Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility

Citation for published version:
Trierweiler Ribeiro, G, Santos, A, Cocco Mariani, V & dos Santos Coelho, L 2021, 'Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility', *Expert Systems with Applications*, vol. 184, 115490. https://doi.org/10.1016/j.eswa.2021.115490

Digital Object Identifier (DOI):
10.1016/j.eswa.2021.115490

Published In:
Expert Systems with Applications

Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility

Gabriel Trierweiler Ribeiro, *e-mail: gabrielribeiro.ee@gmail.com* [a]

André Alves Portela Santos, *e-mail: andreportela@gmail.com* [b, c]

Viviana Cocco Mariani, *e-mail: viviana.mariani@pucpr.br* [d]

Leandro dos Santos Coelho, *e-mail: leandro.coelho@pucpr.br* [e]

[a] Department of Electrical Engineering, Federal University of Parana (UFPR), Curitiba, PR, Brazil

[b] Department of Economics, Federal University of Santa Catarina (UFSC), Florianopolis, SC, Brazil

[c] Business School, University of Edinburgh, Scotland

[d] Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil

[e] Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR), Curitiba, PR, Brazil

Abstract

The prediction of stock price return volatilities is important for financial companies and investors to help to measure and managing market risk and to support financial decision-making. The literature points out alternative prediction models - such as the widely used heterogeneous autoregressive (HAR) specification - which attempt to forecast realized volatilities accurately. However, recent variants of artificial neural networks, such as the echo state network (ESN), which is a recurrent neural network based on the reservoir computing paradigm, have the potential for improving time series prediction. This paper proposes a novel hybrid model that combines HAR specification, the ESN, and the particle swarm optimization (PSO) metaheuristic, named HAR-PSO-ESN, which combines the feature design of the HAR model with the prediction power of ESN, and the consistent PSO metaheuristic approach for hyperparameters tuning. The proposed model is benchmarked against existing specifications, such as autoregressive integrated moving average (ARIMA), HAR, multilayer perceptron (MLP), and ESN, in forecasting daily realized volatilities of three Nasdaq (National Association of Securities Dealers Automated Quotations) stocks, considering 1-day, 5-days, and 21-days ahead forecasting horizons. The predictions are evaluated in terms of r-squared and mean squared error performance metrics, and the statistical comparison is made through a Friedman test followed by a *post-hoc* Nemenyi test. Results show that the proposed HAR-PSO-ESN hybrid model produces more accurate predictions on most of the cases, with an average R^2 (coefficient of determination) of 0.635, 0.510, and 0.298, an average mean squared error of 5.78×10^{-8}, 5.78×10^{-8}, and 1.16×10^{-7}, for 1, 5, and 21 days ahead on the test set, respectively. The improvement is statistically significant with an average rank of 1.44 considering the three different datasets and forecasting horizons.

Keywords: Volatility prediction, Echo state network, Heterogeneous autoregressive model, Particle swarm optimization.

1 INTRODUCTION

Financial institutions (e.g. banks, insurance, and asset management companies), as well as individual investors, deal with uncertainties in investment portfolios. These uncertainties that arise from the fluctuations in asset prices impact the level of the risk in their financial portfolios and affect the decision-making process. A widely used proxy measure of this uncertainty is the notion of *realized volatility* (Andersen, Bollerslev, and Meddahi, 2005), which measures the variability in the changes in asset prices by using the information of high-frequency (intraday) data. Thus, knowing the volatility of a given stock in advance can be valuable for conducting enhanced investment decisions and for supporting both institutional and individual investors in the assessment of the level of risk in their financial portfolios. In that sense, quantitative models designed to forecast realized volatilities have become a key element in the set of tools used to measure and manage the risk associated with the fluctuations in asset prices.

The forecasting of realized volatility has commonly been accomplished by statistical models such as the autoregressive integrated moving average (ARIMA) model and the heterogeneous autoregressive (HAR) model proposed in Corsi (2009), but nowadays has been often accomplished with machine learning models - a branch of artificial intelligence field that develop models which learn from experience - such as multilayer perceptron (MLP), Random Forest, and Support Vector Machines (SVM). Newer models often propose hybrid versions of the existing machine learning models, aggregating them into an ensemble learning approach, tuning the hyperparameters with metaheuristic algorithms, or engineering new features. A systematic review of such techniques for the stock market forecast is found in Bustos and Pomares-Quimbaya (2020), whereas an analysis of deep learning application for stock markets prediction is provided by Chong, Han, and Park (2017).

In the context of ensemble learning models, Kristjanpoller and Minutolo (2015) proposed a stacked generalized autoregressive conditional heteroskedasticity (GARCH) model combined with an artificial neural network with additional handcrafted features for the prediction of gold price return volatility. Pierdzioch, Risse, and Rohloff (2016) proposed a boosting approach for the prediction of gold price return volatility. Di Sanzo (2018) proposed an MRV (Markov Regime Switching) approach for a regime-switching GARCH model for the prediction of crude oil price return volatility. Kim and Won (2018) proposed an ensemble of long short-term memory (LSTM) with variations of GARCH models for prediction of volatilities of the KOPSI 200 (Korea Composite Stock Price Index 200) stock index returns. A stacked learning model has been proposed by Ramos-Pérez *et al.* (2019) for predicting the volatility of the S&P 500 (Standard & Poor's 500) stocks. Alizadeh, Huang, and Marsh (2019) presented a mixture of specialists, with different HAR specifications combined with an MRS approach for the prediction of energy contracts in the Tokyo Commodity Exchange (TOCOM).

In the context of feature engineering, a model based on empirical mode decomposition for generating features had been proposed by Gong and Lin (2019) for predicting the volatility of the S&P 500 stocks; Atkins, Niranjan, and Gerding (2018) employed Latent Dirichlet Allocation to represent information from the financial news feed and to predict the direction of US stock market volatility with naïve Bayes; Afkhami, Cormack, and Ghoddusi (2017) used Google search keywords as features for prediction of energy prices volatility; Choudhury *et al.* (2014) proposed the use of features derived from clustering with Self Organizing Maps (SOM) followed by an SVM prediction model for price return and volatility forecasting in the Indian market.

The alternative prediction models can use only past values of the volatility series, yielding univariate models, or can use any other indicator as a feature, therefore yielding multivariate models. The multivariate approach may lead to better predictions if appropriate exogenous variables are found. In the context of multivariate models, Ma *et al.* (2018) obtained predictions of oil price return volatilities whereas Pierdzioch, Risse, and Rohloff (2016) predicted the gold price return volatility. Finally, Walther, Klein, and Bouri (2019) have tested exogenous variables that most affect the predictions of cryptocurrency volatility.

The HAR model is used to obtain volatility predictions of oil price return (Degiannakis and Filis, 2017; Alizadeh, Huang, and Marsh, 2019; Gong and Lin, 2019) and produced accurate predictions that could not be outperformed by a hybrid model of principal component analysis, GARCH and an artificial neural network (ANN) as concluded by Vortelinos (2017).

On the other hand, the echo state network (ESN) is a recurrent ANN that makes use of the reservoir computing paradigm and often achieves improved modeling performance with a fast training procedure. The ESN has been widely used and can be found in a variety of applications such as remaining useful life prediction (Rigamonti *et al.*, 2018), energy forecasting (Ribeiro *et al.*, 2016; Ribeiro, Mariani and Coelho, 2019; Hu *et al.*, 2020; Hu, Wang, and Lv, 2020), fault prognostics (Xu *et al.*, 2020), credit scoring (Xia *et al.*, 2018), and tourism (Lv, Peng, and Wang, 2018). It has also emerged in deep specifications such as in Ma, Shen, and Cottrell (2020).

Fičura (2018) compares the ESN with HAR models in predicting stock market volatility of several indexes and finds that, on average, the HAR models perform better but also suggests that the ESN has a potential for being improved. Applications of ESN for stock price return forecasting are found in Zhang, Liang, and Chai (2013) and Dan *et al.* (2014). Other studies that consider the application of ESNs are Yao *et al.* (2019) and Yao (2020). A summary of the ESN performance as reported in previous related studies is presented in Table 1.

Table 1 - Summary of the ESN performance as reported in previous related studies

Related Study	Asset	Performance Metric	Value
Afkhami, Cormack, and Ghoddusi (2017)	Energy price return volatility	Adjusted R^2	0.186
Alizadeh, Huang and Marsh (2019)	Gasoline volatility	R^2	0.4584
Dan *et al.* (2014)	Shanghai composite index stock price	Mean squared error	0.016
Degiannakis and Filis (2017)	Oil price return volatility	Mean squared error	69.36
Di Sanzo (2018)	Oil price return volatility	Mean squared error	20.11
Fičura (2018)	S&P500, DJIA, and Nikkei indices volatilities	R^2	0.168
Gong and Lin (2019)	S&P 500 index volatility	R^2	0.6006
Kim and Won (2018)	KOSPI 200 stock index volatility	Mean squared error	0.00149
Kristjanpoller and Minutolo (2015)	Gold	Mean absolute percentage error	0.6493
Ma *et al.* (2018)	Aggregate oil price return volatility	R^2	0.2087
Ramos-Pérez *et al.* (2019)	S&P 500 index volatility	Root mean squared error	0.00254
Vortelinos (2017)	US financial markets volatility	R^2	0.7732
Zhang, Liang, and Chai (2013)	Microsoft Company stock price	Hit rate	0.788

This paper adds to the existing literature on realized volatility forecasting with ESN by putting forward two contributions. First, we specify and implement a new hybrid model for predicting realized volatility of stock price returns, called HAR-PSO-ESN, which combines the engineered features of the HAR model and the potential of ESN for time series prediction, making use of the PSO metaheuristic, a swarm intelligence approach, for hyperparameters tuning. Second, we build on the work of Fičura (2018) and show how the ESN can be extended and used in conjunction with other models along with a metaheuristic-based tuning strategy. To the best of our knowledge, there is only the publication of Fičura (2018) that deploys the ESN model in a horse-race against the HAR specification. In that sense, our paper helps to settle a new hybrid benchmark for research on this challenging topic.

The proposed HAR-PSO-ESN model has an advantage over the traditional ARIMA and HAR models due to its suitability for nonlinear time series. Moreover, it has also an advantage over the traditional MLP model because the ESN, as a type of recurrent neural network, is more appropriate for time series forecasting. Finally, it is an advantage over the conventional ESN due to the inclusion of the HAR features, which are proven very relevant for price return volatility forecasting.

The problem at hand consists of building models to predict future values of stock price return volatilities based only on past values of the series. The proposed HAR-PSO-ESN hybrid model is empirically benchmarked against the other four specifications: ARIMA, HAR, ESN, and MLP. The alternative models are used to obtain forecasts of the daily realized volatilities based on 5-min. intraday squared returns of three Nasdaq stocks for 1-day ahead, 5-days ahead (1 week), and 21-days ahead (1 month) forecasting horizons. The predictions are evaluated in terms of the mean squared error (MSE) and coefficient of determination or R-squared (R^2) metrics, as well as with a statistical significance test. The results show that the proposed model produces more accurate predictions in the majority of the cases. Moreover, the differences in forecasting performance are statistically significant.

The remainder of this paper is organized as follows. In Section 2, we motivate the need for research in stock price return forecasting. In Section 3, we define the concept of realized volatility and details the alternative individual models used to obtain predictions. Section 4 presents the dataset employed in the experiments, the problem formulation, the methodology used for applying machine learning models for time series prediction, the set-up used to compare among alternative models, the accuracy metrics and statistical significance test, as well as the proposed HAR-PSO-ESN hybrid model. Results of prediction accuracy and statistical significance are presented in Section 5. Finally, the conclusions of the paper are presented in Section 6.

2 NEED FOR RESEARCH

Given that forecasts of realized volatilities are key to manage the risks in the asset price fluctuations, more accurate predictions imply a lower level of uncertainties in investment decisions and improved investment performance. However, there is no deterministic forecasting model that can deliver the most accurate forecasts for *all* volatility time series, as supported by the “No Free Lunch theorem” in Wolpert and Macready (1997). Wolpert (2002) recommends that it is important to develop many different types of models to cover the wide variety of data that occurs in the real world.

Hence, the search for more accurate forecasting models in specific problems is unavoidable since a general solution may never emerge. Considering the potential of improvement in ESN-based forecasting models for stock volatilities reported by Fičura (2018), and the limitations of the successful HAR models to nonlinear time series forecasting, would a hybrid model that combines the ESN and HAR characteristics deliver more accurate stock price return volatility forecasts?

3 REALIZED VOLATILITY AND PREDICTION MODELS

This section provides background regarding realized volatility, the individual benchmark models, and the models that serve as the base for building the proposed HAR-PSO-ESN hybrid model.

3.1 Realized Volatility

We consider the estimator of the realized volatility defined in Andersen $et\ al.$, (2000, 2001) which is based on sampling the stock price at time t, denoted by p_t, on regular time intervals (e.g., 1, 5, 10 minutes) within a given market session. Assume that the prices on a trading day t were sampled with a regular interval with $m+1$ points $0,1,...,m$ such that $p_{i,t}$ is the i-th observation of the log price on day t. The realized volatility (*RV*) can the therefore estimated as

$$RV_t^m = \sum_{i=1}^{m} (p_{i,t} - p_{i-1,t})^2 , \tag{1}$$

$$RV_t^m = \sum_{i=1}^{m} r_{i,t}^2 . \tag{2}$$

The estimator of the realized variance in Eqs. (1)-(2) is shown to be consistent for the true unobserved latent variance. Moreover, existing evidence suggests that the use of intraday information results in more accurate volatility measures and predictions.

3.2 Prediction models

Five prediction models have been employed for the stock price return volatility prediction problem. Four existing models, namely ARIMA, ESN, HAR, and MLP, and a new proposed model which is a hybridization of HAR with ESN. Next, we describe each of these approaches.

3.2.1 ARIMA model

The ARIMA model is a traditional model for time series forecasting. The model is based on three kinds of features, which are autoregressive (i.e. past values of the time series), integrated (i.e. differentiated values of the time series), and moving average (i.e. an average of past values of the residual series).

The equation that describes the ARIMA model, as presented in Zhang et al. (2018), is

$$\Delta^d y_t = \alpha_0 + \sum_{i=1}^{p} \beta_i \Delta^d y_{t-1} + \varepsilon_t + \sum_{j=1}^{q} \alpha_j \varepsilon_{t-j}, \tag{3}$$

where $\Delta^d y_t$ is the sample t of the time series y differentiated d times; α_0, β_i, and α_j are the model parameters; p, d, and q are integers that represent the model orders (i.e. hyperparameters) of the autoregressive, integration, and moving average terms respectively; and ε_t is a random white noise signal.

3.2.2 HAR model

The HAR model proposed in Corsi (2009) is a major workhorse for realized volatility modeling and a traditional model often employed in econometrics with good results. The main characteristic of the HAR model is the handcrafted designed features. The equations that describe the HAR model are given by

$$V_{d+1} = \beta_0 + \beta_d RV_d + \beta_w RV_w + \beta_m RV_m, \tag{4}$$

$$RV_d = V_d, \tag{5}$$

$$RV_w = \frac{1}{5}\sum_{i=1}^{5} V_{d-i}, \tag{6}$$

$$RV_m = \frac{1}{22}\sum_{j=1}^{22} V_{d-j}, \tag{7}$$

where V is a proxy for the true value of the volatility, d is the index of the current day in the time series; β_0, β_d, β_w, and β_m are the model parameters; RV_d, RV_w, and RV_m are the average volatility values of the last day, week, and month respectively, in business days.

The parameters of the HAR model (i.e., β_0, β_d, β_w, and β_m) are estimated with the ordinary least squares (OLS) algorithm.

3.2.3 MLP neural network model

The MLP is a feedforward artificial neural network composed of a set of interconnected processing units called neurons as displayed in Fig. 1. The network is fed with inputs X, and propagates them to the first layer of neurons $F^{(1)}$. The output of each neuron $j \in \{1,2,\ldots,N_1\}$ is then calculated as in Eq. (8). The output of each neuron j in a given layer L, with N_L neurons are calculated as in Eq. (9), including the output layer,

$$y_j^{(1)} = f_j\left(\sum_{i=1}^{K} w_{ji}^{(1)} x_i\right), \tag{8}$$

$$y_j^{(L)} = f_j\left(\sum_{i=1}^{N_{L-1}} w_{ji}^{(L)} y_i^{(L-1)}\right), \tag{9}$$

where i and j are the indexes of the neurons in layer L, N is the number of neurons in a given layer, w is the weight of the synapsis connecting two given neurons, y is the output of a given neuron, and f is the activation function.

Two activation functions that are commonly found in MLP architectures are the sigmoid function and the hyperbolic tangent (tanh) defined respectively as

$$f(x) = \frac{1}{1 - e^{-x}} \tag{10}$$

$$f(x) = \tanh(x). \tag{11}$$

The training of the MLP consists of adjusting the connections among neurons (also referred to as synapsis) iteratively by minimizing the error between the desired outputs and the network outputs, given the same inputs. However, the MLP also requires the setting of hyperparameters, which are the number of layers, the number of neurons in each layer, and additional parameters depending on the learning algorithm (e.g. regularization parameters).

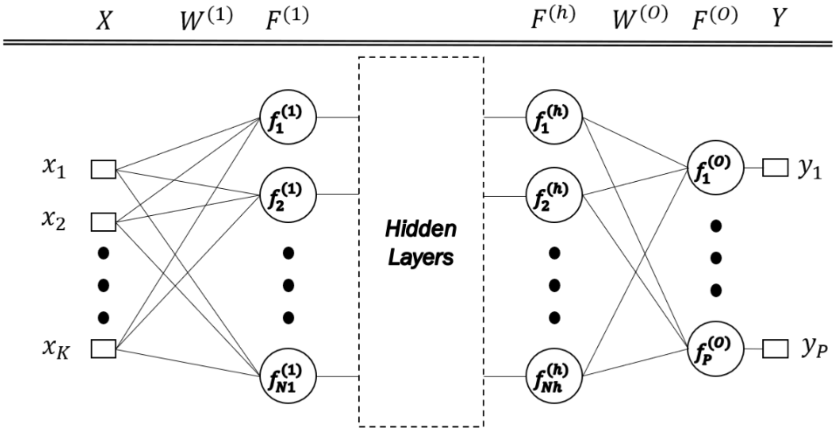


Figure 1 - Architecture of an MLP neural network.

3.2.4 ESN model

ESN is a class of recurrent neural networks (RNNs) that makes use of the reservoir computing paradigm for efficient training. The learning of an RNN is usually performed through an algorithm based on the gradient of a cost function, which may take too many iterations to converge or get stuck into sub-optimal values. On the other hand, ESN uses the echo state property (ESP) such that all the weights are initialized randomly except the output weights, which are obtained with the OLS algorithm, therefore resulting in efficient and fast learning.

The architecture of an ESN is presented in Fig. 2. The architecture presents two kinds of weights, those of the forward connections and those of the recurrent connections. The forward connections link the inputs X with the hidden layer $F^{(h)}$, and the hidden layer with the output layer $F^{(O)}$. The recurrent weights link the neurons in the reservoir with each other, as well as output neurons with hidden layer neurons.

The weights of the connections between the inputs and the hidden layer are called $W^{(in)}$; the weights between the hidden layer and the output layer are called $W^{(O)}$; the weights between the neurons in the hidden layer are called $W^{(R)}$; and the weights between the outputs and the hidden layer are called $W^{(fb)}$. The reservoir is composed of the weights $W^{(in)}$, $W^{(R)}$, and $W^{(fb)}$, which are all initialized randomly using a uniform probability distribution function.

After the training inputs and outputs are propagated through the reservoir, the outputs of the neurons in the hidden layer are collected and compose the states S of the ESN. Then, the weights $W^{(O)}$ are calculated using the OLS algorithm, considering the states S as the inputs and $f^{-1}(Y)$ as outputs. The ESN computations are given by

$$s_j^{(h)}(t+1) = f\Big(\sum_{i=1}^{K} w_{ji}^{(in)}(a_i i_i + b_i) + \sum_{i=1}^{N} w_{ji}^{(R*)}(\eta s_i^{(h)}(t)) + \sum_{i=1}^{P} z_i w_{ji}^{(fb)} {f_i^{(O)}}^{-1}(c_i y_i + d_i)\Big) - \alpha_i s_j^{(h)}(t), \quad (12)$$

$$W^{(R*)} = \rho W^{(R)}, \quad (13)$$

where s is the state, j is the index of the neuron in the hidden layer, t is the current sample in the time series, f is the neuron activation function; K, N, and P are the number of inputs, reservoir neurons, and output neurons respectively, w is the synaptic weight; a, b, c, d, and z are input scaling, input shift, target scaling, target shift, and feedback scaling respectively; α is the leaking rate, ρ is the spectral radius, and η is the noise.

The parameters of the ESN are the weights W, but its application requires the setting of additional hyperparameters that influence its performance, stability, and compliance to the echo state property (Yildiz, Jaeger, and Kiebel, 2012). The a, b, c, d, and z parameters determine the operation region of the internal signals into the activation function f, such that smaller values explore the linear region around zero and higher values explore the nonlinear region near saturation. The α is the leaking rate parameter and set how much the next stage of the network depends on the previous one. The η is the noise parameter and represents a small random value added to the previous states and acts as a regularization parameter, with the intent to improve the generalization ability. The ρ is the spectral radius parameter and is often considered the most important one because it depends if the echo state property is valid, and hence if the ESN states will converge and be able to represent the system dynamics. The reservoir size N must also be set and has a great influence on the ESN performance. Beyond the hyperparameters shown in Eqs. (10) and (11), there is also the sparsity degree parameter φ, which makes the reservoir weights matrix sparse, randomly setting several weights to zero, having little impact on the performance but a great impact on the network computation speed.

The reservoir size impacts directly on the memory capacity and function approximation ability of the ESN, which could be as large as possible since it tends to result in better performance. However, due to computational power limitations and overfitting issues, it is imperative to set an upper limit and look for an optimal value.

Another key parameter of the ESN is the spectral radius, which is related to the compliance with the ESP and impacts directly on the model's performance. It shall assume higher values for very nonlinear systems and smaller values for less nonlinear systems. Values below unity guarantee the ESP for the majority of empirical applications.

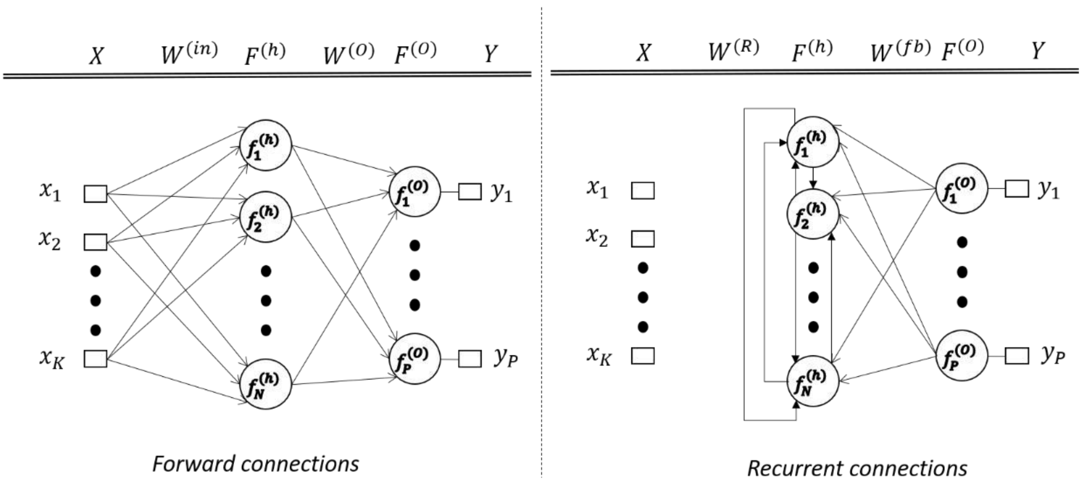


Figure 2 - Architecture of an ESN.

4 Materials and Methods

This section presents the data and the empirical problem at hand, the techniques for handling the data to be processed by machine learning models, the proposed HAR-PSO-ESN model, the compared models that serve as benchmarks, and the statistical significance test.

A flowchart with all steps adopted when performing the empirical analysis is presented in Figure 3. The study is performed in three sequential steps, which are the data pre-processing, modeling, and model evaluation. In the data pre-processing step, we present the data, the formulation of the forecasting problem, the conversion of the data from the time series to the supervised learning format, and the procedure for scaling the data before the modeling step. Next, the modeling step consists of the implementation of the benchmarking models as well as of the proposed HAR-PSO-ESN model. Finally, the forecasts of the previous step are evaluated and compared in terms of accuracy and statistical hypothesis tests. Each step is detailed in the following sections.

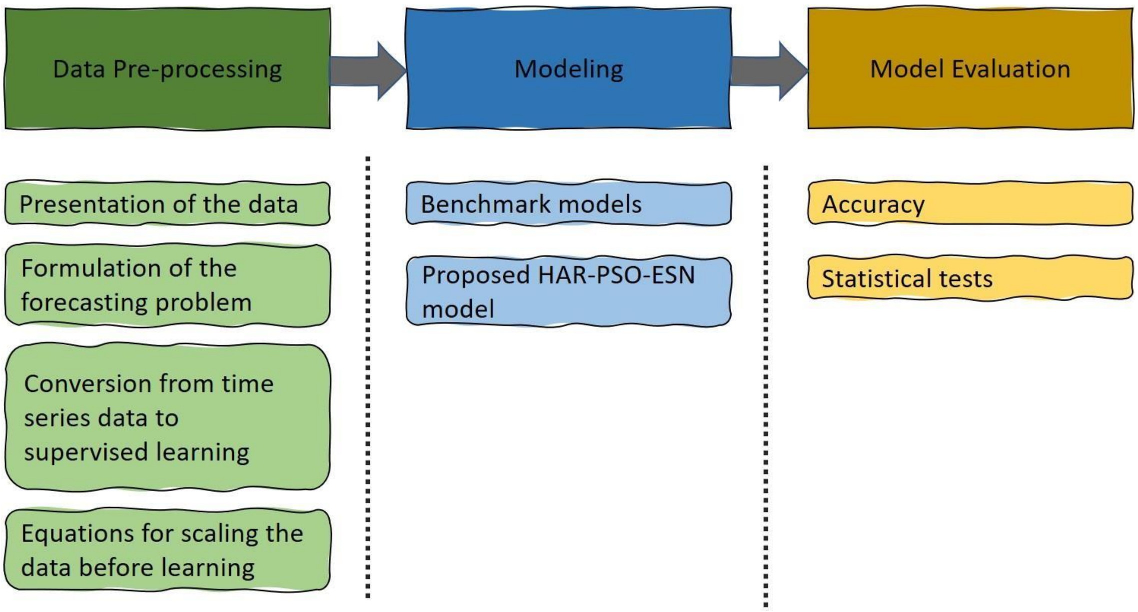


Figure 3 - Flowchart of steps

4.1 Data and Problem

We assemble a dataset of daily realized volatilities from three Nasdaq companies: Caterpillar (CAT), eBay (EBAY), and Microsoft (MSFT). To construct the daily realized volatilities for each stock according to Eqs. (1)-(2), we sum for each day the squared intraday return sampled at the 5-min frequency. A time-series visualization of the three series is presented in Fig. 4. Each time series has 2745 observations, which is equivalent to 549 weeks of records. The time series are noisy and present noticeable bursts at some points, as can be seen between the 1000th and 1500th days. Two additional bursts can be seen between the 1500th and 2000th days. In Fig. 4, it can be noticed that EBAY volatilities (Fig. 4b) present higher daily changes, followed by CAT (Fig. 4a), and MSFT presents smaller daily variations (Fig. 4c), which is observed mainly in the intervals between the 1st and 1000th days and 2000th and 2500th days.

The distribution of the time series values is presented in Fig. 5. The price return volatilities are much more concentrated at lower values, between 0 and 0.001, but eventually assume values from five to six-time higher, demonstrating that the data is highly skewed. Each considered time series also present a different median. EBAY presents the highest median values, followed by CAT, and MSFT presents the lowest one.

The problem at hand consists of building models to predict future values of stock price return volatilities based only on past values of the series. More specifically, three forecasting horizons are considered, which are 1-day ahead, 5-days ahead (1 week), and 21-days ahead (1 month), as stated in Eq. (14) given by

$$\hat{V}_{d+h} = F(V_d, V_{d-1}, \ldots, V_{d-l}) \tag{14}$$

where $\hat{V}$ is the forecasted volatility, h is the forecasting horizon, d is the day number index in the time series, V is the observed values of the volatilities in the time series, and l is the maximum lag of the time series relevant for the prediction model F.

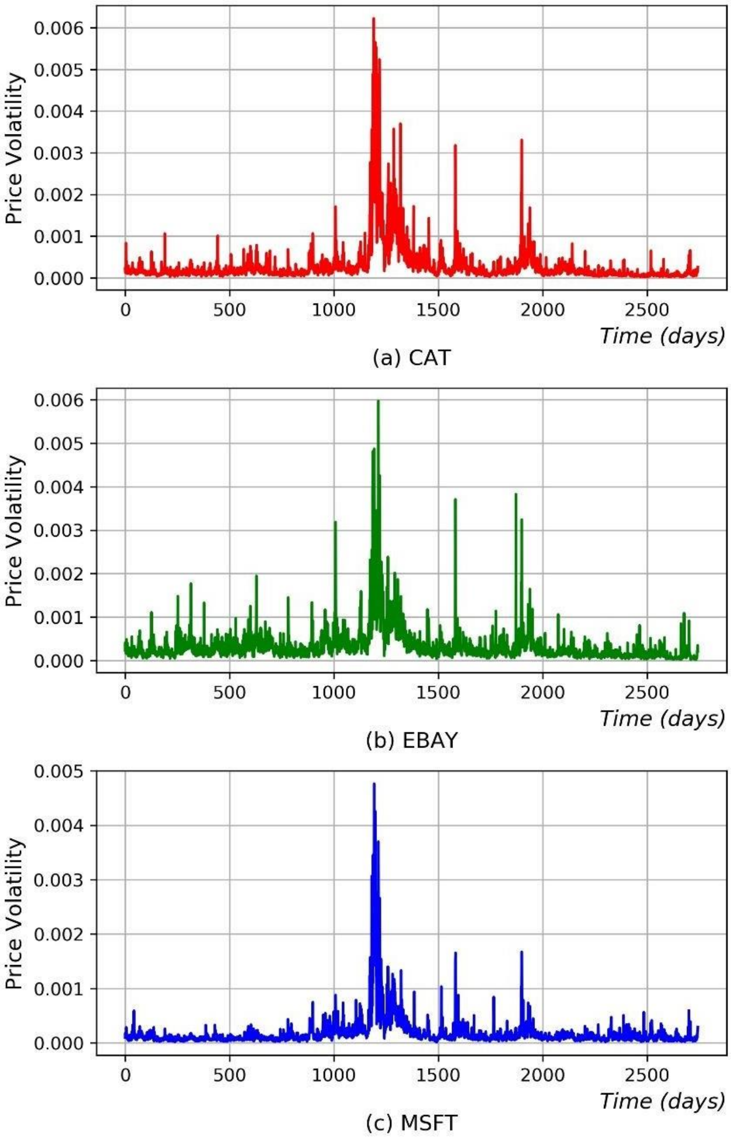


Figure 4 - Line charts of time series to be predicted.

The model selection procedure was performed as described in Table 1. For each forecasting horizon h, three experiments were carried out. In each experiment, a different time series is used as training, validation, and test set. The training set is used for the model learning, while the validation set is used for hyperparameters tuning, if necessary. The test set is used for the evaluation of the predictions. When the model uses the validation set for hyperparameters tuning, the training and validation sets are then used for fitting the best-tuned model to predict the test set.

The predictions were evaluated according to two metrics, which are the MSE, and the coefficient of determination R^2, which are calculated as in Eqs. (15) and (16) respectively. These equations are given by

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 \tag{15}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} \tag{16}$$

where N is the number of prediction samples, $\hat{y}_i$ is the prediction for the test sample i, y_i is the true value of test sample i, and $\bar{y}$ is the average value of true test samples.

Table 2 - Data split for model selection.

Experiment	CAT	EBAY	MSFT
1	Training set	Validation set	Test set
2	Test set	Training set	Validation set
3	Validation set	Test set	Training set

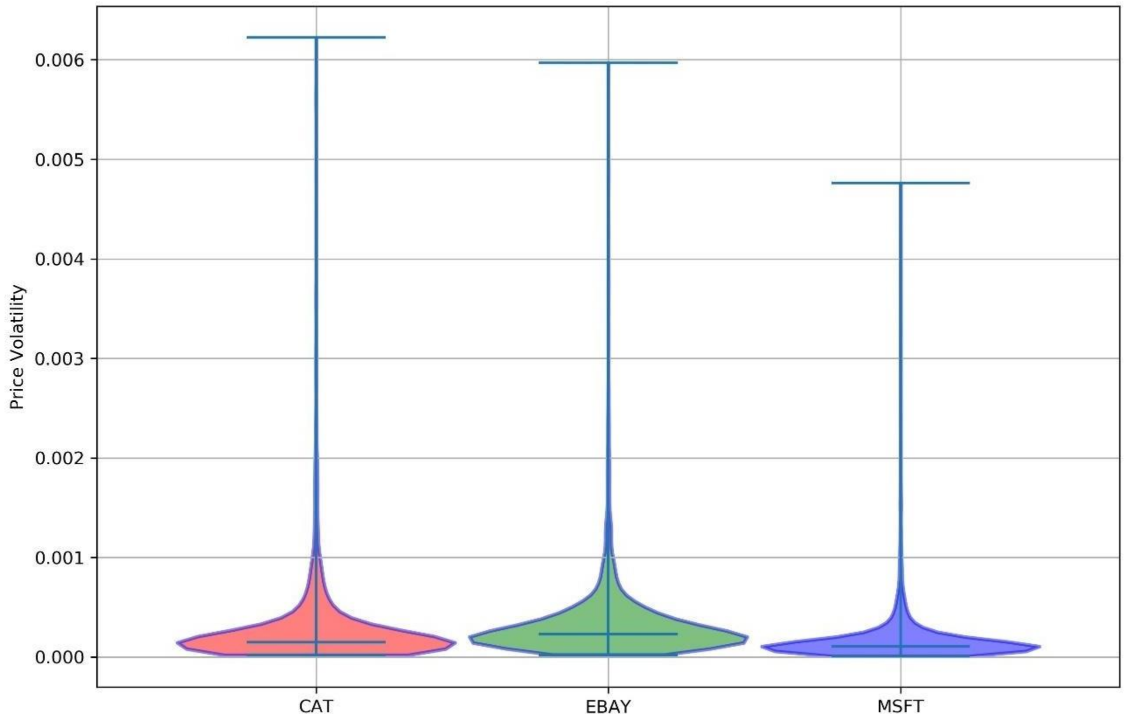


Figure 5 - Distribution of values of price volatilities for each dataset.

4.2 Supervised learning datasets

A common approach for building machine learning models for time series prediction is the transformation of the time series from a sequence of observed values $S = \{s_1, s_2, \ldots, s_N\}$ to a pair of input and output matrices, X and Y, respectively, such that each row is a sample and each column is a variable.

The sliding window method is usually employed and works as illustrated in Fig. 6. The raw time series is converted into a windowed time series, which is a matrix whose size depends on the number of features M, the forecasting horizon H, and the length of the time series N. The first M columns are composed of past values while the latest H columns are composed of future values. The matrix of inputs X is built from the past values of the windowed time series and may vary depending on the employed feature engineering algorithm. The matrix of outputs Y is built from the future values of the windowed time series and the output is selected as the columns relative to the desired forecasting horizons.

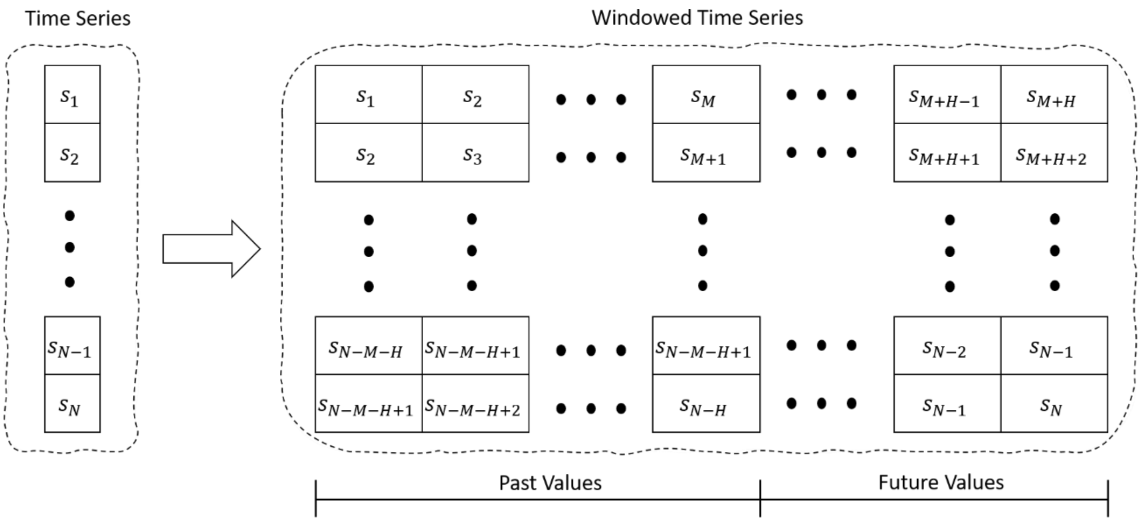


Figure 6 - Sliding window technique.

4.3 Data Scaling

Some prediction models are sensitive to the scales of the features. To avoid such an issue, data has been normalized into the interval $[0,1]$. All datasets (i.e. train, validation, and test) are normalized concerning the minimum and maximum values of the training dataset since the model cannot know the minimum and maximum values of the validation or the test set in advance. Before calculating the performance metrics, the predictions are unscaled and given by the following equations:

$$X^{(j)}_{train,scaled} = \frac{X^{(j)}_{train} - X^{(j)}_{min,train}}{X^{(j)}_{min,train} - X^{(j)}_{min,train}} \tag{17}$$

$$Y^{(h)}_{train,scaled} = \frac{Y^{(h)}_{train} - Y^{(h)}_{min,train}}{Y^{(h)}_{min,train} - Y^{(h)}_{min,train}} \tag{18}$$

$$X^{(j)}_{validation,scaled} = \frac{X^{(j)}_{validation} - X^{(j)}_{min,train}}{X^{(j)}_{min,train} - X^{(j)}_{min,train}} \tag{19}$$

$$Y^{(h)}_{validation,scaled} = \frac{Y^{(h)}_{validation} - Y^{(h)}_{min,train}}{Y^{(h)}_{min,train} - Y^{(h)}_{min,train}} \tag{20}$$

$$X^{(j)}_{test,scaled} = \frac{X^{(j)}_{test} - X^{(j)}_{min,train}}{X^{(j)}_{min,train} - X^{(j)}_{min,train}} \tag{21}$$

$$Y^{(h)}_{test,scaled} = \frac{Y^{(h)}_{test} - Y^{(h)}_{min,train}}{Y^{(h)}_{min,train} - Y^{(h)}_{min,train}} \tag{22}$$

where X is the input matrix, j is the column of the input matrix, Y is the output matrix, h is the column of the output matrix, and min and max are the minimum and maximum values, respectively.

4.4 Proposed HAR-PSO-ESN model

The proposed HAR-PSO-ESN model has three main building blocks. The first is the HAR model, which is commonly employed in econometrics with good performance due to its carefully handcrafted features. The second is the ESN model, which is an efficient ANN model adequate for time series forecasting that can transform lagged values of the series in a higher dimension state space such as a kernel function. We aim to enhance the forecasting performance of the ESN model using the handcrafted features of the HAR model. Finally, the ESN hyperparameters are tuned with the PSO algorithm, therefore yielding the hybrid model named HAR-PSO-ESN.

Specifically, the proposed HAR-PSO-ESN hybrid model makes use of the HAR features and the ESN architecture as exhibited in Fig. 7. The first step in the application of the model requires the transformation from the price return volatility time series to a supervised learning dataset with inputs and outputs. As for the inputs, it is often recommended to have a matrix with samples represented as rows and features as columns. As for the output, it is recommended to have samples as rows and output targets as columns. The inputs are composed of past values of the series and the outputs are composed of future values. The input is then processed for the extraction of the HAR features, which are the inputs of the ESN. The outputs do not require any further processing. Finally, the ESN hyperparameters are tuned with the PSO algorithm as explained in section 4.5.3.

As shown in 7, the volatility time series in the training set is used for $training$ the ESN architecture with an initial set of hyperparameters, resulting in an initially trained ESN. Then, the forecasts obtained with the trained ESN based on the volatility time series in the $validation$ set are evaluated by the PSO algorithm to iteratively tune the hyperparameters. After a given number of iterations, the trained and PSO-tuned ESN is used to obtain forecasts based on the inputs belonging to the $test$ set. Finally, the forecasts obtained with the HAR-PSO-ESN model in the test set are evaluated and compared to those obtained with the alternative models.

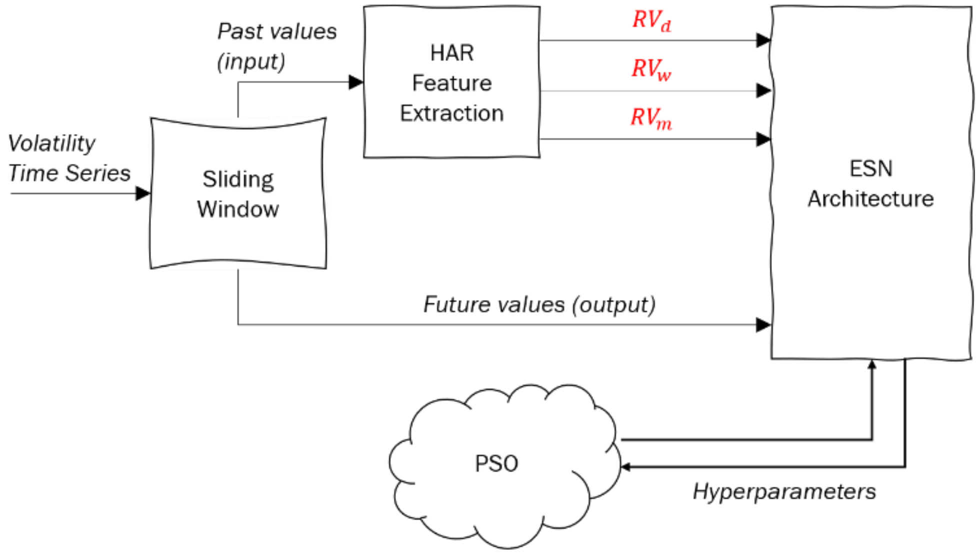


Figure 7 - Proposed HAR-PSO-ESN hybrid model.

4.5 Benchmark models

The proposed model was benchmarked with the ARIMA, HAR, MLP, and ESN models. The standard HAR model does not have hyperparameters to be tuned. However, the ARIMA, MLP, and ESN do require the setting of specific hyperparameters, which are tuned with the algorithms described next.

4.5.1 Grid search combined with ARIMA model

The application of the ARIMA model requires the setting of three parameters, which are the order of the autoregressive term p, the order of the integration term d, and the order of the moving average term q. We have considered a grid search (GS) algorithm for tuning such parameters.

The ARIMA model was implemented using the Stasmodel Python library (Seabold and Perktold, 2010). The hyperparameters p and q have been searched in the set $\{0,1,2\}$, while the hyperparameter d has been searched in the set $\{0, 1\}$. Higher orders have been tested but led the model to a non-convergence state.

4.5.2 GS-MLP

The application of the MLP requires the setting of the number of layers, the size of the layers, i.e., the number of neurons in each layer, and the regularization parameter. The hyperparameters are tuned with the GS algorithm with 5 points in each grid. The number of layers is searched in the range $[1,5]$, the size of the layers was considered the same for all layers and searched in the range $[10,300]$, and the regularization parameter was searched in the range $[10^{-4}, \ldots, 10^{0}]$.

The fitting of the MLP requires a supervised learning dataset. The dataset was obtained using the sliding window technique, and the features have been selected through the partial autocorrelation function of the training time series. The past values that present a coefficient of correlation outside a confidence interval of 95% were selected as significative features.

The learning algorithm employed was Limited-memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (LBFGS). The maximum number of iterations was set as 10^3, the activation function was the $\tanh(.)$, the tolerance set to 10^{-4}, and the maximum number of function evaluations as 10^3. The MLP has been implemented using the sci-kit-learn library for Python (Pedregosa, Weiss, and Brucher, 2011).

4.5.3 PSO-ESN

The application of the ESN model requires the setting of the reservoir size, input scaling, input shift, target scaling, target shift, feedback scaling, the leaking rate, spectral radius, sparsity degree, and the noise hyperparameters. These hyperparameters have been tuned with PSO algorithm.

The reservoir size has been searched in the interval $[1,1500]$, the spectral radius in the interval $[0.1,1.5]$, the sparsity degree in the interval $[0,0.95]$, the noise in the interval $[0,1]$, the input shaft in the interval $[0,10]$, the input scaling in the interval $[0.001,10]$, the feedback scaling in the interval $[0,1]$, the teacher scaling in the interval $[0.001,10]$, the teacher shift in the interval $[0,10]$, the leaking rate in the interval $[0,1]$. An extra hyperparameter has been tuned, which is the regularization parameters in the least-squares algorithm used to train the weights of the output, which has been searched in the interval $[0,10000]$. The activation function of the hidden layer was set as the $\tanh(.)$ And the activation function of the output layer was ser as the identity $f(x) = x$.

The PSO algorithm (Kennedy and Eberhart, 1995) was set with 20 particles (swarm size), a maximum of 50 generations as stopping criterion, acceleration coefficients c_1=1.5 and c_2=2, inertia factor equal to w=1, and inertia damping factor w_{damp}=0.99. The input sequence of the ESN is the output sequence lagged by 1, 5, or 21 samples, according to the desired forecasting horizon. For example, if the goal is to forecast a length N time series 1-step ahead, then the first $N-1$ samples, i.e. samples 1 to $N-1$, is the input sequence while the last $N-1$ samples, i.e., samples 2 to N, is the output sequence. The implementation of the ESN was developed based on the pyESN library (Korndörfer, 2018).

4.6 Statistical significance test

On top of evaluating the accuracy of the prediction models, it is recommended to check if they are significantly different. A statistical significance test suggested by Demsar (2006) is the Friedman test followed by the *post-hoc* Nemenyi test (Nemenyi, 1963). The Friedman test is a nonparametric test that tests the null hypothesis that all compared models have no significant difference. In the case, the null hypothesis is rejected in the Friedman test it is interesting to know between which models the difference is significant. The Nemenyi test is then used for this task. Equations for the implementation of those methods are provided in Demsar (2006).

In Friedman's test, the models are ranked according to their prediction performance, such that the best performing model gets the rank 1. In the case of ties, the average rank is assigned. Then, the Friedman statistic is calculated as

$$p = \frac{12T}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right], \tag{23}$$

$$R_j = \frac{1}{T}\sum_i r_i^j, \tag{24}$$

where p is Friedman's statistic, T is the number of experiments, k is the number of models compared, and r is the rank.

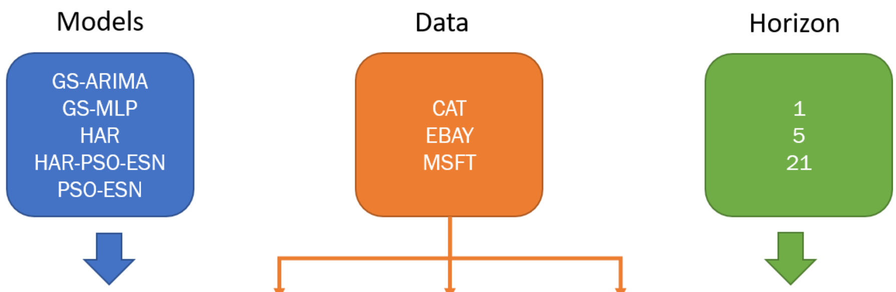
If the p-value is less than 0.05, Friedman's test rejects the hypothesis that all models are equivalent with 95% of confidence, the *post-hoc* Nemenyi test is employed to perform pairwise comparisons among the available methods. If their average ranks are separated for at least a critical distance (CD), the models may be considered statistically different. The CD is calculated as

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6T}}, \tag{25}$$

where CD is the critical distance, q_α is a critical value of the studentized range distribution (Demsar, 2006), k is the number of models compared, and T is the number of experiments.

5 RESULTS AND DISCUSSION

Three time-series of stock price return volatility are considered, and they alternate as training, validation as displayed in Figure 8. The time series are converted to a supervised learning format with the sliding window technique for machine learning models, and the data is scaled before being used by the models. The proposed HAR-PSO-ESN model is compared with the other four models, named GS-ARIMA, GS-MLP, HAR, and PSO-ESN. The predictions are evaluated in terms of accuracy with the R^2 and MSE metrics and of statistical significance with Friedman's test followed by the Nemenyi *post-hoc*.



Experiment	Model	Training set	Validation set	Test set	Horizon
1	GS-ARIMA	CAT	EBAY	MSFT	1
2	GS-ARIMA	MSFT	CAT	EBAY	1
44	PSO-ESN	MSFT	CAT	EBAY	21
45	PSO-ESN	EBAY	MSFT	CAT	21

Figure 8 - Configuration of the experiments.

5.1 Prediction accuracy results

The five models have been tested in each of the three datasets for three different forecasting horizons, resulting in 45 experiments that are evaluated and compared in terms of R^2 and MSE metrics. The average accuracies for 1-day ahead, 5-days ahead, and 21-days ahead are presented in Tables 3, 4, and 5, respectively. The predictions for h days ahead are obtained by selecting the last column of the future values matrix as the output, using a sliding window technique with $H = h$ in Figure 6.

We observe in Tables 3 to 5 that the proposed HAR-PSO-ESN model achieves better prediction accuracies over the test set in terms of highest R^2 and lowest MSE, despite not achieving the best accuracies on the training set. The R^2 and MSE values on the test set for each model and forecasting horizon are also displayed in Figs. 9 and 10, respectively. As expected, the accuracies are better for 1-step ahead forecasts and worsen as the forecasting horizon increases. This behavior is largely expected since longer forecasting horizons involve more uncertainties and affect the predictive capacity of the models. Figs. 9 and 10 also reveal that the proposed HAR-PSO-ESN appears always as the best or second-best model, and its superiority is more salient in the case of the MSFT stock (Figs. 9c and 10c). Finally, we plot in Fig. 11 the time series of the predicted vs. the actual volatilities in the case of the MSFT stock for each of the three forecasting horizons considered.

Table 3 - Average values of volatility prediction metrics for 1-day ahead.

Model	R^2 training set ↑	R^2 test set ↑	MSE training set ↓	MSE test set ↓
GS-ARIMA	0.441	0.553	5.52 x10^{-8}	7.08 x10^{-8}
PSO-ESN	0.634	0.632	5.74 x10^{-8}	5.81 x10^{-8}
HAR-PSO-ESN	0.637	0.635	5.75 x10^{-8}	5.78 x10^{-8}
HAR	0.650	0.633	5.78 x10^{-8}	5.81 x10^{-8}
GS-MLP	0.647	0.625	6.07 x10^{-8}	6.31 x10^{-8}

Table 4 - Average values of volatility prediction metrics for 5-days ahead.

Model	R^2 training set ↑	R^2 test set ↑	MSE training set ↓	MSE test set ↓
GS-ARIMA	0.412	0.444	6.35 x10^{-8}	7.08 x10^{-8}
PSO-ESN	0.485	0.480	8.27 x10^{-8}	5.81 x10^{-8}
HAR-PSO-ESN	0.535	0.510	7.53 x10^{-8}	5.78 x10^{-8}
HAR	0.499	0.481	8.33 x10^{-8}	5.81 x10^{-8}
GS-MLP	0.552	0.496	7.76 x10^{-8}	6.31 x10^{-8}

Table 5 - Average values of volatility prediction metrics for 21-days ahead.

Model	R^2 training set ↑	R^2 test set ↑	MSE training set ↓	MSE test set ↓
GS-ARIMA	0.765	0.222	39.2×10^{-7}	1.26×10^{-7}
PSO-ESN	0.268	0.264	1.18×10^{-7}	1.19×10^{-7}
HAR-PSO-ESN	0.297	0.298	1.11×10^{-7}	1.16×10^{-7}
HAR	0.256	0.224	1.25×10^{-7}	1.27×10^{-7}
GS-MLP	0.308	0.272	1.21×10^{-7}	1.26×10^{-7}

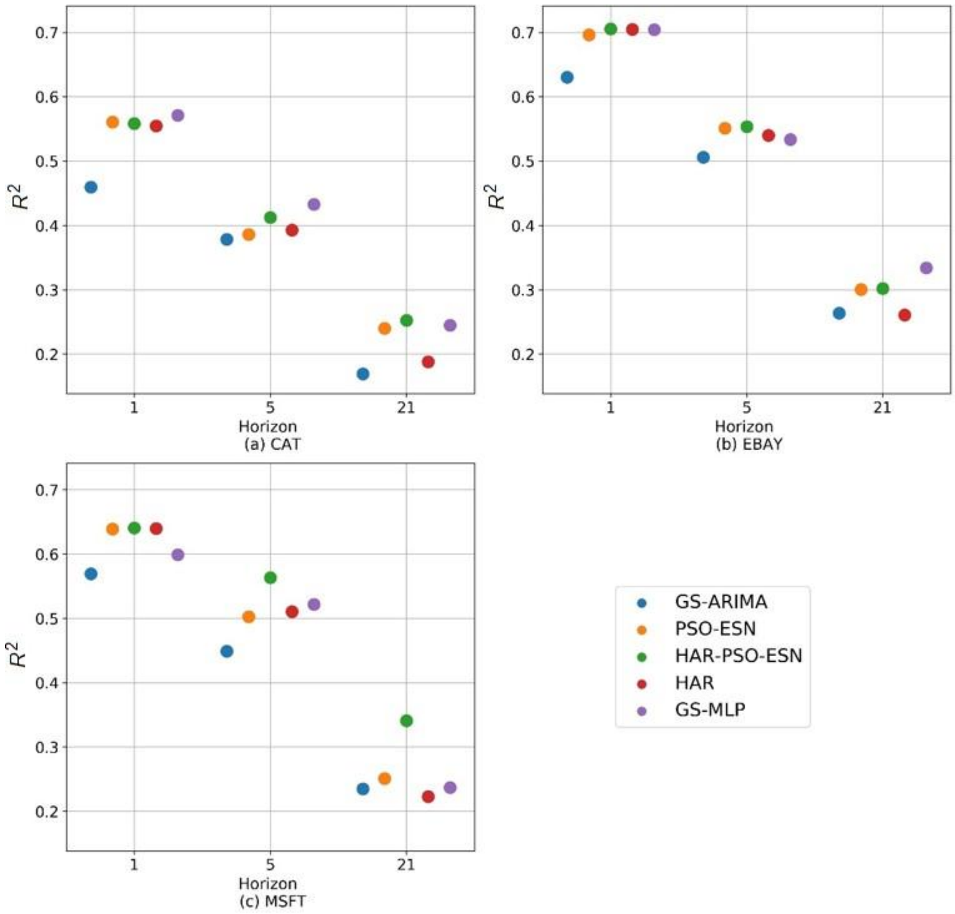


Figure 9 - Swarm plots of R^2 of predictions for (a) CAT, (b) EBAY, and (c) MSFT datasets.

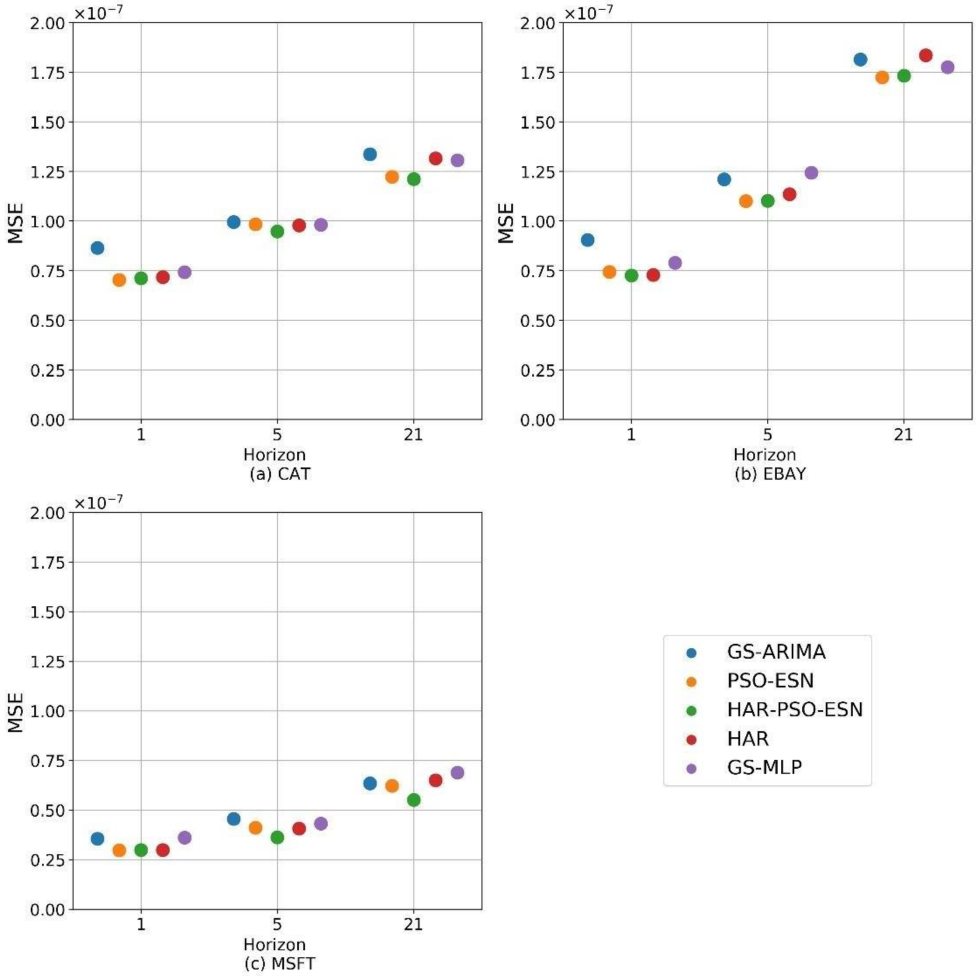


Figure 10 - Swarm plots of MSE of predictions for (a) CAT, (b) EBAY, and (c) MSFT datasets.

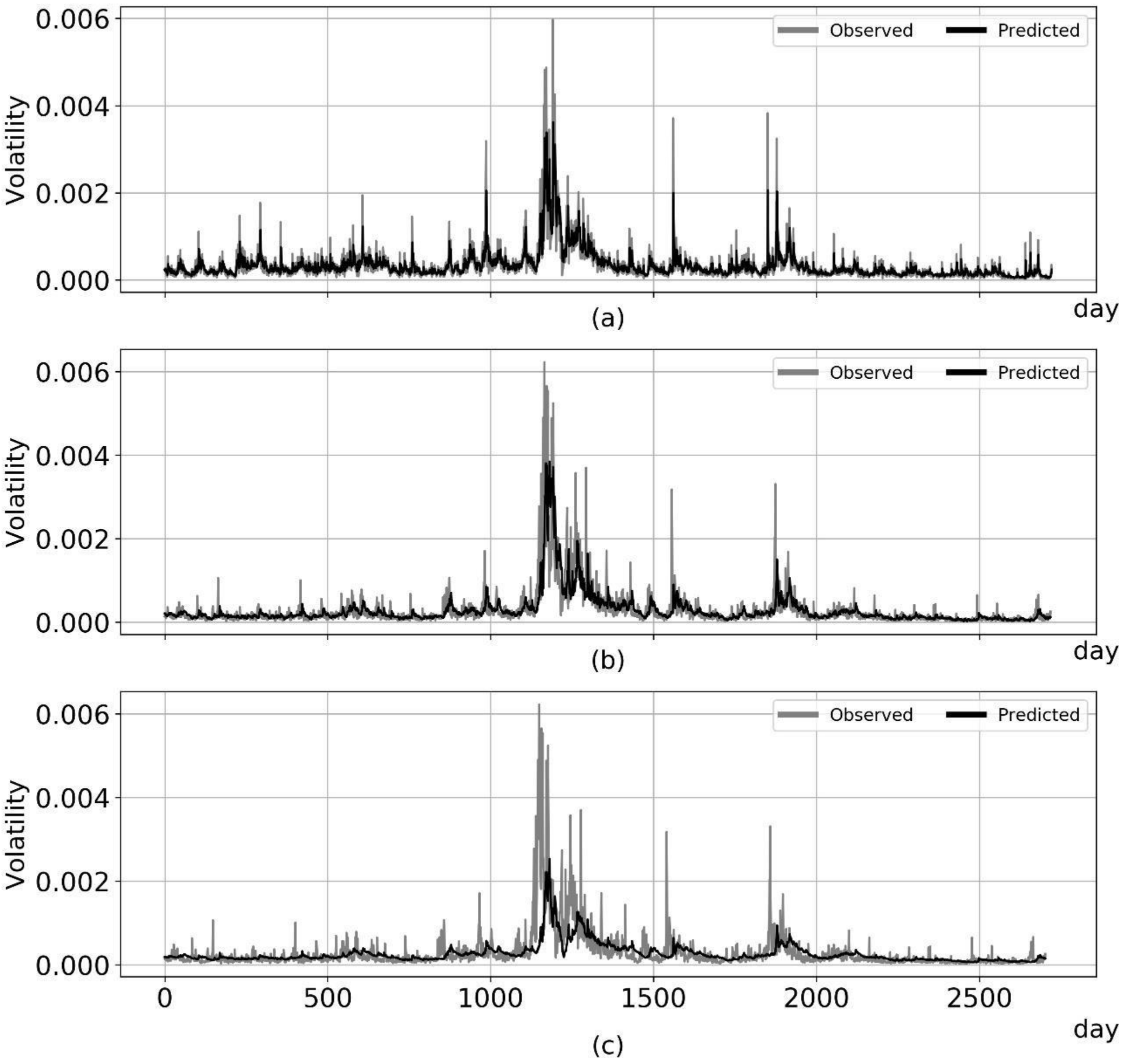


Figure 11 - Comparison of predicted and observed values for (a) EBAY 1-step ahead, (b) MSFT 5-steps ahead, and (c) MSFT 21-steps ahead.

5.2 Statistical significance test

The statistical significance of results is performed through Nemenyi *post-hoc* test after Friedman's rank test (Table 6) considering the results of 45 different experiments (i.e., 5 models, 3 test sets, and 3 forecasting horizons) as illustrated in Figure 8. Significance has been evaluated considering the R^2 (Fig. 10) and the MSE (Fig. 11) performance metrics. The small p-values in Table 5 indicate that there are significant differences in forecasting performance among the models considered.

A prediction model is considered statistically different from another if their ranks differ at least the Nemenyi critical distance. Graphically, there is no statistical evidence to support that the models connected by a thick line in the critical distance (CD) diagram are statistically different. The proposed HAR-PSO-ESN performs significantly better (higher R^2 and lower MSE), with lower ranks in Friedman's test in comparison to all other models (Figs. 12 and 13). However, the ranking for the alternative approaches based on the CD differs depending on whether the forecasting accuracy metric is the R^2 or the MSE.

In the case of the R^2 metric, GS-MLP is the second-best prediction model, and significantly different from the other ones. The PSO-ESN and HAR models presented no significant difference, whereas the GS-ARIMA performs worst. As for the MSE metric, PSO-ESN is the second-better followed by the HAR model, and GS-ARIMA and GS-MLP performed worst, without significant differences from each other.

Table 6 - Friedman's average ranks.

Model	R^2 average rank	MSE average rank
GS-ARIMA	4.78	4.44
PSO-ESN	3.00	2.00
HAR-PSO-ESN	1.44	1.44
HAR	3.44	3.11
GS-MLP	2.33	4.00
p-value	1.67 x10^{-4}	1.02 x10^{-4}

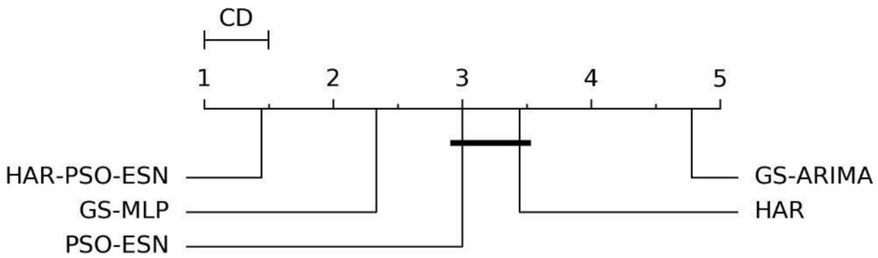


Figure 12 – *CD* of Nemenyi *post-hoc* test considering R^2 metric.

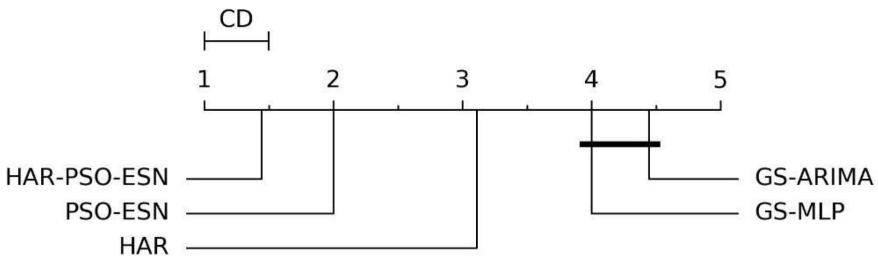


Figure 13 – *CD* of Nemenyi *post-hoc* test considering MSE metric.

6 CONCLUSIONS

Predictions of realized volatilities for three companies listed on the Nasdaq stock exchange have been performed by four existing models as benchmarks, which are ARIMA, HAR, MLP, and ESN, and a novel proposed model named HAR-PSO-ESN. The predictions were obtained for three alternative forecasting horizons: 1-day ahead, 1-week ahead, and 1-mont ahead. The prediction accuracies have been evaluated in terms of the R^2 and MSE performance metrics, and the statistical comparison has been made through a Friedman's test followed by a *post-hoc* Nemenyi test.

The proposed HAR-PSO-ESN model combines the well-established HAR model widely used in econometrics (Vortelinos, 2017; Gong and Lin, 2019) with the emerging ESN recurrent neural network based on reservoir computing paradigm. In the proposed hybrid model, the HAR carefully handcrafted features have been fed into an ESN architecture, which has its hyperparameters tuned by the PSO metaheuristic.

The proposed model delivers more accurate forecasts in comparison to the benchmark models in the vast majority of the cases, and the difference in forecasting accuracy is found to be significant according to the statistical tests performed. The average R^2 (MSE) of the forecasts produced by the proposed HAR-PSO-ESN model on the test is higher (lower) in comparison to the benchmark models in the three forecasting horizons considered.

Future research can consider including additional exogenous features to the proposed model as well as implementing alternative specifications such as the singular spectrum analysis as in Moreno and Coelho (2018), ensemble learning algorithms such as stacked learning, boosting, and bagging (Caldeira *et al.*, 2017; Ribeiro and Coelho, 2020), non-linear system identification techniques (Ayala *et al.*, 2015), expand the set of compared models and datasets, as well as applying different metaheuristics for the ESN hyperparameters tuning, such as the cheetah based optimization algorithm (Klein *et al.*, 2018), the cuckoo optimization algorithm (Rajabioun, 2011; Coelho *et al.*, 2014), the falcon optimization algorithm (Vasconcelos Segundo, Mariani, and Coelho, 2019a), and the owls' optimization algorithm (Vasconcelos Segundo, Mariani, and Coelho, 2019b). The Bayesian optimization of ESN is also a promising approach (Ribeiro *et al.*, 2020). Besides, it is interesting to investigate the performance of deep ESN architectures for stock price volatility prediction.

Finally, it is interesting to consider as future research adopting alternative methods to select the PSO parameters in a prediction context. In this regard, future works may explore this issue building on the works of Armaghani *et al.* (2017), Dehghanbanadaki *et al.* (2020), Huang *et al.* (2020), Harandizadeh *et al*. (2020), and Armaghani *et al.* (2020a). Moreover, a new performance metric named a20-index can be considered in future works to better evaluate and compare the forecasts of different models as in Armaghani *et al.* (2020b).

ACKNOWLEDGEMENTS

The authors Mariani and Coelho would like to thank the National Council of Scientific and Technologic Development of Brazil - CNPq (Grants: 307958/2019-1-PQ, 307966/2019-4-PQ, 405101/2016-3-Univ, 404659/2016-0-Univ) and Fundação Araucária (PRONEX-FA/CNPq 042/2018) for its financial support of this work. Santos acknowledges financial support from the National Council of Scientific and Technologic Development of Brazil - CNPq (Grants 304378/2019-4-PQ and 420038/2018-3-Univ)

REFERENCES

Afkhami, M., Cormack, L. and Ghoddusi, H. (2017) 'Google search keywords that best predict energy price volatility', *Energy Economics*. Elsevier B.V., 67, pp. 17–27. doi: 10.1016/j.eneco.2017.07.014.

Alizadeh, A. H., Huang, C.-Y. and Marsh, I. W. (2019) 'Modelling the volatility of TOCOM energy futures: A regime switching realised volatility approach', *Energy Economics*. Elsevier B.V., (In Press), p. 104434. doi: 10.1016/j.eneco.2019.06.019.

Andersen, T. *et al.* (2000) *Exchange Rate Returns Standardized by Realized Volatility are (Nearly) Gaussian*. Cambridge, MA. doi: 10.3386/w7488.

Andersen, T. G. *et al.* (2001) 'The distribution of realized stock return volatility', *Journal of Financial Economics*. doi: 10.1016/S0304-405X(01)00055-1.

Andersen, T. G., Bollerslev, T. and Meddahi, N. (2005) 'Correcting the errors: Volatility forecast evaluation using high-frequency data and realized volatilities', *Econometrica*, 73(1), pp. 279–296. doi: 10.1111/j.1468-0262.2005.00572.x.

Armaghani, D.J., Mohamad, E.T., Narayanasamy, M.S., Narita, N., et al. (2017) 'Development of hybrid intelligent models for predicting TBM penetration rate in hard rock condition', Tunnelling and Underground Space Technology, 63, p. 29–43. Available from: doi:10.1016/j.tust.2016.12.009.

Armaghani, D.J., Mirzaei, F., Shariati, M., Trung, N.T., et al. (2020a) 'Hybrid ann-based techniques in predicting cohesion of sandy-soil combined with fiber', Geomechanics and Engineering, 20 (3), 191–205. Available from: doi:10.12989/gae.2020.20.3.191.

Armaghani, D.J., Asteris, P.G., Fatemi, S.A., Hasanipanah, M., et al. (2020b) 'On the use of neuro-swarm system to forecast the pile settlement', Applied Sciences (Switzerland), 10 (6). Available from: doi:10.3390/app10061904.

Atkins, A., Niranjan, M. and Gerding, E. (2018) 'Financial news predicts stock market volatility better than close price', *The Journal of Finance and Data Science*. Elsevier Ltd, 4(2), pp. 120–137. doi: 10.1016/j.jfds.2018.02.002.

Ayala, V. H. *et al.* (2015) 'Nonlinear Black-box System Identification through Neural Networks of a Hysteretic Piezoelectric Robotic Micromanipulator', *IFAC-PapersOnLine*. doi: 10.1016/j.ifacol.2015.12.162.

Bustos, O. and Pomares-Quimbaya, A. (2020) 'Stock market movement forecast: A Systematic review', *Expert Systems with Applications*. Elsevier Ltd, 156, pp. 1–15. doi: 10.1016/j.eswa.2020.113464.

Caldeira, J. F. *et al.* (2017) 'Combining multivariate volatility forecasts: An economic-based approach', *Journal of Financial Econometrics*, 15(2), pp. 247–285. doi: 10.1093/jjfinec/nbw010.

Chong, E., Han, C. and Park, F. C. (2017) 'Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies', *Expert Systems with Applications*. Elsevier Ltd, 83, pp. 187–205. doi: 10.1016/j.eswa.2017.04.030.

Choudhury, S. *et al.* (2014) 'A real time clustering and SVM based price-volatility prediction for optimal trading strategy', *Neurocomputing*. Elsevier, 131, pp. 419–426. doi: 10.1016/j.neucom.2013.10.002.

Coelho, L. dos S. *et al.* (2014) 'Optimal chiller loading for energy conservation using a new differential cuckoo search approach', *Energy*. Elsevier Ltd, 75, pp. 237–243. doi: 10.1016/j.energy.2014.07.060.

Corsi, F. (2009) 'A simple approximate long-memory model of realized volatility', *Journal of Financial Econometrics*, 7(2), pp. 174–196. doi: 10.1093/jjfinec/nbp001.

Dan, J. *et al.* (2014) 'Deterministic echo state networks based stock price forecasting', *Abstract and Applied Analysis*, 2014, pp. 1–6. doi: 10.1155/2014/137148.

Degiannakis, S. and Filis, G. (2017) 'Forecasting oil price realized volatility using information channels from other asset classes', *Journal of International Money and Finance*. Elsevier Ltd, 76, pp. 28–49. doi: 10.1016/j.jimonfin.2017.05.006.

Dehghanbanadaki, A., Khari, M., Amiri, S.T. and Armaghani, D.J. (2020) 'Estimation of ultimate bearing capacity of driven piles in c-φ soil using MLP-GWO and ANFIS-GWO models: a comparative study', Soft Computing. [Online] Available from: doi:10.1007/s00500-020-05435-0.

Demsar, J. (2006) 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research*, 7, pp. 1–30. Available at: http://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf.

Fičura, M. (2018) 'Forecasting stock market realized variance with echo state neural networks', *European Financial and Accounting Journal*, 12(3), pp. 145–155. doi: 10.18267/j.efaj.193.

Gong, X. and Lin, B. (2019) 'Modeling stock market volatility using new HAR-type models', *Physica A: Statistical Mechanics and its Applications*. Elsevier B.V., 516, pp. 194–211. doi: 10.1016/j.physa.2018.10.013.

Harandizadeh, H., Armaghani, D.J. and Mohamad, E.T. (2020) 'Development of fuzzy-GMDH model optimized by GSA to predict rock tensile strength based on experimental datasets', Neural Computing and Applications, 32 (17), 14047–14067. Available from: doi:10.1007/s00521-020-04803-z.

Hu, H. *et al.* (2020) 'Effective energy consumption forecasting using enhanced bagged echo state network', *Energy*. Elsevier B.V., 193. doi: 10.1016/j.energy.2019.116778.

Hu, H., Wang, L. and Lv, S. (2020) 'Forecasting energy consumption and wind power generation using deep echo state network', *Renewable Energy*. Elsevier Ltd, 154, pp. 598–613. doi: 10.1016/j.renene.2020.03.042.

Huang, J., Koopialipoor, M. and Armaghani, D.J. (2020) 'A combination of fuzzy Delphi method and hybrid ANN-based systems to forecast ground vibration resulting from blasting', Scientific Reports, 10 (1), 1–21. Available from: doi:10.1038/s41598-020-76569-2.

Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization', in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. Perth, Australia: IEEE, pp. 1942–1948. doi: 10.1109/ICNN.1995.488968.

Kim, H. Y. and Won, C. H. (2018) 'Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models', *Expert Systems with Applications*. Elsevier Ltd, 103, pp. 25–37. doi: 10.1016/j.eswa.2018.03.002.

Klein, C. E. *et al.* (2018) 'Cheetah based optimization algorithm: A novel swarm intelligence paradigm', in *ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning.* Bruges, Belgium. Available at: http://www.i6doc.com/en/.

Korndörfer, C. (2018) 'pyESN'. Available at: https://github.com/cknd/pyESN/blob/master/pyESN.py.

Kristjanpoller, W. and Minutolo, M. C. (2015) 'Gold price volatility: A forecasting approach using the Artificial Neural Network-GARCH model', *Expert Systems with Applications*. Elsevier Ltd, 42 (20), pp. 7245–7251. doi: 10.1016/j.eswa.2015.04.058.

Lv, S.-X., Peng, L. and Wang, L. (2018) 'Stacked autoencoder with echo-state regression for tourism demand forecasting using search query data', *Applied Soft Computing*. Elsevier B.V., 73, pp. 119–133. doi: 10.1016/j.asoc.2018.08.024.

Ma, F. *et al.* (2018) 'Forecasting the aggregate oil price volatility in a data-rich environment', *Economic Modelling*. Elsevier Ltd, 72 (October 2017), pp. 320–332. doi: 10.1016/j.econmod.2018.02.009.

Ma, Q., Shen, L. and Cottrell, G. W. (2020) 'DeePr-ESN: A deep projection-encoding echo-state network', *Information Sciences*. Elsevier Inc., 511, pp. 152–171. doi: 10.1016/j.ins.2019.09.049.

Moreno, S. R. and Coelho, L. S. (2018) 'Wind speed forecasting approach based on singular spectrum analysis and adaptive neuro fuzzy inference system', *Renewable Energy*. Elsevier Ltd, 126, pp. 736–754. doi: 10.1016/j.renene.2017.11.089.

Nemenyi, P. B. (1963) *Distribution-free multiple comparisons*. Princeton University.

Pedregosa, F., Weiss, R. and Brucher, M. (2011) 'Scikit-learn: Machine Learning in Python', *Journal of Machine Learning Research*, (12), pp. 2825–2830.

Pierdzioch, C., Risse, M. and Rohloff, S. (2016) 'A boosting approach to forecasting the volatility of gold-price fluctuations under flexible loss', *Resources Policy*, 47, pp. 95–107. doi: 10.1016/j.resourpol.2016.01.003.

Rajabioun, R. (2011) 'Cuckoo optimization algorithm', *Applied Soft Computing*. Elsevier B.V., 11(8), pp. 5508–5518. doi: 10.1016/j.asoc.2011.05.008.

Ramos-Pérez, E., Alonso-González, P. J. and Núñez-Velázquez, J. J. (2019) 'Forecasting volatility with a stacked model based on a hybridized artificial neural network', *Expert Systems with Applications*, 129, pp. 1–9. doi:10.1016/j.eswa.2019.03.046.

Ribeiro, G. T. *et al.* (2016) 'Short-term load forecasting using wavenet ensemble approaches', in *Proceedings of the International Joint Conference on Neural Networks*. doi: 10.1109/IJCNN.2016.7727272.

Ribeiro, G. T. *et al.* (2020) 'Bayesian optimized echo state network applied to short-term load forecasting', *Energies*, 13(9), p. 2390. doi: 10.3390/en13092390.

Ribeiro, G. T., Mariani, V. C. and Coelho, L. S. (2019) 'Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting', *Engineering Applications of Artificial Intelligence*. Elsevier Ltd, 82(May), pp. 272–281. doi: 10.1016/j.engappai.2019.03.012.

Ribeiro, M. H. D. M. and Coelho, L. S. (2020) 'Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series', *Applied Soft Computing Journal*. Elsevier B.V., 86, p. 105837. doi: 10.1016/j.asoc.2019.105837.

Rigamonti, M. *et al.* (2018) 'Ensemble of optimized echo state networks for remaining useful life prediction', *Neurocomputing*. Elsevier B.V., 281, pp. 121–138. doi: 10.1016/j.neucom.2017.11.062.

Di Sanzo, S. (2018) 'A Markov switching long memory model of crude oil price return volatility', *Energy Economics*. Elsevier B.V., 74, pp. 351–359. doi: 10.1016/j.eneco.2018.06.015.

Seabold, S. and Perktold, J. (2010) 'Statsmodels: Econometric and Statistical Modeling with Python', in *Proceedings of the 9th Pytohn in Science Conference (SCIPY 2010)*. Austin, TX, USA, pp. 92–96.

Vasconcelos Segundo, E. H. de, Mariani, V. C. and Coelho, L. dos S. (2019a) 'Design of heat exchangers using falcon optimization algorithm', *Applied Thermal Engineering*. Elsevier, 156(January), pp. 119–144. doi: 10.1016/j.applthermaleng.2019.04.038.

Vasconcelos Segundo, E. H., Mariani, V. C. and Coelho, L. S. (2019b) 'Metaheuristic inspired on owls behavior applied to heat exchangers design', *Thermal Science and Engineering Progress*. Elsevier Ltd, 14, p. 100431. doi:10.1016/j.tsep.2019.100431.

Vortelinos, D. I. (2017) 'Forecasting realized volatility: HAR against principal components combining, neural networks and GARCH', *Research in International Business and Finance*. Elsevier B.V., 39, pp. 824–839. doi: 10.1016/j.ribaf.2015.01.004.

Walther, T., Klein, T. and Bouri, E. (2019) 'Exogenous drivers of Bitcoin and Cryptocurrency volatility – A mixed data sampling approach to forecasting', *Journal of International Financial Markets, Institutions and Money*. Elsevier B.V., p. 101133. doi: 10.1016/j.intfin.2019.101133.

Wolpert, D.H. and Macready, W.G. (1997) 'No free lunch theorems for optimization', IEEE Transactions on Evolutionary Computation, 1 (1), p. 67–82. Available from: doi:10.1109/4235.585893.

Wolpert D.H. (2002) 'The Supervised Learning No-Free-Lunch Theorems'. In: Roy R., Köppen M., Ovaska S., Furuhashi T., Hoffmann F. (eds) *Soft Computing and Industry*. Springer, London. doi:

10.1007/978-1-4471-0123-9_3

Xia, Y. *et al.* (2018) 'A novel heterogeneous ensemble credit scoring model based on bstacking approach', *Expert Systems with Applications*. Elsevier Ltd, 93, pp. 182–199. doi: 10.1016/j.eswa.2017.10.022.

Xu, M. *et al.* (2020) 'Fault prognostics by an ensemble of Echo State Networks in presence of event based measurements', *Engineering Applications of Artificial Intelligence*. Elsevier Ltd, 87 (October 2019), p. 103346. doi: 10.1016/j.engappai.2019.103346.

Yao, X., Wang, Z. and Zhang, H. (2019) 'Prediction and identification of discrete-time dynamic nonlinear systems based on adaptive echo state network', *Neural Networks*, 113, pp. 11–19. Available from: doi:10.1016/j.neunet.2019.01.003.

Yao, X., Fan, S., Zhao, B. and Cao, S. (2020) 'Controller design based on echo state network with delay output for nonlinear system', Complexity. [Online] 2020. Available from: doi:10.1155/2020/8643029.

Yildiz, I. B., Jaeger, H. and Kiebel, S. J. (2012) 'Re-visiting the echo state property', *Neural Networks*. Elsevier Ltd, 35, pp. 1–9. doi: 10.1016/j.neunet.2012.07.005.

Zhang, H., Liang, J. and Chai, Z. (2013) 'Stock prediction based on phase space reconstruction and echo state networks', *Journal of Algorithms & Computational Technology*, 7(1), pp. 87–100. doi: 10.1260/1748-3018.7.1.87.

Zhang, J. *et al.* (2018) 'Short term electricity load forecasting using a hybrid model', *Energy*. Elsevier B.V., 158, pp. 774–781. doi: 10.1016/j.energy.2018.06.012.