our GRU model compiles correctly and performs consistently under different assumptions and

scenarios. The chosen optimizer is appropriate for our dataset, and can produce sufficiently accurate results, as shown in Table 3. The sensitivity analysis strengthens the robustness of our model and provides valuable insight into the effects of varying the parameters used. It allows us to understand the optimal parameter settings for our particular task of forecasting government spending. The consistent performance of our model across different parameter configurations gives us confidence in its accuracy and reliability.

**Table 2.** Comparison of the number of parameters to optimize for each neural network model.

Optimizer	Methods	Parameters					MAPE
Adam	GRU	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		1000	200	200	256	0.004	2.774
	LSTM	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		1000	200	200	256	0.004	4.208
	CNN	Epochs	CONV1D	CNN_DENSE	Batch-size	Learning rate	
		1000	200	50	256	0.004	3.389
RMSprop	GRU	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		100	400	400	256	0.004	2.809
	LSTM	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		100	400	400	256	0.004	4.209
	CNN	Epochs	CONV1D	CNN_DENSE	Batch-size	Learning rate	
		100	400	100	256	0.004	4.033
Adagrad	GRU	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		32	100	100	256	0.004	2.852
	LSTM	Epochs	Layer <sub>1</sub>	Layer <sub>2</sub>	Batch-size	Learning rate	
		32	100	100	256	0.004	4.207
	CNN	Epochs	CONV1D	CNN_DENSE	Batch-size	Learning rate	
		32	100	10	256	0.004	86.798

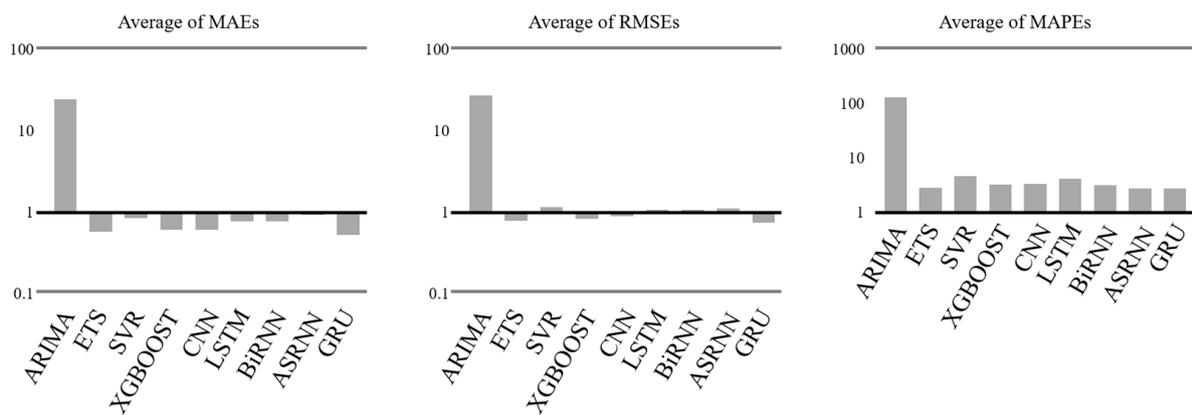
**Table 3.** Sensitivity analysis of GRU model.

	L250, ep100	L150, ep100	L250, ep32	L150, ep32	L200, p1000
MAE	0.518	0.518	0.510	0.521	0.528
RMSE	0.741	0.738	0.720	0.743	0.747
MAPE (%)	2.794	2.784	2.775	2.804	2.774

L: number of units in the layer, ep: Epochs.

### 3.3. Experimental System

Python was used for data mining. Our GRU model was evaluated against ARIMA, ETS, SVR, XGBOOST, CNN, and LSTM models. The LSTM and GRU models were optimized using Adam [42]. Different learning rates  $\alpha$  ranging from 0.1 to 0.0001 were tested, and the corresponding validation loss was observed. Finally, a learning rate of 0.01 was chosen, and  $\beta_1$  and  $\beta_2$  were set to their default values of 0.9 and 0.999, respectively. The training and testing sets comprised data from 1990–2014 and 2015–2020, respectively. Figure 4 and Table 4 present the MAE, RMSE, and MAPE results of the seven models for all and each of the 15 countries, respectively.



**Figure 4.** Performance evaluation based on the average government expenses forecasting metrics for the 15 countries.

**Table 4.** The experimental results of the traffic volume forecast model: MAPE, MAE, and MAPE values.

Country	Metric	ARIMA	ETS	SVR	XGB	CNN	LSTM	BiRNN	ASRNN	GRU
Australia	MAE	24.854	<b>0.513</b>	1.267	0.747	0.788	1.345	1.341	0.977	0.618
	RMSE	27.672	<b>0.782</b>	1.492	0.970	1.082	1.548	1.544	0.959	0.852
	MAPE(%)	135.732	<b>2.585</b>	6.414	3.794	3.976	6.818	5.633	2.967	3.089
Brazil	MAE	4.637	<b>0.186</b>	1.136	0.750	0.5134	0.977	0.975	0.982	0.322
	RMSE	4.954	<b>0.286</b>	1.164	0.821	0.6463	1.008	1.007	0.884	0.379
	MAPE(%)	24.177	<b>1.196</b>	5.628	3.727	2.538	4.837	3.551	2.781	1.594
Canada	MAE	30.841	0.543	0.585	0.585	0.587	0.867	0.865	1.053	<b>0.526</b>
	RMSE	35.102	<b>0.773</b>	0.926	0.836	0.920	1.108	1.106	1.155	0.825
	MAPE(%)	150.088	<b>1.844</b>	2.665	2.684	2.679	4.007	3.098	2.998	2.402
China	MAE	1.332	0.973	0.786	0.721	0.310	0.669	0.670	0.628	<b>0.166</b>
	RMSE	1.592	0.209	0.850	0.772	0.395	0.699	0.700	0.328	<b>0.197</b>
	MAPE(%)	9.100	1.128	4.741	4.363	1.874	4.043	3.668	2.885	<b>1.005</b>
France	MAE	35.299	0.712	0.578	0.569	0.570	0.769	0.767	1.007	<b>0.586</b>
	RMSE	39.687	0.798	0.728	0.993	0.775	1.010	1.008	0.864	<b>0.796</b>
	MAPE(%)	150.914	<b>2.333</b>	2.403	2.319	2.363	3.167	2.553	3.225	2.422
Germany	MAE	36.711	<b>0.575</b>	0.788	0.854	0.625	0.753	0.752	1.235	0.591
	RMSE	41.099	0.990	1.262	1.150	1.089	1.206	1.205	1.255	<b>0.790</b>
	MAPE(%)	188.378	<b>2.666</b>	3.675	4.049	2.894	3.509	3.029	2.884	2.810
India	MAE	3.752	0.543	0.570	0.479	0.636	0.694	0.694	1.439	<b>0.419</b>
	RMSE	3.902	0.735	0.841	0.521	0.909	1.007	1.006	1.160	<b>0.617</b>
	MAPE(%)	34.850	4.746	4.861	4.286	5.468	5.922	4.771	3.274	<b>3.584</b>
Italy	MAE	37.742	<b>0.460</b>	0.692	0.662	0.530	0.588	0.587	0.776	0.598
	RMSE	42.589	<b>0.852</b>	0.817	0.959	0.849	0.957	0.957	0.923	0.936
	MAPE(%)	196.691	<b>2.260</b>	3.527	3.318	2.641	2.916	1.072	2.884	2.979
Japan	MAE	18.049	<b>0.335</b>	1.133	0.401	0.398	0.451	0.451	0.617	0.366
	RMSE	19.050	0.588	1.556	0.581	0.588	0.719	0.718	0.626	<b>0.532</b>
	MAPE(%)	100.373	<b>1.627</b>	5.577	1.964	1.952	2.195	0.225	3.164	1.796
Korea	MAE	13.022	0.973	1.892	0.520	1.029	1.074	1.070	0.884	<b>0.540</b>
	RMSE	13.252	1.086	2.381	0.646	1.313	1.526	1.520	2.861	<b>0.644</b>
	MAPE(%)	99.823	5.972	11.182	3.090	6.224	6.245	6.400	2.776	<b>3.235</b>
Mexico	MAE	1.774	<b>0.436</b>	0.441	0.432	0.477	0.555	0.554	2.045	0.441
	RMSE	2.182	0.617	0.762	0.631	0.589	0.746	0.745	1.055	<b>0.616</b>
	MAPE(%)	18.7325	<b>3.513</b>	3.519	3.506	3.953	4.478	3.527	2.818	3.596
Russia	MAE	40.023	<b>0.661</b>	0.644	0.652	0.724	0.828	0.828	0.791	0.832
	RMSE	45.101	<b>1.079</b>	1.060	0.980	1.165	1.306	1.305	1.163	1.005
	MAPE(%)	222.661	4.420	3.277	3.337	3.684	4.204	3.057	3.116	<b>3.362</b>
	MAE	52.074	0.712	0.765	0.731	0.798	0.802	0.800	1.487	<b>0.675</b>

Table 4. Cont.

Country	Metric	ARIMA	ETS	SVR	XGB	CNN	LSTM	BiRNN	ASRNN	GRU
Spain	RMSE	58.233	1.258	1.437	1.039	1.253	1.384	1.383	1.483	<b>1.242</b>
	MAPE(%)	279.389	3.399	3.637	3.604	3.877	3.831	2.982	1.975	<b>3.214</b>
	MAE	60.055	<b>0.850</b>	0.928	0.916	0.963	1.047	1.046	0.055	1.021
UK	RMSE	67.578	1.499	1.506	1.515	1.495	1.715	1.715	1.781	<b>1.456</b>
	MAPE(%)	311.637	<b>4.010</b>	4.450	4.362	4.651	4.973	2.478	2.578	4.957
	MAE	12.714	0.239	0.737	0.208	0.2950	0.286	0.286	0.013	<b>0.227</b>
USA	RMSE	14.765	0.352	0.783	<b>0.243</b>	0.380	0.401	0.402	0.343	0.318
	MAPE(%)	85.847	1.660	5.243	1.459	2.068	1.979	1.372	1.875	<b>1.570</b>
	MAE	24.858	0.581	0.863	0.615	0.616	0.780	0.779	0.933	<b>0.528</b>
Total Average	RMSE	27.784	0.793	1.171	0.844	0.897	1.089	1.088	1.123	<b>0.747</b>
	MAPE(%)	133.893	2.891	4.720	3.324	3.389	4.208	3.161	2.813	<b>2.774</b>

Bold means the lowest value.

### 3.4. Comparison and Discussion

Artificial intelligence encompasses a range of techniques that aim to mimic living things' behavior to improve decision-making and mitigate potential risks to economic stability and growth. One such technique is GRU, which focuses on automatically identifying meaningful patterns in data. GRU has become a valuable tool for extracting information from large datasets [43]. Departure from traditional computing methods is a common feature of GRU applications. Due to the complexity of the patterns identified, it is impractical for a human programmer to provide explicit and detailed instructions for performing these tasks. In addition, the GRU techniques can learn from a wide variety of data types and recognize patterns that may be impossible for a human to detect. This enables GRU to operate in resource-constrained environments [44]. As a result, GRU could play a critical role in public administration, and provide insights that can inform decision-makers.

The GRU model was evaluated against ARIMA, ETS, SVR, XGBOOST, CNN, LSTM, BiRNN, and ASRNN models. The LSTM, BiRNN, ASRNN, and GRU models were optimized using Adam [42]. ARIMA had much lower performance compared with the other models. To improve ARIMA performance, the lag order ( $p$ ), moving average order ( $q$ ), and differencing degree ( $d$ ) can be manually rather than automatically configured [45], and other diagnostic measures can be applied [46,47]. The remaining models produced reasonably accurate predictions. We analyzed different RNN variants, including the bidirectional RNN (BiRNN) [48] and attention-based [49] sequential RNN (ASRNN) models. For the BiRNN model, we performed dataset preprocessing and used a BiRNN architecture along with a dropout layer to prevent overfitting. Similarly, for the ASRNN model, the dataset underwent the same preprocessing step, and the architecture employed an attention mechanism to capture the relevant information from the sequential data. Different learning rates  $\alpha$  ranging from 0.1 to 0.0001 were tested, and the corresponding validation loss was observed. Finally, a learning rate of 0.01 was chosen, and  $\beta_1$  and  $\beta_2$  were set to their default values of 0.9 and 0.999, respectively. The training and testing sets comprised data from 1990–2014 and 2015–2020, respectively. Figure 4 and Table 4 present the MAE, RMSE, and MAPE results of the seven models for all and each of the 15 countries, respectively.

Our GRU model performed the best when used to forecast government expenditure (Table 4). The average MAE, RMSE, and MAPE values for country-level predictions were 0.5284, 0.7469, and 2.774%, respectively. ETS, XGBoost, CNN, BiRNN, and ASRNN also produced satisfactory country-level predictions. The country-level predictions of the seven models and the actual values are plotted in Figure 5. The actual and predicted trends for all models except the ARIMA model had a close fit upon visual inspection. Accurate predictions for China were difficult to achieve, although the predictions of ETS, CNN, BiRNN, ASRNN, and GRU were reasonably close. In general, GRU captured outliers and changes in trends well, unlike ARIMA. Some methods, such as LSTM and SVR, follow an

average trend starting from a particular value. Data spanning a longer period are required to more fully analyze the capabilities of these methods, especially GRU.

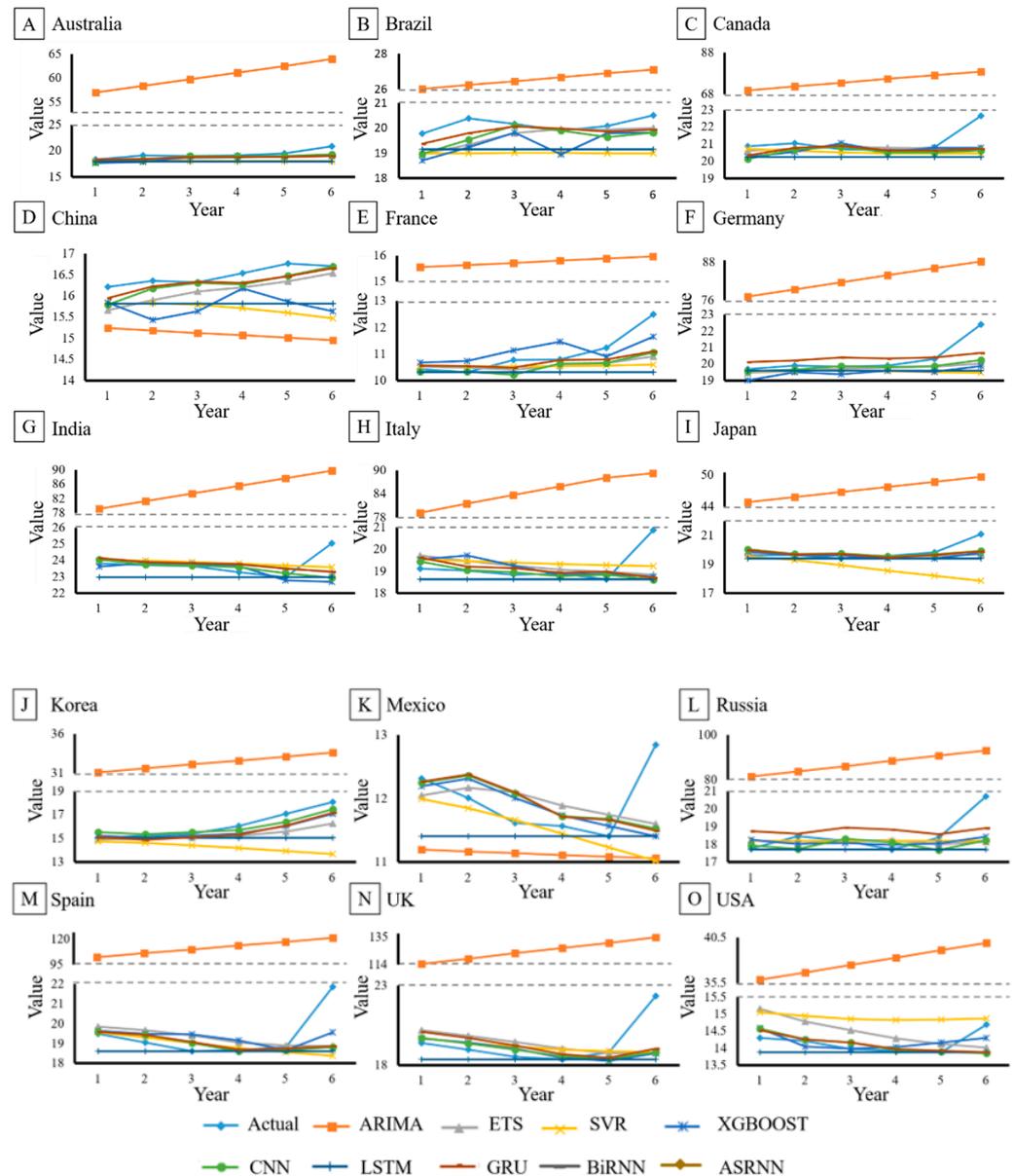


Figure 5. Comparison of seven forecasting algorithms for government expenditure.

Since GRU uses only two gates instead of three, and one memory cell, it is considered an efficient implementation of LSTM. Despite the simplicity of GRU, it still achieves better results than LSTM [43]. LSTM was introduced to address the problems of vanishing and exploding gradients in RNNs. LSTM uses memory cells and gates to facilitate the flow of information over time. Subsequently, GRU was introduced as a more efficient form of LSTM, which combines two LSTM gates without a memory cell and reveals the full hidden content without any control [41]. Compared to LSTM, GRU, therefore, exhibits higher computational efficiency. For the advantage of input information, bidirectional RNNs were introduced. BiRNN uses information from both directions to predict output sequences by cascading and connecting backward and forward hidden states to each output node. The GRU addresses the problems of gradient explosion and the long-term dependency of RNNs, and requires fewer training parameters compared to the LSTM, which is also a variant of the RNN. Both the GRU and the LSTM have their advantages and disadvantages in

practical applications [41]. Determining effective hyperparameters, such as the number of neuron units and the learning rate, remains a challenging problem, despite GRU's good performance in time series prediction and its wide applicability. Optimizers are powerful tools for the optimization of hyperparameters. In this study, the hyperparameters of the GRU were optimized using Adam, RMSprop, and RMSprop. Additionally, a sensitivity analysis improved the robustness of our model and provided valuable insight into how changing the parameters affected the model. As a result, we were able to understand the optimal parameter settings for the prediction of government expenditure, the specific task at hand. We are confident in the accuracy and reliability of our model because of its consistent performance across different parameter configurations.

The choice of input variables is critical to the accuracy of expenditure forecasting. Traditionally, a priori expert knowledge, trial and error, or linear cross-correlation analysis are used to select these variables. However, expert knowledge is often challenging and time-consuming to acquire, and relying solely on expert knowledge can introduce bias. Trial and error input selection can be computationally intensive, especially for data-driven models with many potential input candidates. In addition, the linear correlation coefficients commonly used only assess linear relationships and do not capture the non-linear dynamics often present in data-driven models. Artificial intelligence encompasses a range of techniques that aim to mimic living things' behavior to improve decision-making and mitigate potential risks to economic stability and growth. One such technique is GRU, which focuses on automatically identifying meaningful patterns in data. GRU has become a valuable tool for extracting information from large datasets [43]. Departure from traditional computing methods is a common feature of GRU applications. Due to the complexity of the patterns identified, it is impractical for a human programmer to provide explicit and detailed instructions for performing these tasks. In addition, GRU techniques can learn from a wide variety of data types and recognize patterns that may be impossible for a human to detect. This enables GRU to operate in resource-constrained environments [45]. As a result, GRU could play a critical role in public administration and provide insights that can inform decision-makers.

In general, GRU outperformed ARIMA, SVR, ETS, XGBoost, CNN, LSTM, BiRNN, and ASRNN in predicting government expenditure on the basis of data on prior government spending, household consumption, investment, imports, and exports, which are correlated with GDP. Furthermore, unlike ARIMA, GRU does not require refitting to produce accurate predictions. ETS, CNN, LSTM, BiRNN, and ASRNN also provided reasonably accurate predictions.

#### 4. Conclusions

Algorithms such as GRU are becoming increasingly capable of economic prediction as machine learning becomes more sophisticated and datasets become larger. In our study, our proposed GRU method outperformed ARIMA, ETS, SVR, XGBOOST, CNN, and LSTM in terms of MAE, RMSE, and MAPE (average MAPE = 2.774%) when used to predict government spending on the basis of 1990–2020 data from 15 of the largest economies in the world, namely Australia, Brazil, Canada, China, France, Germany, India, Italy, Japan, South Korea, Mexico, Russia, Spain, the United Kingdom, and the United States. Further studies should incorporate predictors pertaining to political events, social changes, and the environment, and apply the resultant models to emerging markets. In general, accurate forecasts of economic trends are indispensable for ensuring sustainable economic growth.

**Author Contributions:** C.-H.Y.: conceptualization, supervision, project administration. T.M.: methodology, writing—review and editing. Y.-D.L.: conceptualization, project administration. All authors have read and agreed to the published version of the manuscript.

**Funding:** The funding source is the Ministry of Science and Technology, Taiwan (under grant no. 111-2221-001-MY3 and 111-2221-002-MY3).

**Data Availability Statement:** Data were obtained from World Bank (<https://data.worldbank.org/>).

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Kumar, R.; Lal, K. Forecasting government expenditure using machine learning algorithms: Evidence from India. *Technol. Forecast. Soc. Chang.* **2017**, *123*, 251–260.
2. Harmsen, R.; Sterneberg, A. Forecasting government expenditure with macroeconomic aggregates in small open economies. *Econ. Model.* **2017**, *60*, 302–312.
3. Dwivedi, Y.K.; Hughes, L.; Ismagilova, E.; Aarts, G.; Coombs, C.; Crick, T.; Duan, Y.; Dwivedi, R.; Edwards, J.; Eirug, A. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy. *Int. J. Inf. Manag.* **2021**, *57*, 101994. [[CrossRef](#)]
4. Lara-Rubio, J.; Navarro-Galera, A.; Buendia-Carrillo, D.; Gomez-Miranda, M.E.J.C. Analysing financial risks of local governments to design sustainability policies for public services: An empirical study by the population size. *Cities* **2022**, *128*, 103795. [[CrossRef](#)]
5. Wei, X.; Mohsin, M.; Zhang, Q.J.R.E. Role of foreign direct investment and economic growth in renewable energy development. *Renew. Energy* **2022**, *192*, 828–837. [[CrossRef](#)]
6. Ginn, W.; Pourroy, M.J.E.M. The contribution of food subsidy policy to monetary policy in India. *Econ. Model.* **2022**, *113*, 105904. [[CrossRef](#)]
7. Guerrero, O.A.; Castañeda, G.; Trujillo, G.; Hackett, L.; Chávez-Juárez, F.J.S.-E.P.S. Subnational sustainable development: The role of vertical intergovernmental transfers in reaching multidimensional goals. *Socio-Econ. Plan. Sci.* **2022**, *83*, 101155. [[CrossRef](#)]
8. Barro, R.J. Government spending in a simple model of endogenous growth. *J. Political Econ.* **1990**, *98*, S103–S125. [[CrossRef](#)]
9. Scotti, F.; Flori, A.; Pammolli, F. The economic impact of structural and Cohesion Funds across sectors: Immediate, medium-to-long term effects and spillovers. *Econ. Model.* **2022**, *111*, 105833. [[CrossRef](#)]
10. Jeong, K.; Koo, C.; Hong, T. An estimation model for determining the annual energy cost budget in educational facilities using SARIMA (seasonal autoregressive integrated moving average) and ANN (artificial neural network). *Energy* **2014**, *71*, 71–79. [[CrossRef](#)]
11. Robinson, H.; Pawar, S.; Rasheed, A.; San, O. Physics guided neural networks for modelling of non-linear dynamics. *Neural Netw.* **2022**, *154*, 333–345. [[CrossRef](#)] [[PubMed](#)]
12. Palmer, A.; Montano, J.J.; Sesé, A. Designing an artificial neural network for forecasting tourism time series. *Tour. Manag.* **2006**, *27*, 781–790. [[CrossRef](#)]
13. Lago, J.; De Ridder, F.; De Schutter, B. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms. *Appl. Energy* **2018**, *221*, 386–405. [[CrossRef](#)]
14. Farah, S.; Humaira, N.; Aneela, Z.; Steffen, E. Short-term multi-hour ahead country-wide wind power prediction for Germany using gated recurrent unit deep learning. *Renew. Sustain. Energy Rev.* **2022**, *167*, 112700. [[CrossRef](#)]
15. Li, Q.; Yu, C.; Yan, G. A New Multipredictor Ensemble Decision Framework Based on Deep Reinforcement Learning for Regional GDP Prediction. *IEEE Access* **2022**, *10*, 45266–45279. [[CrossRef](#)]
16. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [[CrossRef](#)]
17. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
18. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*; OTexts: Melbourne, Australia, 2018.
19. Gardner, E.S., Jr. Exponential smoothing: The state of the art—Part II. *Int. J. Forecast.* **2006**, *22*, 637–666. [[CrossRef](#)]
20. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 2.
21. Shumway, R.H.; Stoffer, D.S.; Stoffer, D.S. *Time Series Analysis and Its Applications*; Springer: Berlin/Heidelberg, Germany, 2000; Volume 3.
22. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999.
23. Sharifzadeh, M.; Sikinioti-Lock, A.; Shah, N. Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression. *Renew. Sustain. Energy Rev.* **2019**, *108*, 513–538. [[CrossRef](#)]
24. Sermpinis, G.; Stasinakis, C.; Theofilatos, K.; Karathanasopoulos, A. Modeling, forecasting and trading the EUR exchange rates with hybrid rolling genetic algorithms—Support vector regression forecast combinations. *Eur. J. Oper. Res.* **2015**, *247*, 831–846. [[CrossRef](#)]
25. Murillo-Escobar, J.; Sepulveda-Suescun, J.; Correa, M.; Orrego-Metaute, D. Forecasting concentrations of air pollutants using support vector regression improved with particle swarm optimization: Case study in Aburrá Valley, Colombia. *Urban Clim.* **2019**, *29*, 100473. [[CrossRef](#)]
26. Yang, C.-H.; Shao, J.-C.; Liu, Y.-H.; Jou, P.-H.; Lin, Y.-D. Application of Fuzzy-Based Support Vector Regression to Forecast of International Airport Freight Volumes. *Mathematics* **2022**, *10*, 2399. [[CrossRef](#)]
27. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
28. Kumar, P.; Nestsiarovich, A.; Nelson, S.J.; Kerner, B.; Perkins, D.J.; Lambert, C.G. Imputation and characterization of uncoded self-harm in major mental illness using machine learning. *J. Am. Med. Inform. Assoc.* **2020**, *27*, 136–146. [[CrossRef](#)]

29. Memon, N.; Patel, S.B.; Patel, D.P. Comparative analysis of artificial neural network and XGBoost algorithm for PolSAR image classification. In Proceedings of the Pattern Recognition and Machine Intelligence: 8th International Conference, PReMI 2019, Tezpur, India, 17–20 December 2019; pp. 452–460.
30. Su, Y.-C.; Wu, C.-Y.; Yang, C.-H.; Li, B.-S.; Moi, S.-H.; Lin, Y.-D. Machine learning data imputation and prediction of foraging group size in a Kleptoparasitic spider. *Mathematics* **2021**, *9*, 415. [[CrossRef](#)]
31. LeCun, Y.; Bottou, L.; Orr, G.B.; Müller, K.-R. Efficient backprop. In *Neural Networks: Tricks of the Trade*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 9–50.
32. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
33. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-term residential load forecasting based on LSTM recurrent neural network. *IEEE Trans. Smart Grid* **2017**, *10*, 841–851. [[CrossRef](#)]
34. Nelson, D.M.; Pereira, A.C.; De Oliveira, R.A. Stock market's price movement prediction with LSTM neural networks. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1419–1426.
35. Duan, J.; Wang, P.; Ma, W.; Tian, X.; Fang, S.; Cheng, Y.; Chang, Y.; Liu, H. Short-term wind power forecasting using the hybrid model of improved variational mode decomposition and Correntropy Long Short-term memory neural network. *Energy* **2021**, *214*, 118980. [[CrossRef](#)]
36. Huang, X.; Li, Q.; Tai, Y.; Chen, Z.; Zhang, J.; Shi, J.; Gao, B.; Liu, W. Hybrid deep neural model for hourly solar irradiance forecasting. *Renew. Energy* **2021**, *171*, 1041–1060. [[CrossRef](#)]
37. Schwalbert, R.A.; Amado, T.; Corassa, G.; Pott, L.P.; Prasad, P.V.; Ciampitti, I.A. Satellite-based soybean yield forecast: Integrating machine learning and weather data for improving crop yield prediction in southern Brazil. *Agric. For. Meteorol.* **2020**, *284*, 107886. [[CrossRef](#)]
38. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv* **2014**, arXiv:1409.1259.
39. Jia, P.; Liu, H.; Wang, S.; Wang, P. Research on a mine gas concentration forecasting model based on a GRU network. *IEEE Access* **2020**, *8*, 38023–38031. [[CrossRef](#)]
40. Sankaranarayanan, S.; Balan, J.; Walsh, J.R.; Wu, Y.; Minnich, S.; Piazza, A.; Osborne, C.; Oliver, G.R.; Lesko, J.; Bates, K.L. COVID-19 mortality prediction from deep learning in a large multistate electronic health record and laboratory information system data set: Algorithm development and validation. *J. Med. Internet Res.* **2021**, *23*, e30157. [[CrossRef](#)]
41. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv* **2014**, arXiv:1412.3555.
42. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
43. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.
44. Gutierrez-Torre, A.; Bahadori, K.; Baig, S.-U.; Iqbal, W.; Vardanega, T.; Berral, J.L.; Carrera, D. Automatic distributed deep learning using resource-constrained edge devices. *IEEE Internet Things J.* **2021**, *9*, 15018–15029. [[CrossRef](#)]
45. Tsay, R.S. *Analysis of Financial Time Series*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
46. Brockwell, P.J.; Davis, R.A. *Time Series: Theory and Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
47. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
48. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [[CrossRef](#)]
49. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; Volume 30.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.