Quantum Computing and Deep Learning Methods for GDP Growth Forecasting

David Alaminos[1] , *M. Belén Salas*[2], *Manuel A. Fernández − Gámez*[3]

1.Department of Financial Mangement, Universidad Pontifcia Comillas, Madrid, Spain

2 PhD Program in Economics and Business, Universidad de Málaga, Málaga, Spain

3 Department of Finance and Accounting, Universidad de Málaga, Málaga, Spain

Abstract

Precise macroeconomic forecasting is one of the major aims of economic analysis because it facilitates a timely assessment of future economic conditions and can be used for monetary, fiscal, and economic policy purposes. Numerous works have studied the behavior of the macroeconomic situation and have developed models to forecast them. However, the existing models have limitations, and the literature demands more research on the subject given that the accuracy of the models is still poor, and they have only been expanded for developed countries. This paper presents a comparison of methodologies for GDP growth forecasting and, consequently, new forecasting models of GDP growth have been constructed with the ability to estimate accurately future scenarios globally. A sample of 70 countries was used, which has allowed the use of sample combinations that consider the regional heterogeneity of the warning indicators. To the sample under study, different methods have been applied to achieve a high accuracy model, comparing Quantum Computing with Deep Learning procedures, being Deep Neural Decision Trees, which has provided excellent prediction results thanks to large-scale processing with mini-batch-based learning and can be connected to any larger Neural Networks model. Our model has a great potential impact on the adequacy of macroeconomic policy, providing tools that help to achieve macroeconomic and monetary stability at the global level, and creating new methodological opportunities for GDP growth forecasting.

Keywords: Macroeconomic forecasting, GDP growth, Deep Learning, Quantum Computing, Macroeconomic stability.

1. Introduction

Forecasting GDP growth with great precision is vitally important for policy-makers, central banks, and private economic agents since they use the macroeconomic forecasts to determine fiscal and monetary policies and plan future operating activities. On the other hand, the selection of suitable GDP growth forecasting methods has been of huge interest among researchers and academics (Kapetanios, Marcellino and Papailias, 2016).

In the most recent literature, some GDP growth forecasting models stand out (Carriero, Galvão and Kapetanios, 2019; Carriero, Clark and Marcellino, 2019; Claveria, Monte and Torra, 2019; Kapetanios, Marcellino and Papailias, 2016; Marcellino, Porqueddu and Venditti, 2016; Clark and Ravazzolo, 2015; Ferrara, Marcellino and Mogliani, 2015; Schorfheide and Song, 2015). Most of the existing literature on GDP growth has been focused on developed economies, mainly the United States and Europe. The evidence is limited to emerging countries. These models have shown that vector autoregressions and Bayesian vector autoregressions are widely used in empirical macroeconomics to predict macroeconomic and financial variables (Carriero, Galvão and Kapetanios, 2019; Marcellino, Porqueddu and Venditti, 2016; Schorfheide and Son, 2015). So, the existing models have used different methods for GDP growth forecasting, especially statistical methods, and some computational methods as an evolutionary algorithm have been developed (Claveria, Monte and Torra, 2019), but no quantum computing or deep learning methods have been applied. For this reason, the literature demands a new GDP growth forecasting model, specifically in terms of new models that provide a better fit in scenarios globally and compare different methods to achieve more accurate results (Carriero, Galvão and Kapetanios, 2019). Besides, although the explanatory capacity of these models is significant, they still have certain limitations related to their levels of precision and their exclusive focus on groups of developed countries.

To contribute to the accuracy capability of GDP growth forecasting models, in the present study a comparison of methodologies to forecast GDP growth has been analyzed and, as a consequence new models that will generate better forecasts of behavior in the future of GDP growth. These models can predict in all countries, also achieving accuracy levels above 93%. These models have been constructed from a sample of 70 countries (47 emerging countries and 23 developed countries). Quantum computing and deep learning methods have been applied in the construction of the GDP growth forecasting models to compare them and determine the highest accuracy model. Specifically, the quantum computing methods are Support Vector Regression Chaotic Quantum Bat Algorithm, Quantum Boltzmann Machines, and Quantum Neural Networks. For its part, the Deep Learning methods are Deep Recurrent Convolution Neural Network, Deep Belief Network, Deep Neural Decisions Trees and Deep Learning using Support Vector Machines. The Deep Neural Decisions Trees model has obtained the highest levels of precision.

We make at least three further contributions to the literature. First, we consider explanatory variables for predicting GDP growth, testing the importance of these variables which have not been considered so far. It has important implications for policymakers, who will know which indicators provide reliable, accurate, and potential macroeconomic forecasting. Second, we improve the forecasting accuracy concerning that obtained in previous studies with innovative methodologies, concluding that the deep learning methods predict macroeconomic forecasting better than quantum computing methods, although the last ones have obtained very good results. Third, our study has studied GDP growth globally, and so not restricted to developed countries. It is interesting for those responsible for the economic policies of any country in the world.

This study is structured as follows: Section 2 provides a literature review of empirical research on GDP growth forecasting. Section 3 sets out the methodology used. Section 4 provides details of the data and the variables used in the study. Finally, section 5 analyses the results obtained. The article concludes by stating the conclusions of the study and its implications.

1.1. Literature review

The existing literature has mainly focused on predicting GDP growth in the United States (Batchelor and Dua, 1992; Batchelor and Dua, 1998; Stock and Watson, 2002; Clements and Galvão, 2008; Marcellino, 2008; Clark, 2011; Clark and Ravazzolo, 2015; Barsoum and Stankiewicz, 2015; Carriero, Galvão and Kapetanios, 2019; Carriero, Clark and; Marcellino, 2019). On the other hand, a large majority of studies have developed predictions of GDP growth for the euro area and in specific countries of Europe, mainly for Scandinavian economies (Bergstroöm, 1995; Camba-Mendez, Kapetanios, Smith and

Weale, 2001; Hansson, Jansson and Löf, 2005; Martinsen, Ravazzolo and Wulfsberg, 2014; Smets, Warne and Wouters, 2014; Kapetanios, Marcellino and Papailias, 2016; Marcellino, Porqueddu and Venditti, 2016; Carriero, Galvão and Kapetanios, 2019; Claveria, Monte and Torra, 2019). For their part, Ferrara, Marcellino and Mogliani (2015) developed GDP growth prediction models in 19 OECD countries. On the other hand, Stock and Watson (2003), and Kuzin, Marcellino and Schumacher (2013) investigated the prediction of GDP growth in developed economies, specifically Canada, France, Germany, Italy, Japan, the United Kingdom, and the United States.

From another point of view, taking into account the predictive variables, the most widely used indicators in the literature to forecast GDP growth have been the financial variables such as long-term interest rate on government bonds, short-term interest rate on government bonds, real effective exchange rate index, and S&P Index (Carriero, Galvão and Kapetanios, 2019; Carriero, Clark and Marcellino, 2019; Schorfheide and Song, 2015; Smets, Warne and Wouters, 2014; Stock and Watson, 2003). Also, various studies use other financial variables, such as broad money, money supply (Kapetanios, Marcellino and Papailias, 2016; Koop, 2013; Stock and Watson, 2003). Finally, recent literature also uses macroeconomic predictors, such as industrial production, exports and imports goods services, trade, labor force, and unemployed rate (Claveria, Monte and Torra, 2019; Barsoum and Stankiewicz, 2015; Ferrara, Marcellino and Mogliani, 2015; Kuzin, Marcellino and Schumacher, 2013). Among them, Stock and Watson (2003) determined that predictors as interest rate, long-term and short-term interest rate on government bonds, exchange rate, money supply, and broad money contain very useful information to forecast GDP growth.

Regarding the methods used, a considerable number of researchers have applied statistical methods for GDP growth forecasting, highlighting Vector autoregressions (Hansson, Jansson and Löf, 2005; Koop, 2013; Clark and Ravazzolo, 2015; Ferrara, Marcellino and Mogliani, 2015; Schorfheide and Song, 2015; Kapetanios, Marcellino and Papailias, 2016; Marcellino, Porqueddu and Venditti, 2016; Carriero, Galvão and Kapetanios, 2019), and Vector autoregression with stochastic volatility (Clark, 2011; Smets, Warne and Wouters, 2014; Diebold, Schorfheide and Shin, 2017; Carriero, Clark and Marcellino, 2019). On the other hand, the authors Barsoum and Stankiewicz (2015) and Kuzin, Marcellino and Schumacher (2013) developed Mixed-Data Sampling (MIDAS) regression models to forecast GDP growth. Among them, Carrierio, Clark and

Marcellino (2019) and Clark (2011) conclude that adding stochastic volatility to BVARs substantially improves the real-time accuracy of density forecasts, reducing mean squared errors. In turn, these models also improve the precision of point forecasts. For their part, Barsoum and Stankiewicz (2015) concluded that MIDAS models open new possibilities for researchers to use available data of different frequencies in forecasting and to address the possible problem of delays in the publication of macroeconomic variables. Some studies have used statistical models to assess trends in the macroeconomic situation through surveys (Batchelor and Dua, 1998; Martinsen, Ravazzolo and Wulfsberg, 2014). Finally, few previous studies have developed computational techniques. Claveria, Monte and Torra (2019), applied evolutionary computation, a branch of artificial intelligence, implementing evolutionary algorithms, to forecast the evolution of GDP. These authors concluded that using this technique, they significantly improve the precision of macroeconomic predictions.

Lastly, regarding the level of precision reached in GDP growth forecasting' literature, the range moves in 60–70% (Batchelor and Dua, 1992; Hansson, Jansson and Löf, 2005; Clark, 2011; Kuzin, Marcellino and Schumacher, 2013; Martinsen, Ravazzolo and Wulfsberg, 2014; Smets, Warne and Wouters, 2014; Ferrara, Marcellino and Mogliani, 2015; Schorfheide and Song, 2015). With a higher level of precision range (70-80%), we find the research by Barsoum and Stankiewicz (2015), Gambetti and Giannone (2013), Clark and Ravazzolo (2015), Kapetanios, Marcellino and Papailias (2016), Marcellino, Porqueddu and Venditti (2016); Diebold, Schorfheide and Shin (2017), Carriero, Galvão and Kapetanios (2019), Claveria, Monte and Torra (2019). Table 1 shows a summary of this literature. In this table we can highlight, on the one hand, the autoregression methodology, as the most used in recent previous literature and, on the other hand, the United States´prevalence as a sample of countries.

Table 1 Literature summary

Authors	Year	Countries	Methodology	Accuracy
Barsoum et al.	2015	United States	Vector Autoregression	72%
Batchelor and Dua	1992	United States	Statistical model	66%
Carriero, Galvão et al.	2019	U.S., Euro area, Japan	Bayesian Vector Autoregressive	80%

Carriero, Clark et al.	2019	United States	Vector Autoregression with stochastic volatility	78%
Clark	2011	United States	Vector Autoregression with stochastic volatility	69%
Clark and Ravazzolo	2015	United States	Bayesian Vector Autoregressive	72%
Claveria et al.	2019	Scandinavian economies	Evolutionary Computation	79%
Diebold et al.	2017	United States	Dynamic Stochastic General Equilibrium	72%
Ferrara et al.	2015	OECD economies	Vector Autoregressions	50%
Hansson et al.	2005	Sweden	Vector Autoregressions	68%
Kapetanios et al.	2016	Euro area	Bayesian Vector Autoregressive	75%
Kuzin et al.	2013	USA, Japon, Euro area	Panel Vector Autoregressions	68%
Marcellino et al.	2016	Euro area	Bayesian Vector Autoregressive	72%
Martinsen et al.	2014	Norway and Sweden	Statistical model	70%
Schorfheide and Song	2015	United States	Bayesian Vector Autoregressive	60%
Smets et al .	2014	Euro area	Dynamic Stochastic General Equilibrium	69%

2. Methodology

As previously stated, to solve the question of research we have used different methods in the construction of the GDP growth forecasting models. The use of different methods aims to achieve a high accuracy model, which is contrasted not only through a classification technique but by applying all those that have shown success in the previous literature. Specifically, this study applies Support Vector Regression Quantum Bat Algorithm, Quantum Boltzmann Machines, Quantum Neural Networks, on the side of Quantum Computing; Deep Recurrent Convolution Neural Network, Deep Belief Network, Deep Neural Decision Trees and Deep Learning Linear Support Vector Machines, on the side of Deep Learning methods. A synthesis of the methodological aspects of each of these classification techniques appears below. Besides, the method of analysis of the sensitivity of variables used in the present study, in particular, the method of Sobol (Satelli, 2002), which is necessary to determine the level of significance of the variables used in the forecasting of GDP growth is obtained.

In the present study, the eight sixteen-core Intel Core i7-10700 processor has been used as computing resources to make estimates. The code for the estimation of our methods has been performed by Python (3.8 version), with the support of the libraries such as NumPy, PyTorch, and QisKit to create the mathematical routines, Deep Learning algorithms, and Quantum processing, respectively.

2.1. Quantum Computing Methods

2.1.1 Support Vector Regression Quantum Bat Algorithm (SVRQBA)

Support Vector Regression (SVR) is defined from a non-linear mapping function, φ (x), is defined to draw the input data set, $\{(\mathrm{xi}, \mathrm{yi})\}_{\mathrm{i}=1}^{\mathrm{N}}$, into a high dimensional feature space. Then, there is a theoretical linear function, *f*, to formulate the non-linear relationships between input data and output data. The linear function, *f*, is the called SVR function, and is shown as Equation (1).

$$f(x) = W^T \varphi(x) + b \qquad (1)$$

where *f(x)* represents the forecasting values; φ (x) is the feature mapping function, non-linearly mapping the input space, *x*, into the feature space; the coefficients, *w* and *b* are determined by minimizing the empirical risk, as shown in Equation (2).

$$\mathrm{R_{emp}(f)} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{L}_{\epsilon} \left(\mathrm{Y_i}, \mathrm{W^T} \varphi(\mathrm{X_i}) + \mathrm{b}\right. \qquad (2)$$

where $L_{\epsilon}(y, f(x))$ is the ε-insensitive loss function as shown in Equation (3).

$$L_{\epsilon}(y, f(x)) = \begin{cases} |f(x) - y| - \varepsilon & if\ |f(x) - y| \geq \varepsilon \\ 0\ otherwise \end{cases} \qquad (3)$$

Also, $L_{\epsilon}(y, f(x))$ is used to search for an optimum hyperplane in the feature space, to maximize the distance dividing the training data into two subsets. Consequently, the SVR modeling problem could be explained as minimizing the overall errors, shown in Equation (4).

$$\min_{\omega, b, \Im^*, \Im} R_{\varepsilon}(\omega, \Im^*, \Im) = \frac{1}{2} w^T w + C \sum_{i=1}^{N} (\Im_i^* + \Im_i) \qquad (4)$$

Equation (4) is applied to penalize large weights, at the same time, to keep the flatness of *f(x)*. The second term penalizes training errors by means of the ε-insensitive loss function. *C* is a parameter to trade off of *f(x)* and *y*. Training errors under ε are denoted as $\Im_i^*$, while training errors above ε are denoted as $\Im_i$. Finally, the SVR regression function is obtained as Equation (5).

$$f(x) = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i)K(x_i, x) + b \quad (5)$$

where $K(x_i, x_j)$ is the kernel function, and its value can be estimated through the inner product of two vectors, α_i^*, α_i are calculated and named as Lagrangian multipliers, x_i and x_j, in the feature space, $\varphi(x_i)$and $\varphi(x_j)$, respectively, being $K(x_i, x_j) = \varphi(x_i).\varphi(x_j)$ the kernel function used.

The bat algorithm is a meta-heuristic algorithm for intelligent search (Moss and Sinha, 2003; Li, Geng, Wang and Hong, 2017). This algorithm follows the next steps, (1) Bat's position and velocity are initialized, and are tackled as the solution in problem space; (2) The optimal function value of the problem is estimated; (3) The volume and velocity of bat units are adjusted, and are converted towards optimal unit; (4) The optimal solution is reached. The bat algorithm involves global search and local search and with this last development, SVRQBA is obtained.

In global search, assume that the search space is with *d* dimensions, at the time, *t*, the *i*-th bat has its position, x_i^t, and velocity, v_i^t. At the time, $t + 1$, its position, x_i^{t+1}, and velocity, v_i^{t+1}, are updated as expressed Equations (6) and (7), respectively.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (6)$$

$$v_i^{t+1} = v_i^t + \left(x_i^t - x_*\right)F_i \quad (7)$$

where x_* is the global optimal solution, F_i is the frequency, as shown in Equation (8).

$$F_i = F_{min} + (F_{max} - F_{min})\beta \quad (8)$$

where $\beta \in [0, 1]$ is a random number, F_{max} and F_{min} are respectively the max frequency and the min frequency of the *i*-th bat at this moment.

In local search, once a solution is chosen in the global optimal solution, each bat would generate an alternative solution in the form of the random walk as expressed in Equation (9).

$$x_{new}(i) = x_{old} + \lambda A^t \quad (9)$$

where x_{old} is a random solution produced in optimal disaggregation, A^t is the average volume in the bat population, λ is a *D* dimensional vector in $[-1, 1]$.

2.1.2 Quantum Boltzmann Machines (QBM)

A Boltzmann Machine (BM) is a fully-connected graph of (N) binary units or neurons (Benedetti, Realpe-Gómez, Biswas, and Perdomo-Ortiz, 2017; Adachi and Henderson, 2015). These neurons can be visible (directly model some aspect of data distribution) or hidden (not tied to any aspect of the data distribution and used only for capturing features from the data distribution). Each network has 2^N possible states, and the probability of sampling a state $s = (s_1, \dots, s_N)$, the model is expressed as appears in Equation (10).

$$p(s) = \frac{e^{-E(s)}}{Z} \quad (10)$$

where Z is the partition function, and E is an energy function defined as appears in Equation (11).

$$E(s) = -\sum_{s_i \in S} b_i s_i - \sum_{s_i, s_j \in S} W_{ij}\, s_i s_j \quad (11)$$

where b expresses the linear "bias" on each unit and W delimits the "weight" of the coupling between two units (b and W will be denoted as model parameters). To calculate the model parameters for maximizing the log-likelihood (*L)*, the gradient descent method is applied and learning rate η to get model parameter update Equations (12) and (13).

$$\Delta W_{ij} = \frac{1}{\eta}\left[\left(s_i s_j\right)D - \left(s_i s_j\right)M\right] \quad (12)$$

$$\Delta b_i = \frac{1}{\eta}\left[(s_i)D - (s_i)M\right] \quad (13)$$

In equation 14, the values of $\left(s_i s_j\right)$ and (s_i) correspond to expectation values over the data (*D*) and model (*M*) distributions. The model would be trained if first, (s_i), and second-order moments, $\left(s_i s_j\right)$, were the same for both the data and model distribution. This algorithm can exploit quantum effects to take an initial quantum system that is in a ground state and become this into a final Hamiltonian, one in which the system should still be in the ground state to get QBM (Farhi, Goldstone, Gutmann, Lapan, Lundgren and Preda, 2001).

The target is to produce better approximations of $(s_i)M$ and $\left(s_i s_j\right)M$ than classical heuristics. This method seems more natural as the form of the Hamiltonian (*H*) is expressed in Equation (14).

$$H(S) = -\sum_{s_i \in S} h_i S_i - \sum_{s_i, s_j \in S} J_{ij}\, S_i S_j \quad (14)$$

where S is the vector of qubit spin states, h_i are the bias terms on each qubit, and J_{ij} are the (anti)ferromagnetic couplings between the qubits.

2.1.3 Quantum Neural Networks (QNN)

Wan, Dahlsten, Kristjánsson, Gardner and Kim (2017) showed the possibilities of combining Convolutional Neural Networks (CNN) unique computational capabilities and quantum computing. This combination can create a computational technique with great predictive potential. Qubit is defined as the smallest unit of information in quantum computation which is a probabilistic representation. A qubit may either be in the "1" or "0" or any superposition of the two (Gonçalves, 2019; Mahajan, 2011). The state of the qubit can be defined as follows in Equation (15):

$$|\psi\rangle = \alpha|o\rangle + \beta|1\rangle \quad (15)$$

where α and β are the numbers that point out the amplitude of the corresponding states such that $|\alpha|^2 + |\beta|^2 = 1.$ It is determined as a pair of numbers. $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ The angle θ is the specification that represents geometrical aspects and is defined such that: $\cos(\theta) = |\alpha|\ and\ \sin(\theta) = |\beta|$. Quantum gates may be applied for adjusting the probabilities as a result of weight upgrading (Mahajan, 2011; Zidan, Abdel-Aty, El-shafei, Feraig, Al-Sbou, Eleuch and Abdel-Aty, 2019). An example of a rotation gate can be as defined in Equation (16):

$$U(\Delta\theta) = \begin{bmatrix}\cos(\Delta\theta) & -\sin(\Delta\theta\\ \sin(\Delta\theta) & \cos(\Delta\theta)\end{bmatrix} \quad (16)$$

A state of the qubit can be upgraded by applying the quantum gate explained previously. Application of rotation gate on a qubit is defined as follows:

$$\begin{bmatrix}\alpha'\\ \beta'\end{bmatrix} = \begin{bmatrix}\cos(\Delta\theta) & -\sin(\Delta\theta\\ \sin(\Delta\theta) & \cos(\Delta\theta)\end{bmatrix}\begin{bmatrix}\alpha\\ \beta\end{bmatrix} \quad (17)$$

QNN is proposed for forecasting GDP growth. The process is begun with a quantum hidden neuron from the state $|o\rangle$, which prepares the superposition as indicated in Equation (18).

$$\sqrt{p}|O\rangle + \sqrt{1} - p|1\rangle\ with\ O \leq |p| \leq 1 \quad (18)$$

where p expresses the random probability of starting the system in the state $|O\rangle$. The classical neurons are inducted by random number generation (Mahajan, 2011; Zidan, Abdel-Aty, El-shafei, Feraig, Al-Sbou, Eleuch and Abdel-Aty, 2019). The output from the quantum neuron is determined as follows in Equation (19).

$$v_j = f\left(\sum_{i=1}^{n} w_{ji} \ * \ x_i\right) \tag{19}$$

where f is a problem-dependent sigmoid or Gaussian function (Mahajan, 2011). The output from the network is defined as appears in Equation (20).

$$y_k = f\left(\sum_{j=1}^{l} w_{jk} \ * \ v_j\right) \tag{20}$$

The desired output is the o_k corresponding squared error is represented in Equation (21).

$$E_k^2 = \frac{1}{2} \left|y_k - o_k\right|^2 \tag{21}$$

The learning uses the rules of the feed-forward backpropagation algorithm. The upgrading of output layer weight is defined in Equation (22).

$$\Delta w_{jk} = \eta e_k f' v_j \tag{22}$$

2.2. Deep Learning Methods

2.2.1. Deep Recurrent Convolution Neural Network (DRCNN)

Recurrent neural networks (RNN) have been successfully used in many fields for time-series prediction due to his huge prediction performance. The common structure of RNN is organized by the output of which is dependent on its previous computations (Wan, Dahlsten, Kristjánsson, Gardner and Kim, 2017). Given an input sequence vector x, the hidden states of a recurrent layer s, and the output of a single hidden layer y, can be calculated as follows in Equations (23) and (24).

$$s_t = \sigma(W_{xs} x_t + W_{ss} s_{t-1} + b_s) \tag{23}$$

$$y_t = o\left(W_{so} s_t + b_y\right) \tag{24}$$

Where W_{xs}, W_{ss}, and W_{so} denote the weights from the input layer x to the hidden layer s, the hidden layer to itself, and the hidden layer to its output layer, respectively. b_y are the biases of the hidden layer and the output layer. Equation (25) points out σ and o are the activation functions.

$$STFT\{z(t)\}(\tau,\omega)=\int_{-\infty}^{+\infty} z(t)\omega\big(t-\tau\big)e^{-j\omega t}dt\big) \quad (25)$$

where *z(t)* is the vibration signals, *ω(t)* is the Gaussian window function focused around 0. *T(τ, ω)* is a complex function that describes the vibration signals over time and frequency. To calculate the hidden layers with the convolutional operation Equations (26) and (27) are applied.

$$S_t = \sigma(W_{TS} *\ T_t + W_{SS} *\ S_{t-1} + B_s) \quad (26)$$

$$Y_t = o\big(W_{YS} *\ S_t + B_y\big) \quad (27)$$

where W term indicates the convolution kernels. The convolution is operated between weights and inputs and is performed in the transition of inputs to the hidden layers.

Recurrent Convolutional Neural Network (RCNN) can be heaped to establish a deep architecture, named deep recurrent convolutional neural network (Huang and Narayanan, 2017). When DRCNN is employed for prediction, the last part of the model is a supervised learning layer, which is determined as Equation (28).

$$\hat{r} = \sigma(W_h * h + b_h) \quad (28)$$

where W_h is the weight and b_h is the bias, respectively. The error between predicted observations and actual ones in the training data for the prediction can be calculated and backpropagated to train the model (Ma and Mao, 2019). Stochastic gradient descent is applied for optimization to learn the parameters. Considering that the actual data at time t is r, the loss function is determined as shown in Equation (29).

$$L(r,\hat{r}) = \frac{1}{2}\|r-\hat{r}\|_2^2 \quad (29)$$

2.2.2. Deep Belief Network (DBN)

The Deep Belief Network (DBN) is a kind of deep neural network where the top two layers are modeled as an undirected bipartite associative memory, that is, restricted Boltzmann machines (RBM) as expressed in Equation (30).

$$P(v, h^1, \ldots, h^l) = P(h^{l-1}, h^l)\left(\prod_{K=0}^{l-2} P\left(h^{k|}|h^{k+1}\right)\right) \tag{30}$$

where v is the vector of visible units, $P(h^{k-1}|h^k)$ is the conditional probability of visible units given the hidden ones in an RBM at level k. The joint distribution at the top level, $P(h^{l-1}, h)$, is a RBM, being $x(n) = [1, x_1(n), x_2(n), \ldots, x_m(n)]^T$. Another way to describe a DBN with a simpler model is shown in Equation (31).

$$w(n) = [b, w_1(n), w_2(n), \ldots, w_m(n)]^T \tag{31}$$

and explains why a DBN is a generative model. We can see that there are two types of arrows, point arrows, and solid arrows.

A DBN is made by stacking the RBMs on top of each other. The visible layer of each RBM in the stack is set to the hidden layer of the previous RBM. By learning a model for a data set, we want to find a model $Q(h^{l-1}, h)$ for the true posterior $P(h^{l-1}, h)$. The subsequent Q's are all approximations except the top-level $Q(h^{l-1}, h)$ posterior which is equal to the true posterior $P(h^{l-1}, h)$, where the top level RBM allows us to make an exact inference (Bengio, 2009).

2.2.3. Deep Neural Decision Trees (DNDT)

Deep neural decision trees are Decision Tree (DT) models executed by deep-learning neural networks, where an assignment of DNDT weightings corresponds to a specific decision tree and is thus interpretable (Yang, Garcia-Morillo, Hospedales, 2018). All parameters are optimized with stochastic gradient descent (SGD) instead of a complex greedy splitting process; this allows large-scale processing with mini-batch-based learning and can be linked to any larger Neural Network (NN) model for end-to-end learning with backward propagation. Besides, conventional DTs learn through a greedy and recursive division of characteristics (Quinlan, 1993). This may have benefits for the selection of functions; however, this greedy search may transform inefficient (Norouzi, Collins, Johnson, Fleet and Kohli, 2015). The algorithm starts by implementing a soft binning function to estimate the error rate for each node, making it possible to make decisions divided into DNDTs (Dougherty, Kohavi and Sahami, 1995). Generally, the input of a binning function is a real scalar x which produces an index of the containers to which x belongs. Supposing x is a continuous variable, group it into n+1 intervals. This requires n cut-off points which are trainable variables in this context. The cut-off points are denoted as $(\beta_1, \beta_2, \ldots, \beta_n)$ and are strictly ascending such that $\beta_1 < \beta_2 < \ldots < \beta_n$.

The activation function of the DNDT algorithm is implemented based on the NN defined in Equation (32).

$$\pi = fw, b, \tau(x) = softmax((wx + b)/\tau) \quad (32)$$

where *w* is a constant with value *w* = [1, 2, ..., *n* + 1], $\tau > 0$ is a temperature factor, and *b* is defined in Equation (33).

$$b = [0, -\beta_1, -\beta_1, -\ \beta_2 \ldots, -\beta_1 -\ \beta_2 - \cdots -\ \beta_{n]} \quad (33)$$

The NN defined in Equation (32) gives a coding of the binning function *x*. Additionally, if τ tends to 0 (often the most common case), the vector sampling is implemented using the Straight-Through (ST) Gumbel–Softmax method (Ho, 1998).

Given the binning function described above, the key idea is to build the DT using the Kronecker product. Assuming we have an input instance $x \in R^D$ with *D* characteristics. Associating each characteristic x_d with its NN $f_d(x_d)$, we can determine all the final nodes of the DT, in line with Equation (34).

$$z = f_1(x_1) \otimes f_2(x_2) \otimes \ldots \otimes f_D(x_D) \quad (34)$$

where *z* is a vector that points out the index of the leaf node reached by instance *x*. We assume that a linear classifier on each leaf *z* classifies the instances that reach it. The number of cut points per feature is the complexity parameter of the model. The cut-off point values are not limited, which means that some of them may be inactive. For example, they are smaller than the minimum x_d or greater than the maximum x_d.

2.2.4. Deep Learning Linear Support Vector Machines (DSVR).

Linear support vector machines (SVM) are designed for binary classification. Given training data and its corresponding labels $(x_n, y_n), n = 1, \ldots, N, x_n \in \mathbb{R}^D, t_n \in \{-1, +1\}$, SVMs learning is based on the constrained optimization defined in Equation (35).

$$\min_{w\xi_n} \frac{1}{2} W^T W + C \sum_{n=1}^{N} \xi_n \quad (35)$$

$$subject\ to. \quad W^T x_n t_n \geq 1 - \xi_n > \forall n$$

$$\xi_n \geq 0 \quad \forall n$$

where ξ_n are slack variables that penalize data points that violate the margin requirements. Note that we can include the bias by augmenting all data vectors x_n with a scalar value of 1. The corresponding unconstrained optimization problem is expressed in Equation (36).

$$\min_{w} \frac{1}{2} W^T W + C \sum_{n=1}^{N} max(1 - W^T x_n t_n, 0) \tag{36}$$

For classification problems using deep learning techniques, it is standard to use the softmax or 1-of-*K* encoding at the top. For example, given 10 possible classes, the softmax layer has 10 nodes denoted by p_i , where $i = 1, \ldots, 10$; p_i specifies a discrete probability distribution, therefore, $\sum_i^{10} p_i = 1$.

Let *h* be the activation of the penultimate layer nodes, *W* is the weight connecting the penultimate layer to the softmax layer, the total input into a softmax layer, given by Equation (37). Then, we obtained the Equation (38).

$$a_i = \sum_k h_k W_{ki} \tag{37}$$

$$p_i = \frac{exp(a_i)}{\sum_j^{10} exp(a_j)} \tag{38}$$

The predicted class $\hat{\imath}$ would be as follows in Equation (39).

$$\hat{\imath} = arg \max_i p_i = arg \max_i a_i \tag{39}$$

Since Linear-SVM is not differentiable, a popular variation is known as the DSVR, which minimizes the squared hinge loss as indicates in Equation (40).

$$\min_{w} \frac{1}{2} W^T W + C \sum_{n=1}^{N} max(1 - W^T x_n t_n, 0)^2 \tag{40}$$

The target of DSVR is to train deep neural nets for classification. Lower layer weights are learned by back-propagating the gradients from the top layer linear SVM. To do this, we need to differentiate the SVM objective concerning the activation of the penultimate layer. Let the goal in Equation (41) be *l(w)*, and the input *x* is replaced with the penultimate activation *h*.

$$\frac{\partial l(w)}{\partial h_n} = -C t_n w(\mathbb{I}\{1 > w^T h_t t_n\}) \tag{41}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function. Likewise, for the DSVR, we have Equation (42).

$$\frac{\partial l(w)}{\partial h_n} = -2Ct_n w\big(max(1 - W^T h_n t_n, 0)\big) \quad (42)$$

From this point on, the backpropagation algorithm is the same as the standard softmax-based deep learning networks.

2.3. Sensitivity analysis

Sensitivity analysis is used to show the determined impact of explanatory variables, allowing quantification of the relative significance of the independent variables related to the dependent variable (Delen**,** Kuzey, Uyar, 2013). The computational methods used in this study build an appropriate measure of significance. The sensitivity analysis is also used to reduce the models to the most significant variables, eliminating or ignoring those of lesser significance. A variable is considered more significant than another if it increases the variance compared to the set of variables of the model. Each model generates significance scores for each independent variable. This is done using the Sobol method (Saltelli, 2002), which decomposes the variance of the total output *V(Y)* in line with the equations expressed in Equation (43).

$$V(Y) = \sum_i V_i + \sum_i \sum_{j>1} V_{ij} + \cdots + V_{1,2,.,k} \quad (43)$$

Where $V_i = V\big(E(Y|X_i)\big)$ and $V_{ij} = V\left(E\big(Y|X_i, X_j\big)\right) - V_i - V$.

The sensitivity indexes are determined by $S_i = V_i/V$ and $S_{ij} = V_{ij}/V$, where S_{ij} indicates the effect of interaction between two factors. The Sobol decomposition allows the estimation of a total sensitivity index ST_i, which measures the sum of all the sensitivity effects involved in the independent variables.

3. Sample, data, and variables

3.1 Sample and data

The period selected is from 1980 to 2018 for a sample of 70 countries (47 emerging countries and 23 developed countries; see Appendix A-C), which has made it possible to build three forecasting models of GDP growth. The data of the dependent and independent variables have been obtained from the IMF's International Financial Statistics (IFS) and the World Bank.

The sample data set has been divided into three groups mutually exclusive, one for training (70% of the data), another for validation (10% of the data), and the third group for testing (20% of the data). The training data is used initially to build the models. As is well known, the validation data is used to evaluate the methods during training, and to detect an over-training of it. If the error for the validation grows during a certain number of training times, the training is stopped. For its part, the testing data is used to evaluate the built model and make forecasts. The percentage of correctly classified cases (accuracy) and the root of the mean square error have been used for the evaluation. Furthermore, for the treatment of each of the three groups, the 10-fold cross-validation procedure has been applied with 500 iterations (Alaminos, Fernández, García and Fernández, 2018; Zhang and Qi, 2005; Wu, et al., 2019; Zhang, Li and Chen, 2020; Pesantez-Narvaez, Guillen, and Alcañiz, 2020). These three groups of data are used independently in order not to bias the results of each stage with possible data previously used. Therefore, we can observe the ability of success in the prediction of the models through the test results. This scheme for evaluating the precision of the built models has been followed by other recent previous works (Zhao, Li and Yu, 2017; Moews, Herrmann and Ibikunle, 2019; Zhong and Enke, 2019; Ghoddusi, Creamer and Rafizadeh, 2019; Henrique, Sobreiro and Kimura, 2019; Wu, et al., 2019; Zhang, Li and Chen, 2020; Sanhudo et al., 2020)

The precision capacity has been calculated from the mean of success of the 500 iterations carried out for each model and expressed as a percentage. (Seng, Ang, Schmidtke and Rogiers, 2018; Zhong and Enke, 2019; Moews, Herrmann and Ibikunle, 2019; Henrique, Sobreiro and Kimura, 2019; Montano et al., 2020; Sanhudo et al., 2020), while the RMSE values are estimated through the expression 44 (Reyes et al., 2010; Pesantez-Narvaez, Guillen, and Alcañiz, 2020).

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\widehat{y_t} - y_t)^2}{T}} \quad (44)$$

where is the real observation and is the observation predicted by the model, at time t and for T prediction periods.

Tables 2 shows the independent variables used to forecast GDP growth. We use 24 independent variables for GDP growth as possible predictors. These are standard variables used in the existing literature and they are classified according to macroeconomic and financial attributes.

Table 2. Independent variables for GDP growth

Code	Description	Source
Financial variables		
IRL	Long-Term interest rate on government bonds	Camba-Mendez et al (2001)
IRS	Short-Term interest rate on government bonds	Kapetanios, Marcellino and Papailias (2016)
RIR	Real Interest Rate (%)	Camba-Mendez et al. (2001)
RER	Real Effective Exchange Rate Index (2010=100)	Carriero, Galvao and Kapetanios (2019)
M2	Broad Money (% of GDP)	Koop (2013)
M2G	Broad Money Growth (annual %)	Koop (2013)
M1	Money Supply (2015=100)	Kapetanios, Marcellino and Papailias (2016)
VIX	CBOE Volatility Index	Kuzin, Marcellino and Schumacher (2013)
SPI	S&P 500 Index	Carriero, Galvao and Kapetanios (2019)
Macroeconomic variables		
IP	Industrial production (% GDP)	Barsoum and Stankiewicz (2015)
X	Exports Goods Services (% of GDP)	Claveria, Monte and Torra (2019)
M	Imports Goods Services (% of GDP)	Claveria, Monte and Torra (2019)
T	Trade (% of GDP)	Claveria, Monte and Torra (2019)
GFCF	Gross Fixed Capital Formation (% of GDP)	Ferrara, Marcellino and Mogilani (2015)
GCE	General Government Final Consumption Expenditure (% of GDP)	Ferrara, Marcellino and Mogilani (2015)
LF	Labor force	Carriero, Clark and Marcellino (2019)
EPR	Employment to Population Ratio, 15+, Total (%)	Carriero, Clark and Marcellino (2019)
UE	Unemployment (%Total Labor Force)	Carriero, Galvao and Kapetanios (2019)
CE	Compensation Employees (current LCU)	Carriero, Clark and Marcellino (2019)
EP	Employees Non-farm Payrolls	Koop (2013)
ICP	Inflation Consumer Prices (annual %)	Ferrara, Marcellino and Mogilani (2015)

CPI	Consumer Price Index (2010 = 100)	Ferrara, Marcellino and Mogilani (2015)
OP	Oil Price	Carriero, Galvao and Kapetanios (2019)
GP	Gold Price	Carriero, Galvao and Kapetanios (2019)

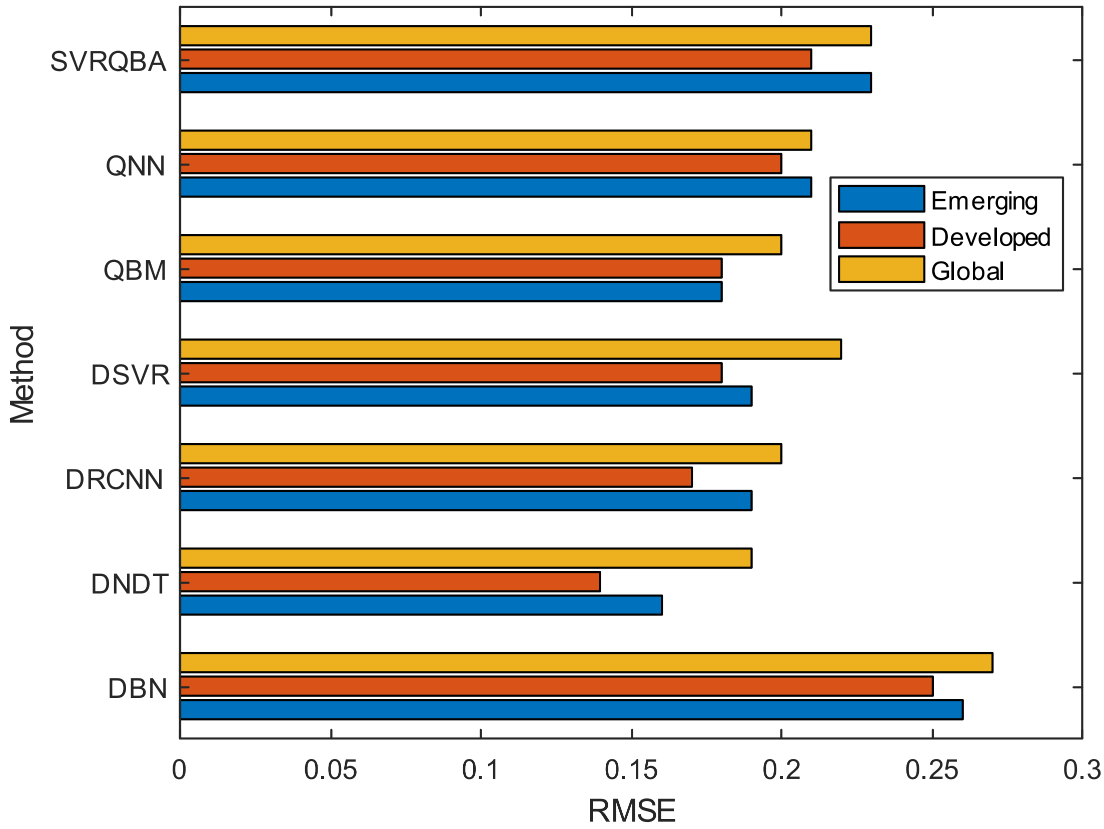
4. Results

Tables 3 shows the accuracy level and the significant variables of each of the methods by a group of countries for forecasting GDP growth. In emerging, developed, and global countries, the highest accuracy values for each model occur in the DNDT method, followed by the DRCNN one, with a range of 98,29%-96,02%. Figure 1compares the Root Mean Square Error (RMSE) obtained in the forecasting of GDP growth of each methodology in the samples described. For the values of emerging countries, the lowest RMSE value is in the DNDT method, followed by the QBM method. However, in the case of developed countries and global sample, the lowest RMSE value is also firstly in the DNDT method, but secondly the DRCNN method. In all cases, accuracy is higher than 93.02% for forecasting GDP growth. Also, the RMSE values are adequate, if we compare it with the results of errors obtained by previous work in macroeconomic forecasting (Salas, Alaminos, Fernández and López-Valverde, 2020).

Table 3. Results of estimated models

Method	Classification (%)			Greater sensitivity variables
	Training	Validation	Testing	
Emerging countries				
SVRQBA	95.26	94.05	93.12	RER, M2G, VIX, SPI, X, GFCF
QNN	96.52	96.31	95.53	RER, M2G, VIX, SPI, T, CPI
QBM	97.14	96.89	95.15	RIR, RER, M2, VIX, SPI, T, CPI
DBN	93.51	93.02	92.57	RIR, M2, SPI, IP, X, T, GCE
DRCNN	97.27	96.83	96.32*	IRL, RER, M2, VIX, SPI, X, CPI
DNDT	97.92	97.68	96.83*	IRL, RER, M2G, VIX, SPI, T, CPI
DSVR	96.83	96.65	96.02*	RER, M2G, VIX, SPI, T, GFCF, CPI
Developed countries				
SVRQBA	96.78	96.42	95.74	IRS, RIR, M2, VIX, IP, GFCF, LF
QNN	97.66	97.23	96.38	IRS, RIR, M2, IP, T, GFCF, UE
QBM	98.35	98.12	97.61*	RIR, M2G, T, GFCF, GCE, OP
DBN	95.08	94.46	93.67	IRS, M2G, VIX, IP, T, GCE, UE
DRCNN	98.62	98.37	97.75*	RIR, M2, IP, T, GFCF, CPI, OP
DNDT	99.21	98.95	98.29*	RIR, M2G, IP, T, GFCF, GCE, CPI, OP
DSVR	98.39	98.16	97.48	RIR, M2G, IP, T, GFCF, CPI, OP
Global countries				
SVRQBA	95.98	95.64	95.15	IRS, RER, M2, VIX, X, GFCF, OP
QNN	97.13	96.87	96.20	IRL, RER, M2, VIX, SPI, GCE
QBM	98.14	97.73	97.11*	RIR, RER, M2G, SPI, T, GFCF, CPI
DBN	94.23	93.64	93.22	RIR, RER, M2, SPI, T, GFCF, CPI
DRCNN	98.04	97.71	97.18*	IRL, RIR, M2G, VIX, T, GFCF, CPI
DNDT	98.27	98.03	97.63*	IRL, RER, M2G, SPI, T, GCE, CPI, OP
DSVR	97.49	97.20	96.57	IRL, RER, M2G, VIX, T, GFCF, OP

Figure 1. RMSE values



At the same time, Figures 2, 3 and 4 show additional information on the significant variables. The greater sensitivity variables in the models for forecasting GDP growth have been SPI, RER, and VIX in emerging countries, repeated in seven and six models; RIR, IP, GFCF, and T in developed countries, reiterated in six models; and RER, GFCF, and T in global, repeated in six and five models. The variables VIX, M2G, GFCF, T, CPI, and M2 have been reiterated in all three country samples. For example, VIX has been a significant variable in emerging countries in all methods except DBN. In developed countries, RIR has been a meaningful variable, also in all the methodologies, excluding the DBN one. Finally, in the global sample of countries, RER has been highly relevant in all methods, save for the DRCNN.

Figure 2. Number of repetitions of the greater sensitivity variables by Quantum methods

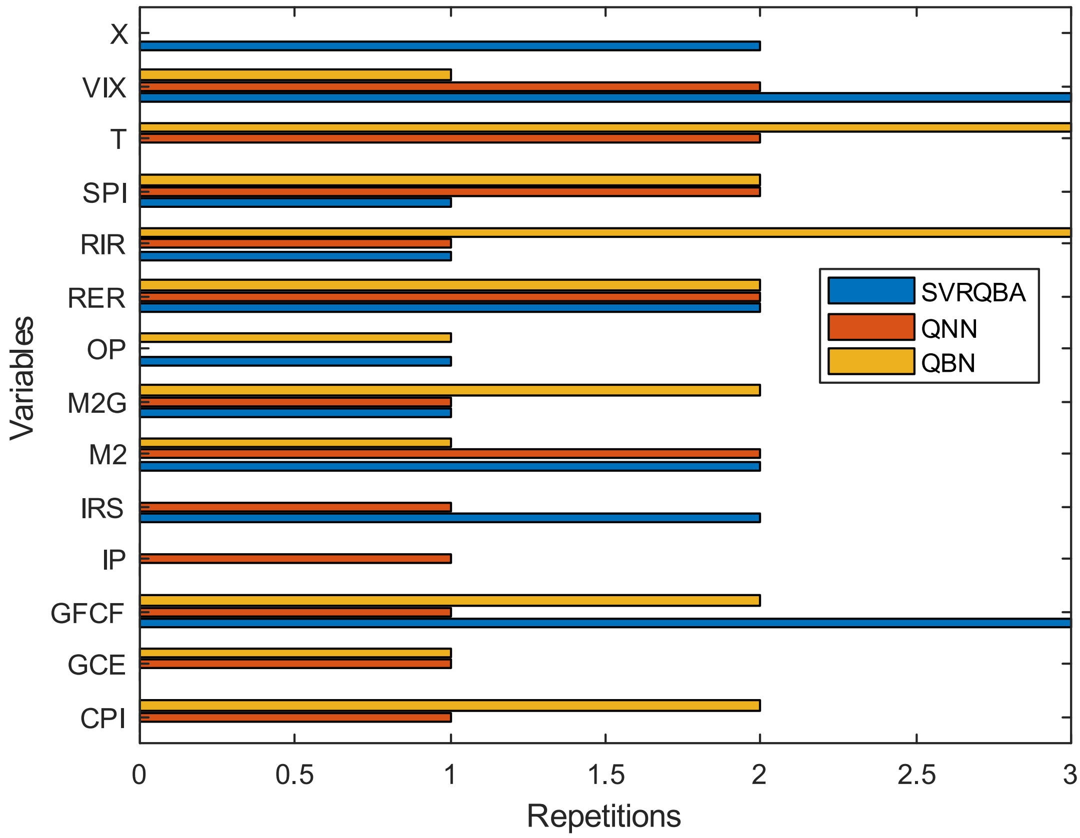


Figure 3. Number of repetitions of the greater sensitivity variables by Deep Learning methods

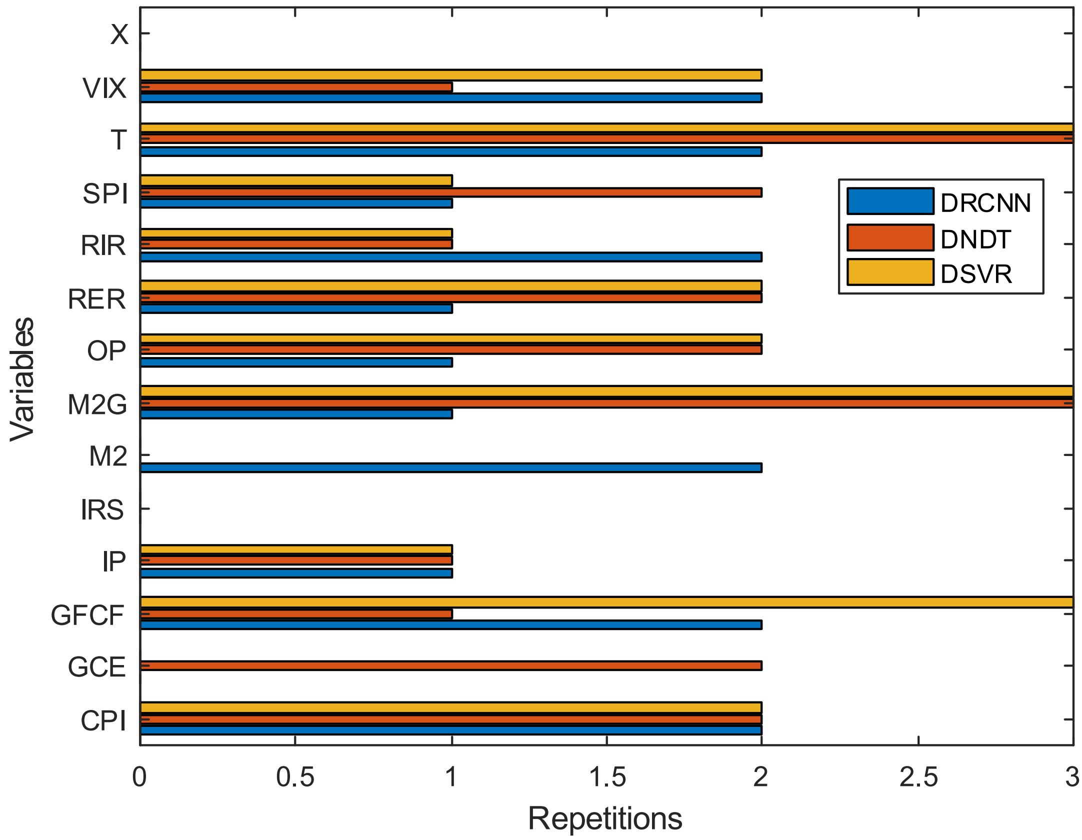
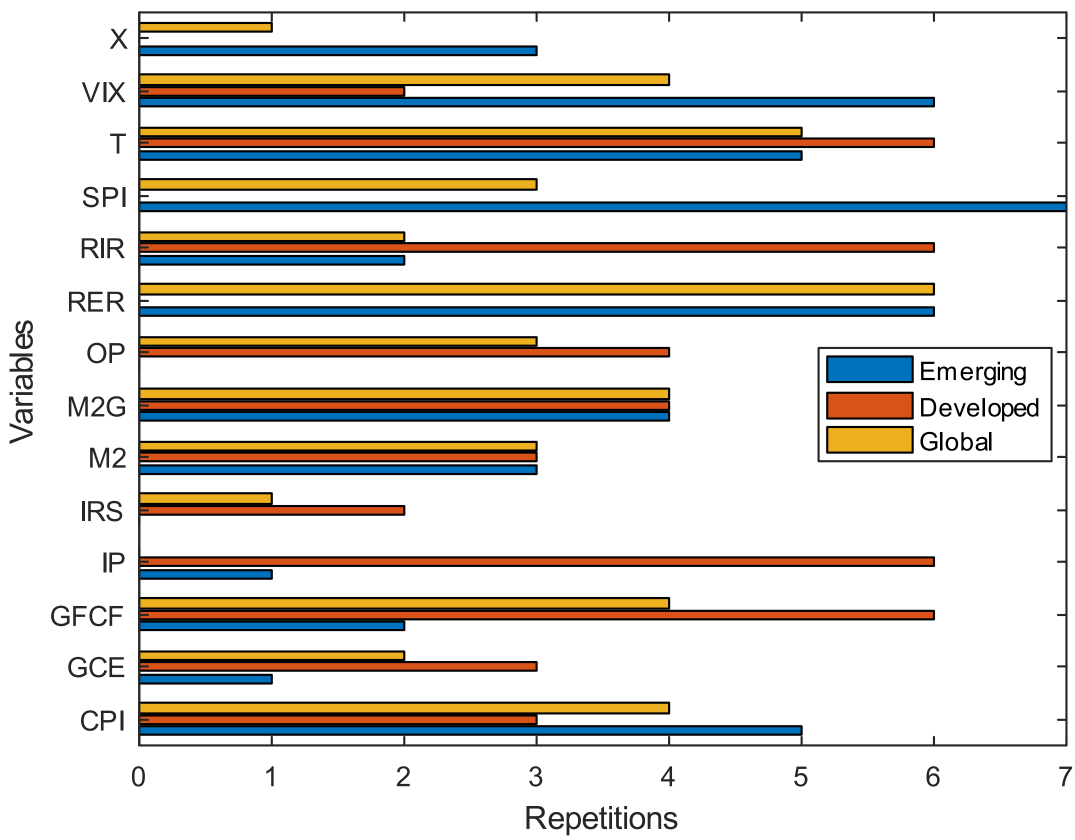


Figure 4. Number of repetitions of the greater sensitivity variables by sample



The best results obtained for predicting GDP growth have been in the method of DNDT and show that, for developed countries, RIR, M2G, IP, T, GFCF, GCE, CPI, and OP are the significant variables. In comparison with previous studies, the variables RIR, IP, CPI were significant in the work of Carriero, Clark and Marcellino (2019). The study of Marcellino, Porqueddu and Venditti (2016) concluded that RIR, RER, IP have significant effects on the GDP growth forecasting. For its part, Martinsen, Ravazzolo and Wulsberg (2014) found UE, IP, and SPI are crucial to the production of accurate forecasts of Norwegian GDP. All the above confirms that our research has validated new significant variables (M2G, T, GFCF, GCE, and OP), detecting a new set of relevant variables different from what was shown in previous studies. M2G is positively related to GDP growth, an increase in M2G should lower the interest rates in the economy, leading to more consumption and lending/borrowing. So, in the short run, this should correlate to an increase in total GDP. There is also a correlation between GDP growth and T (countries with higher rates of GDP growth also tend to have higher rates of trade growth). At the same time, GFCF has a positive relationship with economic growth in the short as well as in the long-run.

For its part, the results for emerging economies show that IRL, RER, M2G, VIX, SPI, T, and CPI are the most significant variables to predict GDP growth. Therefore, emerging countries should be on alert regarding the behaviour of these variables. Given that there are no previous studies that develop forecasts expressly for emerging countries, the results of this research represent a novel contribution to the foreign GDP growth literature.

Finally, after observing the results of the global models, it is deduced that the most significant variables to predict GDP growth are IRL, RER, M2G, SPI, T, GCE, CPI, and OP. There are also no previous studies on GDP growth forecasting globally to be able to compare with our results. However, we can understand that our research provides new significant variables to make forecasts in any country.

From precision levels, the accuracy results of three estimated models for forecasting GDP growth are higher than 93.02%. Taken together, these results provide a level of accuracy far superior to that of previous studies. Thus, in the work of Carriero, Clark and Marcellino (2019), an accuracy of around 78% is revealed; in the case of Claveria, Monte and Torra (2019) it is close to 79%; and in the study of Carriero, Galvão and Kapetanios (2019), it approaches 80%.

If we compare the accuracy among our methodological techniques, we can conclude that the model with the highest accuracy in GDP growth forecasting is that of the DNDT method, with 98.95% in developed countries, followed by the model of DNDT method with 98.03% in global sample, and come next also by the model of DNDT method with 97.68% in emerging countries. Therefore, the method with the best levels of precision to predict the GDP growth is DNDT in the three models. In the second position, QBM in emerging and global sample and DRCNN in developed countries. In the third position, DRCNN for emerging and global and DSVR for developed countries. In conclusion, deep learning methods predict macroeconomic forecasting better than quantum computing methods, although the QBM method has obtained very good results.

4.1 Post-estimations

Using multiple-step-ahead forecasting, we have considered the iterative strategy, where models that are trained for the forecasting of 1-step forward are developed (Koprinska, Rana and Rahman, 2019). At time *t*, forecasting is made for moment *t*+1, and this forecasting is used to make the forecasting for moment *t*+2 and so on. This means that the predicted data for *t*+1 are considered real data and are added to the end of the available data (Makridakis, Spiliotis and Assimakopoulos, 2018). Table 4 shows the accuracy and error results for *t*+1 and *t*+2 and *t*+3 forecasting horizons. For *t*+1, the highest accuracy values for each model occur in the DNDT method, with a range of 97.28-94.80%. The highest accuracy occurs in the models for developed countries. The highest level of accuracy for *t*+2 is also for developing countries in the DNDT method (96.81%). In this case, the accuracy range is 96.81-89.59%. The accuracy range for *t*+3 is 93.81-86.88%, being again in the model of DNDT where the percentage of accuracy is higher (93.81%). These results show that high precision and great stability of the models have been achieved.

Table 4. Multiple-step ahead forecasts and RMSE score

Method	Classification (%)			RMSE		
	t+1	t+2	t+3	t+1	t+2	t+3
Emerging countries						
SVRQBA	92.40	91.62	89.10	0.26	0.28	0.32
QNN	93.03	91.93	89.75	0.23	0.27	0.30
QBM	93.43	92.52	90.02	0.22	0.26	0.29
DBN	90.16	89.76	86.68	0.29	0.31	0.34
DRCNN	95.29*	93.72*	91.01	0.21	0.24	0.27
DNDT	94.80*	93.22*	91.38*	0.22	0.24	0.27
DSVR	94.38	93.03	90.15*	0.23	0.25	0.28
Developed countries						
SVRQBA	94.37	93.39	89.52	0.26	0.28	0.31
QNN	95.81	94.45	90.64	0.23	0.25	0.29
QBM	95.85	94.26	91.17	0.22	0.24	0.28
DBN	92.37	91.17	88.76	0.29	0.31	0.33
DRCNN	96.21*	95.33*	93.02*	0.22	0.22	0.26
DNDT	97.28*	96.81*	93.45*	0.21	0.23	0.26
DSVR	96.01	94.16	92.10	0.23	0.25	0.27
Global countries						
SVRQBA	94.60	92.90	90.15	0.26	0.29	0.31
QNN	95.67	94.68	91.07	0.24	0.25	0.29
QBM	95.51	94.16	90.79	0.22	0.24	0.30
DBN	91.28	89.59	87.44	0.30	0.32	0.34
DRCNN	95.72	94.61	92.79*	0.22	0.26	0.29
DNDT	96.59*	95.84*	93.81*	0.23	0.25	0.28
DSVR	96.14*	94.86*	92.58	0.23	0.26	0.29

5. Conclusions

This study has developed a comparison of methodologies to predict GDP growth and, therefore, new models have been generated for GDP growth forecasting. The period selected is from 1980 to 2018 for a sample of 70 countries (emerging, develop, and a global sample of countries). It has been applied different methods in the construction of the GDP growth forecasting model to achieve a high accuracy model, such as SVRQBA, QBM, and QNN in Quantum computing, and DRCNN, DBN, DNDT, and DSVR in Deep learning. The DNDT model is the one that has obtained the highest levels of precision. Specifically, the goal has been to improve the predictive accuracy of previous studies using different methodologies and increase the sample size to all countries in the world. The results obtained in this research are significantly higher than those obtained in the existing literature, with an accuracy range of 93.02–98.95% for forecasting GDP growth. It has also detected new significant variables to consider in GDP growth forecasting models, allowing a high level of stability in the models developed over forecasting horizons of t+1, t+2, and t+3.

In contrast to previous research, this study has been able to expand GDP growth forecasts beyond developed countries to the global level. The results have identified different significant variables for emerging and developed economies, as well as at the global level. Our study suggests new explanatory significant variables to allow these agents to predict GDP growth forecasting. This research compares also, from a macroeconomic forecasting perspective, alternative models using seven methods, being the DNDT the most accurate, thus, contributing to existing knowledge in the field of macroeconomics forecasting. This new model can be used as a reference for setting macroeconomic policy and improved decision-making.

In summary, this study provides a significant opportunity to contribute to the field of GDP growth forecasting, since the results obtained have significant implications for the future decisions of political agents, making it possible to develop a clear improvement in the analysis of the economic situation. The conclusions are relevant to central bankers, investors, policymakers, private forecasters, and business professionals for economic policy in any country in the world, who are generally interested in knowing which indicators provide reliable, accurate, and potential forecasts of GDP growth. So, policymakers need to know GDP growth forecasting to support policy decisions.

Further research in this field includes developing predictive models considering political factors that evaluate the possible influence of the management and effectiveness of the economic policy on the phenomenon of GDP growth. Also, although in our study Deep Learning methods improve Quantum Computing methods, the high precision also of Quantum Computing techniques shows an interesting trend for its development, and upcoming studies in this field could be evaluated.

Appendix

See Tables 5, 6 and 7

Table 5. Sample of developed countries

Australia	Greece	Norway
Austria	Iceland	Portugal
Belgium	Ireland	Spain
Canada	Italy	Sweden
Denmark	Japan	Switzerland
Finland	Luxembourg	United Kingdom
France	Netherlands	United States
Germany	New Zealand	

Table 6. Sample of emerging countries

Argentina	Honduras	Panama
Bangladesh	Hungary	Paraguay
Bulgaria	India	Peru
Bolivia	Indonesia	Philippines
Brazil	Israel	Poland
Chile	Jordan	Romania
China	Kenya	Russian Federation
Colombia	Korea, Rep,	Slovenia

Costa Rica	Latvia	South Africa
Croatia	Lithuania	Thailand
Czech Republic	Malaysia	Tunisia
Ecuador	Malta	Turkey
Egypt	Mexico	Uruguay
El Salvador	Morocco	Venezuela
Estonia	Nigeria	Vietnam
Guatemala	Pakistan	

Table 7. Global sample

Argentina	Honduras	Panama
Australia	Greece	Norway
Austria	Iceland	Portugal
Bangladesh	Hungary	Paraguay
Belgium	Ireland	Spain
Bolivia	Indonesia	Philippines
Brazil	Israel	Poland
Bulgaria	India	Peru
Canada	Italy	Sweden
Chile	Jordan	Romania
China	Kenya	Russian Federation
Colombia	Korea, Rep,	Slovenia
Costa Rica	Latvia	South Africa
Croatia	Lithuania	Thailand
Czech Republic	Malaysia	Tunisia
Denmark	Japan	Switzerland
Ecuador	Malta	Turkey
Egypt	Mexico	Uruguay
El Salvador	Morocco	Venezuela

Estonia	Nigeria	Vietnam
Finland	Luxembourg	United Kingdom
France	Netherlands	United States
Germany	New Zealand	
Guatemala	Pakistan	

Funding The authors did not receive support from any organization for the submitted work.

Declarations

Confict of interest The authors declare that there is no confict of interest.

Ethical Approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Adachi, S.H. and Henderson, M.P. (2015). Application of Quantum Annealing to Training of Deep Neural Networks. *ArXiv eprints*, 1510.06356.

Alaminos, D. Fernández, S.M., García, F. Fernández, M.A. (2018). Data Mining for Municipal Financial Distress Prediction, Advances in Data Mining, Applications and Theoretical Aspects. Lecture Notes in *Computer Science*, 10933, 296-308. https://doi.org/10.1007/978-3-319-95786-9_23.

Barsoum, F. and Stankiewicz, S. (2015). Forecasting GDP growth using mixed-frequency models with switching regimes. *International Journal of Forecasting,* 31, 33-50. Doi: 10.1016/j.ijforecast.2014.04.002.

Batchelor, R. and Dua, P. (1992). Survey Expectations in the time series consumption function. *The Review of Economics and Statistics*, 74, 598-606.

Batchelor, R. and Dua, P. (1998). Improving macro-economic forecasts: The role of consumer confidence. *International Journal of Forecasting*, 14, 71–81. PII S0169-2070(97)00052-6.

Benedetti, M., Realpe-Gómez, J., Biswas, R. and Perdomo-Ortiz, A. (2017). Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models. *Phys.Rev*. X 7. DOI: 10.1103/PhysRevX.7.041052.

Benedetti, M., Realpe-Gómez, J., Biswas, R. and Perdomo-Ortiz, A. (2016). Estimation of effective temperaturas in quantum annealers for sampling applications: A case study with possible applications in deep learning. *Phys. Rev*. A 92. DOI: https://doi.org/10.1103/PhysRevA.4,022308.

Bengio, Y. (2009) Learning Deep Architectures for Artificial Intelligence. Foundations and Trends in Machine Learning, 2 (1): 1-127.

Bergström, R. (1995). The relationship between manufacturing production and different business survey series in Sweden 1968-1992. *International Journal of Forecasting*, 11, 379-393. SSDI: 0169-2070(95)00601-X.

Carriero, A., Clark, T.E. and Marcellino, M. (2019). Large Bayesian vector autoregressions with stochastic volatility and non-conjugate priors. *Journal of Econometrics*, 212, 137-154. https://doi.org/10.1016/j.jeconom.2019.04.024.

Carriero, A., Galvão, A.B. and Kapetanios, G. (2019). A comprehensive evaluation of macroeconomic forecasting methods. *International Journal of Forecasting*, 35, 1226–1239. https://doi.org/10.1016/j.ijforecast.2019.02.007.

Camba-Mendez, G., Kapetanios, G., Smith, R.J. and Weale, M.R. (2001). An automatic leading indicator of economic activity: forecasting GDP growth for European countries. *Econometrics Journal*, 4, S56–S90.

Claveria, O., Monte, E. and Torra, S. (2019). Evolutionary Computation for Macroeconomic Forecasting. *Computational Economics*, 53 (21), 833-849. https://doi.org/10.1007/s10614-017-9767-4.

Clark, T,E. (2011). Real-Time Density Forecasts From Bayesian Vector Autoregressions With Stochastic Volatility. *Journal of Business & Economic Statistics*, 29 (3), 327-341. DOI: 10.1198/jbes.2010.09248.

Clark, T.E. and Ravazzolo, F. (2015). Macroeconomic Forecasting Performance under alternative specifications of time-varying volatility. *Journal of Applied Econometrics*, 30, 551–575. DOI: 10.1002/jae.2379.

Clements, M.P. and Galvão, A.B. (2008) Macroeconomic Forecasting With Mixed-Frequency Data. *Journal of Business & Economic Statistics*, 26 (4), 546-554. DOI: 10.1198/073500108000000015.

Delen, D., Kuzey, C. and Uyar, A. (2013). Measuring firm performance using financial ratios: A decision tree approach. *Expert Systems with Applications*, 40, 3970–3983. https://doi.org/10.1016/j.eswa.2013.01.012.

Diebold, F.X., Schorfheide, F. and Shin, M. (2017). Real-time forecast evaluation of DSGE models with stochastic volatility. *Journal of Econometrics*, 201, 322-332. doi: 10.1016/j.jeconom.2017.08.011.

Dougherty, J., Kohavi, R. and Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, 9–12 July 1995.

Farhi, E., Goldstone, J., Gutmann, S., Lapan, J., Lundgren, A. and Preda, D. (2001). A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem. *Science*, 292 (5516), 472-475. DOI: 10.1126/science.1057726.

Ferrara, L., Marcellino, M. and Mogliani, M. (2015). Macroeconomic Forecasting during the Great Recessions: The return of non-linearity? *International Journal of Forecasting*, 31, 664-679. DOI: https://doi.org/10.1016/j.ijforecast.2014.11.005

Ghoddusi, H., Creamer, G.G., and Rafizadeh, N. (2019). Machine learning in energy economics and finance: A review. *Energy Economics*, 81(C), 709-727. DOI: https://doi.org/10.1016/j.eneco.2019.05.006

Gonçalves, C.P.S. (2019). Quantum Neural Machine Learning: Theory and Experiments, Chapter 5, Artificial Intelligence-Applications in Medicine and Biology. IntechOpen, London (2019). DOI: 10.5772/intechopen.84149.

Hansson, J., Jansson, P. and Löf, M. (2005). Business survey data: Do they help in forecasting GDP growth? *International Journal of Forecasting*, 21, 377 – 389. DOI: 10.1016/j.ijforecast.2004.11.003.

Henrique, B.M., Sobreiro, V.A., and Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. *Expert Systems with Applications*, 124, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012

Ho, T.K. (1998) The random subspace method for constructing decision forests, IEEE Trans, Pattern Anal, Mach, Intell, 20, 832–844.

Huang, C.W. and Narayanan, S.S. (2017). Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition. IEEE International Conference on Multimedia and Expo (ICME).

Kapetanios, G., Marcellino, M. and Papailias, F. (2016). Forecasting inflation and GDP growth using heurtisic optimisation of information criteria and variable reduction methods. Computational Statistics & Data Analysis, 100, 369-382. doi:10.1016/j.csda.2015.02.017.

Koop, G.M. (2013). Forecasting with Medium and Large Bayesian VARs. *Journal of Applied Econometrics*, 28, 177–203. DOI: 10.1002/jae.1270.

Koprinska, I., Rana, M. and Rahman, A. (2019). Dynamic ensemble using previous and predicted future performance for Multi-step-ahead solar power forecasting. ICANN 2019: *Artificial Neural Networks and Machine Learning*, 11730, 436-449. doi:10.1007/978-3-030-30490-4_35.

Kuzin, V., Marcellino, M. and Schumacher. (2013). Pooling versus model selection for nowcasting GDP with many predictors: empirical evidence for six industrialized countries. *Journal of applied econometrics*, 28, 392–411. DOI: 10.1002/jae.2279.

Li, M.W., Geng, J., Wang, S., and Hong, W.C. (2017). Hybrid Chaotic Quantum Bat Algorithm with SVR in Electric Load Forecasting. *Energies*, 10, 2180.

Ma, M. and Mao, Z. (2019). Deep Recurrent Convolutional Neural Network for Remaining Useful Life Prediction. IEEE International Conference on Prognostics and Health Management (ICPHM).

Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13 (3), e0194889. doi:10.1371/journal.pone.0194889.

Mahajan, R.P. (2011). A Quantum Neural Network Approach for Portfolio Selection. *International Journal of Computer Applications*, 29(4), 47-54.

Marcellino, M. (2008). A Linear Benchmark for Forecasting GDP Growth and Inflation. *Journal of Forecasting,* 27, 305–340 (2008). DOI: 10.1002/for.1059.

Marcellino, M., Porqueddu, M. and Venditti, F. (2016). Short-Term GDP Forecasting with a Mixed-Frequency Dynamic Factor Model with Stochastic Volatility. J*ournal of Business & Economic Statistics*, 34:1, 118-127. DOI: 10.1080/07350015.2015.1006773.

Martinsen, K., Ravazzolo, F. and Wulsberg, F. (2014), Forecasting macroeconomic variables using disaggregate survey data. *International Journal of Forecasting*, 30, 65–77. Doi: 10.1016/j.ijforecast.2013.02.003.

Moews, B., Herrmann, J.M., and Ibikunle, G. (2019). Lagged correlation-based deep learning for directional trend change prediction in financial time series*. Expert Systems with Applications*, 120, 197–206. https://doi.org/10.1016/j.eswa.2018.11.027

Montano, I.H., Marques, G., Alonso, S.G. et al. (2020). Predicting Absenteeism and Temporary Disability using Machine Learning: a Systematic Review and Analysis. *Journal of Medical Systems*, 44, 162. https://doi.org/10.1007/s10916-020-01626-2

Moss, C.F. and Sinha, S.R. (2003). Neurobiology of echolocation in bats. *Curr, Opin, Neurobiol*, 13, 751–758. DOI:10.1016/j.conb.2003.12.001.

Norouzi, M., Collins, M.D., Johnson, M., Fleet, D.J. and Kohli, P. (2015). Efficient non-greedy optimization of decision trees. In Advances in Neural Information Processing Systems 28 (NIPS 2015); The MIT Press, Cambridge, MA, USA, 2015.

Pesantez-Narvaez, J., Guillen, M., and Alcañiz, M.A. (2020). A Synthetic Penalized Logitboost to Model Mortgage Lending with Imbalanced Data. *Computational Economics*, published online. https://doi.org/10.1007/s10614-020-10059-5

Quinlan, J.R. (1993). C4,5: Programs for Machine Learning, Morgan Kaufmann PublishersInc,: Burlington, MA, USA, 1993.

Reyes, C, Hilaire, T., Paul, S., and Mecklenbräuker, C.F. (2010). Evaluation of the root mean square error performance of the PAST-Consensus algorithm. *IEEE*, 2010 International ITG Workshop on Smart Antennas (WSA), Bremen, 156-160. doi: 10.1109/WSA.2010.5456452

Salas, M.B., Alaminos, D., Fernández, M.A., López-Valverde, F. (2020). A global prediction model for sudden stops of capital flows using decision trees. *PLOS ONE*, 15(2): e0228387. Doi: 10.1371/journal.pone.0228387.

Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communications*, 145, 280–297. https://doi.org/10.1016/S0010-4655(02)00280-1.

Sanhudo, L., Calvetti, D., Martins, J.P., Ramos, N.M.M., Mêda, P., Gonçalves, M.P, and Sousa, H. (2020). Activity Classification using Accelerometers and Machine Learning for Complex Construction Worker Activities. *Journal of Building Engineering*, In Press, Corrected Proof. https://doi.org/10.1016/j.jobe.2020.102001

Schorfheide, F. and Song, D. (2015). Real-Time Forecasting with a Mixed-Frequency VAR. *Journal of Business and Economic Statistics*, 33:3, 366-380. Doi: 10.1080/07350015.2014.954707.

Seng, K.P., Ang, L., Schmidtke, L.M., and Rogiers, S.Y. (2018). Computer Vision and Machine Learning for Viticulture Technology. *IEEE Access*, 6, 67494 – 67510. doi: 10.1109/ACCESS.2018.2875862.

Smets, F., Warne, A. and Wouters, R. (2014). Professional forecasters and real-time forecasting with a DSGE model. *International Journal of Forecasting,* 30,0981-995. doi:10.1016/j.ijforecast.2014.03.018.

Stock, J.H. and Watson, M.W. (2002). Macroeconomic Forecasting using Diffusion Indexes. *Journal of Business and Economic Statistics*, 20:2, 147-162. Doi: 10.1198/073500102317351921.

Stock, J.H. and Watson, M.W. (2003). Forecasting Output and Inflation: The role of Asset Prices. *Journal of Economic Literature.* Vol. CLI, 788-829.

Wan, K.H., Dahlsten, O., Kristjánsson, H., Gardner, R. and Kim, M.S. (2017). Quantum generalisation of feedforward neural networks. *NPJ Quantum Information*, 3 (36). doi:10.1038/s41534-017-0032-4.

Wu, Z., Zhang, W., Zhao, J., Chen, C., and Ji, P. (2019). Optimized Complex Network Method (OCNM) for Improving Accuracy of Measuring Human Attention in Single-

Electrode Neurofeedback System. *Computational Intelligence and Neuroscience*, 2167871, 1-10. https://doi.org/10.1155/2019/2167871

Yang, Y., Garcia-Morillo, I, Hospedales, T.M. (2018). Deep Neural Decision Trees. ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden.

Zidan, M., Abdel-Aty, A.-H., El-shafei, M., Feraig, M., Al-Sbou, Y., Eleuch. H. and Abdel-Aty, M. (2019). Quantum Classification Algorithm Based on Competitive Learning Neural Network and Entanglement Measure. *Appl Sci*, 9, 1277. https://doi.org/10.3390/app9071277.

Zhang, J., Li, L., and Chen, W. (2020). Predicting Stock Price Using Two-Stage Machine Learning Techniques. *Computational Economics*, published online. https://doi.org/10.1007/s10614-020-10013-5

Zhang, G.P. and Qi, M. (2005). Neural network forecasting for seasonal and trend time series. *European Journal of Operational Research*, 160, 501–514. doi:10.1016/j.ejor.2003.08.037.

Zhao, Y., Li, J., and Yu, L. (2017). A deep learning ensemble approach for crude oil price forecasting. *Energy Economics*, 66(C), 9-16. https://doi.org/10.1016/j.eneco.2017.05.023

Zhong, X., and Enke, D. (2019). Predicting the daily return direction of the stock market using hybrid machine learning algorithms. *Financial Innovation*, 5, 24. https://doi.org/10.1186/s40854-019-0138-0