# Multi-step ahead Bitcoin Price Forecasting Based on VMD and Ensemble Learning Methods

Ramon Gomes da Silva*, Matheus Henrique Dal Molin Ribeiro*†, Naylene Fraccanabbia‡,
Viviana Cocco Mariani‡§ and Leandro dos Santos Coelho*§
*Industrial & Systems Engineering Graduate Program (PPGEPS), Pontifical Catholic University of Parana (PUCPR),
Curitiba, Parana, Brazil. 80215–901
†Department of Mathematics, Federal Technological University of Parana (UTFPR),
Pato Branco, Parana, Brazil. 85503–390
‡Mechanical Engineering Graduate Program (PPGEM), Pontifical Catholic University of Parana (PUCPR),
Curitiba, Parana, Brazil. 80215–901
§Department of Electrical Engineering, Federal University of Parana (UFPR),
Curitiba, Parana, Brazil. 81530–000
Emails: gomes.ramon@pucpr.edu.br, matheus.dalmolinribeiro@gmail.com, nayfraccanabbia@outlook.com,
viviana.mariani@pucpr.br, leandro.coelho@pucpr.br

***Abstract*—Bitcoin is the leading currency in the cryptocurrency market capturing attention worldwide. Forecasting the Bitcoin price as accurate as possible is essential, but due to its high volatility this task is challenging. Many researchers try, through the years, to develop efficient models for predicting the Bitcoin price using several different data-driven approaches. The objective of this paper is to develop a novel decomposition-ensemble learning model that combines Variational Mode Decomposition (VMD) and Stacking-ensemble learning (STACK) with machine learning algorithms to forecast the Bitcoin price multi-step ahead. The algorithms are $k$-Nearest Neighbors, Support Vector Regression with Linear kernel, Feed-forward Artificial Neural Network with single-layer perceptron, Generalized Linear Model, and Cubist. Correlation matrix (CORR), principal component analysis (PCA), and Box-Cox transformation (BOXCOX) were used as data preprocessing techniques. Estimating the performance of the proposed models (namely VMD–STACK–CORR, VMD–STACK–PCA, and VMD–STACK–BOXCOX) using relative root mean square error, symmetric mean absolute percentage error, and absolute percentage error measures, defined that for one-day-ahead forecast VMD–STAK–BOXCOX model presented the better performance, and for two and three-days-ahead forecast VMD–STACK–CORR model was chosen, compared to VMD, STACK, and machine learning algorithms models' performance. Diebold-Mariano statistical test was conducted to evaluate a reduction in forecasting errors. Therefore, the proposed models (VMD–STACK–CORR, VMD–STACK–PCA, and VMD–STACK–BOXCOX) indeed forecast accurately Bitcoin price and outperformed the compared models (VMD, STACK, and machine learning models).**

***Index Terms*—Bitcoin price, variational mode decomposition, stacking-ensemble learning, forecasting, time series.**

## I. Introduction

A cryptocurrency is a digital asset designed to work as a medium of exchange that uses strong cryptography to secure financial transactions, control the creation of additional units, and verify the transfer of assets. Cryptocurrencies use decentralized control as opposed to centralized digital currency and central banking systems. [1]. Bitcoin is generally considered the first decentralized cryptocurrency, and it was first released as open-source software in 2009 [2]. With the emergence of the cryptocurrency market, the Bitcoin, its leading currency, has captured global attention [3].

Due to high volatility of Bitcoin [4]–[6], many studies have been conducted through the years using different approaches with the objective to develop an efficient model to predict the Bitcoin price as accurate as possible [3], [7]–[9].

Variational Mode Decomposition (VMD) is a recent advanced multiresolution technique for signal decomposition into a set of sub-signals, where each sub-signals is compact around a center pulsation and has a limited bandwidth [10]. VMD is applied in many different fields such as biomedical [11], electronic [12], industrial material supply [13], and economics/financial [14]–[17].

Moreover, ensemble learning approaches help increasing the accuracy and efficiency of the models which learns different data patterns, combining potentialities of each base (weak) model making them efficient [18]. Stacking-ensemble learning (STACK) is one of the many ways to work with ensemble. STACK combines different prediction models in a single model, working at levels or layers [19]. STACK do predictions of several base learners to compose the stacking layer-0. These predictions are used as inputs in the next layer for a meta-learner on stacking layer-1 [20].

In this respect, the objective of this paper is to develop a heterogeneous stacking-ensemble learning model by using Variational Mode Decomposition (VMD) with machine learning models algorithms to train the intrinsic mode functions (IMFs) generated by decomposition for Bitcoin price forecasting multi-month-ahead (one, two and three-days ahead). The time series is split into five different IMFs and trained each

The authors would like to thank the National Council of Scientific and Technologic Development of Brazil – CNPq (Grants number: 307958/2019-1-PQ, 307966/2019-4-PQ, 404659/2016-0-Univ, 405101/2016-3-Univ), PRONEX '*Fundação Araucária*' 042/2018, and *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* - Brasil (CAPES) - Finance Code 001 for financial support of this work.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

IMF of the VMD with $k$-Nearest Neighbors ($k$-NN), Support Vector Regression (SVR) with Linear kernel, Feed-forward Neural Network (NNET) with single-layer perceptron, and Generalized Linear Model (GLM) as the Base-Learners (weak models), and Cubist as the Meta-Learner (strong model). The predictions of the IMFs in the Layer-0 are summed giving four different predictions, one for each weak model. Those four predictions are used as inputs in the Layer-1, which they were preprocessed using three different techniques: correlation matrix analysis(CORR), principal component analysis (PCA), and Box-Cox transformation (BOXCOX). The three approaches are trained using Cubist as Meta-Learner giving three proposed models, namely VMD–STACK–CORR, VMD–STACK–PCA, and VMD–STACK–BOXCOX, respectively.

The main contributions of this study are: (i) to develop a novel heterogeneous decomposition-ensemble learning model by using VMD combined with stacking-ensemble learning method and machine learning models algorithms; (ii) to compare the proposed decomposition-ensemble model to the decomposition model, the stacking-ensemble model, and the machine learning algorithms to evaluate the performance of the proposed model; (iii) to realize comparisons between the predictions of the models of the multi-step-ahead; (iv) to compare different algorithms to preprocess the data and evaluate their performance; and (v) to present a relevant novel to the cryptocurrency field, as well to time series forecasting multi-step-ahead using VMD decomposition-ensemble model.

The remainder of this paper is structured as follows: Section II-A illustrates the dataset adopted in this paper. Section II-B defines the models used in this paper. Section III details the procedures of the research methodology applied. Section IV presents the results obtained and discussions. Finally, Section V concludes with the final considerations and some proposals of future works.

## II. Material & Methods

This Section presents the description of the material analyzed (Section II-A) as well as the definitions of the models applied in this paper (Section II-B).

### A. Material

The dataset analyzed in this paper refers to the Bitcoin price in United States dollars (US$). The dataset consists of 2045 daily observations from July 18th, 2010 to February 21st, 2016, and it is composed by four variables, as follows in Table I, where the 'Closing Price' is the output and the others are the system inputs.

TABLE I
Inputs and output of the system

| Type | Description | Unit Measure |
|---|---|---|
| Input | Opening Price | US$ |
| Input | High Price | |
| Input | Low Price | |
| Output | Closing Price | |

The data was split into Training and Testing sets in the proportion of 70 and 30%, respectively. In Table II is presented a summary of the statistical indicators of the dataset, which are the Maximum (Max), Minimum (Min), Mean, Median and Standard Deviation (Std). The output was plotted as illustrated in Figure 1. Moreover, the dataset is available at Github Repository [21].

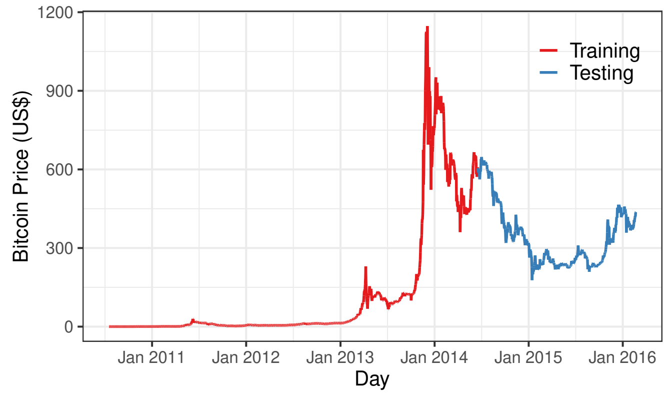



Fig. 1. Daily 'Closing Price' observed over time

### B. Methods

This section presents the main aspects of the methods proposed in this paper. We will present the VMD method and STACK approach, followed by the description of the layers that compose the final ensemble model.

*1) Variational Mode Decomposition:* Variational Mode Decomposition (VMD) is an effective method for signal decomposition. The goal of VMD is to decompose an input data into a finite and predefined $k$ number of Intrinsic Mode Functions (IMF) $u_k(t)$ (1) that reproducing the input signal with different sparsity properties. The IMF $(u_k(t))$ can be defined as

$$u_k(t) = A_k(t) \cos\left(\phi_k(t)\right), \tag{1}$$

where the phase $\phi_k(t)$ is a non-decreasing function, $\phi'_k(t) \geq 0$, the envelope is a non-negative $A_k(t)$ and the instantaneous frequency $\omega_k(t) := \phi'_k(t)$ vary much slower than the phase $\phi_k(t)$ [10].

VMD method relies on three main concepts which are Wiener filtering, Hilbert transform and analytic signal, and frequency mixing and heterodyne demodulation. Sparsity prior of each mode is chosen as bandwidth in the spectral domain and can be accessed by the following scheme for each mode: (i) compute associated analytic signal utilizing the Hilbert transform to obtain a unilateral frequency spectrum; (ii) shift frequency spectrum of mode to baseband by mixing the exponential tune to the respective estimated center frequency; and (iii) the bandwidth estimated through the Gaussian smoothness of the demodulated signal [10], [22], [23].

The specific process of the VMD algorithm is summarized by [10], [24] as follows:

(a) Initialize $\left\{u_k^1\right\}, \left\{\omega_k^1\right\}, \lambda^1$ and $n$;

TABLE II
SUMMARY OF THE STATISTICAL INDICATORS OF THE INPUTS AND OUTPUT OF THE DATASET

| **Variable** | **Samples** | **Percentage** | **# of Samples** | **Statistical indicator** | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | Max | Min | Mean | Median | Std |
| Opening Price | All set | 100% | 2045 | 1147.25 | 0.05 | 188.7906 | 93.105 | 230.1545 |
| | Training set | 70% | 1431 | 1147.25 | 0.05 | 122.9767 | 10.940 | 235.7119 |
| | Testing set | 30% | 614 | 647.34 | 177.28 | 342.0704 | 314.880 | 115.6923 |
| High Price | All set | 100% | 2045 | 1165.89 | 0.06 | 194.5988 | 96.900 | 238.1393 |
| | Training set | 70% | 1431 | 1165.89 | 0.06 | 128.3533 | 11.200 | 246.1440 |
| | Testing set | 30% | 614 | 654.77 | 211.51 | 348.8839 | 321.070 | 116.9866 |
| Low Price | All set | 100% | 2045 | 1110.21 | 0.01 | 182.5121 | 89.445 | 221.1107 |
| | Training set | 70% | 1431 | 1110.21 | 0.01 | 117.0763 | 10.645 | 223.8205 |
| | Testing set | 30% | 614 | 635.53 | 170.21 | 334.9114 | 309.185 | 113.6089 |
| Closing Price | All set | 100% | 2045 | 1147.25 | 0.05 | 189.2220 | 93.330 | 230.2082 |
| | Training set | 70% | 1431 | 1147.25 | 0.05 | 123.8177 | 10.950 | 236.3334 |
| | Testing set | 30% | 614 | 647.34 | 177.28 | 341.5478 | 314.880 | 114.8934 |

(b) Update the value of $\{u_k\}$, $\{\omega_k\}$ and $\lambda$, according to (2), (3) and (4), respectively:

$$u_k^{n+1} \leftarrow \underset{u_k}{\arg\min}\mathcal{L}\left(\left\{u_{i<k}^{n+1}\right\}, \left\{u_{i\geq k}^{n}\right\}, \left\{\omega_i^n\right\}, \lambda^n\right), \quad (2)$$

$$\omega_k^{n+1} \leftarrow \underset{\omega_k}{\arg\min}\mathcal{L}\left(\left\{u_i^{n+1}\right\}, \left\{\omega_{i<k}^{n+1}\right\}, \left\{\omega_{i\geq k}^{n}\right\}, \lambda^n\right), \quad (3)$$

$$\lambda^{n+1} \leftarrow \lambda^n + \tau\left(f - \sum_k u_k^{n+1}\right), \quad (4)$$

where $\lambda$ is Lagrangian multipliers;

(c) Judge whether or not $u_k$ meets the convergence condition (5),

$$\sum_{k=1}^{K} \frac{\left\|u_k^{n+1} - u_k^n\right\|_2^2}{\left\|u_k^n\right\|_2^2} < e, \quad (5)$$

and repeat the above steps to update parameters until the convergence stop condition is satisfied;

(d) The corresponding modal subsequences are obtained according to the given modal number.

*2) Stacking-Ensemble Learning:* Stacking-ensemble learning (STACK) is an ensemble learning method proposed by [25] as stacked generalization, with the purpose to improve prediction accuracy by integrating several diverse sub-models and operates using layers. Simplifying, several base-learners (weak) are combined in the layer-0. The predictions of the base-learners are used as an input set for the meta-learner (strong) in the layer-1. The prediction from the layer-1 is the desired result. In general lines, the model in level-1 learns with the predictions of the models of the level-0. Special attention should be given to the fact that the number of levels need not be restricted to two [19].

*3) Models used in STACK methodology :* This subsection describes the base-learners $k$-NN, SVR, NNET, and GLM, as well as Cubist as meta-learner.

- $k$-NN is an algorithm proposed by [26], which works by mapping $k$ nearest past similar values to new values drives, where $k$ values are named nearest neighbors. A similarity measure is adopted to find the nearest values, where the $k$-nearest neighbors are those that similarity measure between past and new values is the smallest. Then, by calculating the average of past similar values, the future values will be obtained [19].
- SVR consists in determining support vectors close to a hyperplane that maximizes the margin between two-point classes obtained from the difference between the target value and a threshold [27]. To deal with nonlinear problems SVR takes into account kernel (function that calculates the similarity between two observations) functions. In this paper, the linear kernel is adopted.
- NNETs are computing systems vaguely inspired by the biological neural networks that constitute animal brains [28]. NNET is known due to its superiority over traditional regression methods due to its efficient computations, generalization and limited dependence on prior knowledge [29]. An NNET is specified by the information processing unit of the NNET (neuron model), a set of neurons and links connecting neurons (architecture) – each link has a weight and a learning algorithm used for training the NNET by modifying the weights to model a particular learning task correctly on the training examples. In this paper, Single-layer perceptron neural network approach [30], which is a kind of feed-forward neural network, is used.
- GLM, proposed by [31], is a flexible generalization of ordinary linear regression that allows for response variables that have error distribution models other than a normal distribution, which includes in its framework logistic, probit, Poisson and gamma models besides others [32]. The purpose of the GLM is to extend the idea of linear modeling to cases for which the linear relationship between an independent variable and mean response, while the normal distribution is not appropriate for the error distribution [33].
- Cubist is a rule-based algorithm used to construct predictive models based on the analysis of input data. constructs such prediction models using a rule-based model tree

approach. Basically, Cubist branch the data to grow a complete tree; develops of a regression model at each node for pruning and prediction; prune the tree to avoid the overfitting problem; and smooth the tree to compensate for the sharp discontinuities caused by the splitting [34].

## III. Proposed model framework

This section describes the proposed model framework applied in this paper, illustrated in Figure 2.

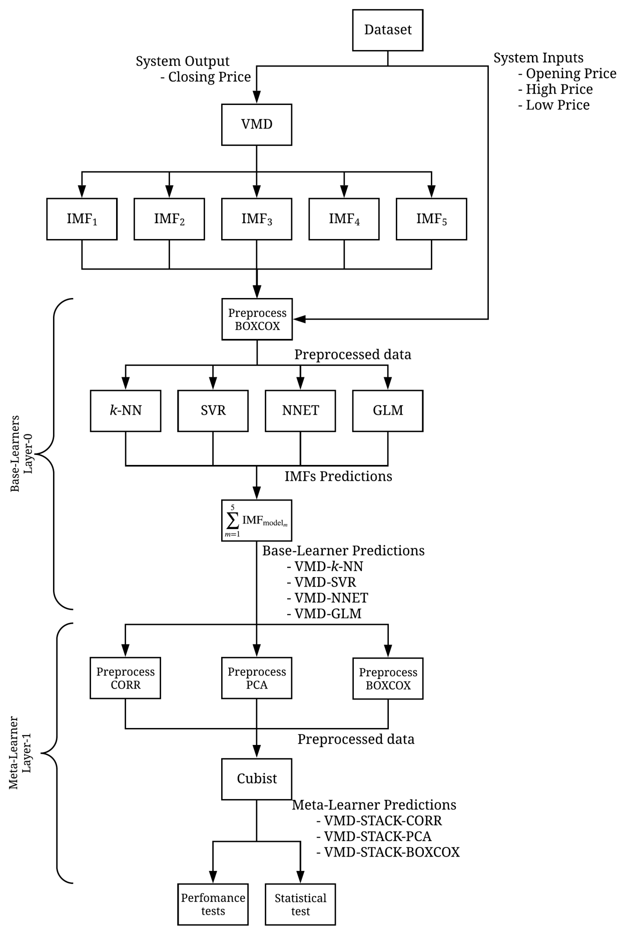



Fig. 2. Framework of the proposed forecasting models

### A. Framework description

**Step 1:** The output variable is decomposed into 5 IMFs by performing VMD, as illustrated on Figure 3;

**Step 2:** The lags equal 1, 2 and 3 for one, two and three-days-ahead were chosen, respectively, and were applied on the IMFs, as well on inputs variables;

**Step 3:** A Box-Cox transformation [35] preprocessing was applied on the IMFs and the inputs;

**Step 4:** Training each IMF with each base-learner model described in Section II-B3 using time-slice validation;

**Step 5:** The IMFs predictions were recomposed by a simple summation grouping by model. In other words, the IMFs

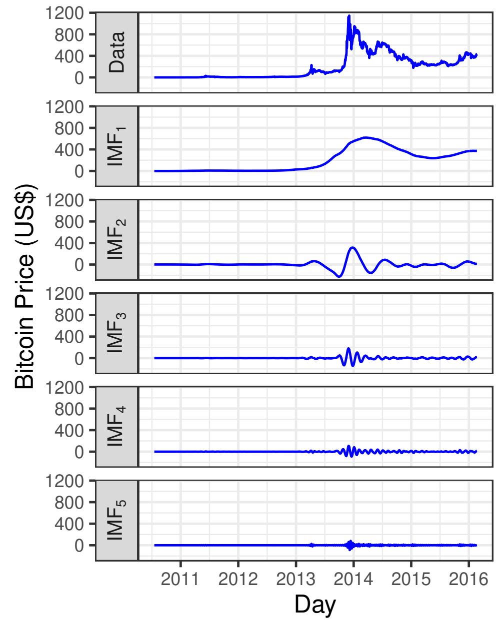



Fig. 3. Data decomposed into IMFs by VMD

trained by the same base-learner model are summed. Then, four predictions were generated namely VMD–$k$-NN, VMD–SVR, VMD–NNET and VMD–GLM;

**Step 6:** The four predictions generated in layer-0 were used as input in the layer-1. They were preprocessed using correlation matrix (CORR) which removes those predictors whose correlation is greater than a threshold [36], principal component analysis (PCA) [37], and Box-Cox transformation (BOXCOX). Training each one using Cubist as meta-learner gives three different final predictions, which are the proposed models, namely VMD–STACK–CORR, VMD–STACK–PCA and VMD–STACK–BOXCOX, respectively.

Table III presents the hyperparameters of the models used in this paper, exception for GLM since it does not have hyperparameters to be defined. A Grid-Search defined the best tunes for the base-learners and meta-learner.

To forecast one (6), two (7) and three-days-ahead (8) the applied structures are defined as,

$$\hat{y}(t+h) = f\left\{y(t+h-1), \mathbf{X}(t+h-1)\right\}, \tag{6}$$

$$\hat{y}(t+h) = f\left\{\hat{y}(t+h-1), y(t+h-2), \mathbf{X}(t+h-2)\right\}, \tag{7}$$

$$\hat{y}(t+h) = f\left\{\hat{y}(t+h-1), \hat{y}(t+h-2), y(t+h-3), \mathbf{X}(t+h-3)\right\}, \tag{8}$$

where $f$ is a function that maps the Bitcoin price, $\hat{y}(t+h)$ is the forecast Bitcoin price in horizon $h = 1, 2, 3$ at time $t$ $(1, \ldots, 2044)$, $y(t+h-1)$, $y(t+h-2)$, $y(t+h-3)$, $\hat{y}(t+h-1)$, $\hat{y}(t+h-2)$ are the previous observed and predicted Bitcoin price, $\mathbf{X}(t+h-n_x)$ is the inputs vector composed by Opening, High and Low Price at the maximum lag of inputs ($n_x = 1$ if $h = 1$, $n_x = 2$ if $h = 2$, and $n_x = 3$ if $h = 3$).

**Step 7:** Performance (Section III-B) and statistical tests (Section III-C) were conducted to evaluate the accuracy of the

TABLE III
CONTROL HYPERPARAMETERS FOR THE MODELS

| Base-learner model | Components | Control Hyperparameters | | Meta-learner model | Forecasting Horizon | Control Hyperparameters | |
|---|---|---|---|---|---|---|---|
| | | # of Neighbors | | | One-day-ahead | # of Committees | # of Instances |
| $k$-NN | $IMF_1$ | 7 | | Cubist | VMD–STACK–CORR | 20 | 0 |
| | $IMF_2$ | 5 | | | VMD–STACK–PCA | 1 | 9 |
| | $IMF_3$ | 9 | | | VMD–STACK–BOXCOX | 1 | 9 |
| | $IMF_4$ | 9 | | | STACK–CORR | 1 | 0 |
| | $IMF_5$ | 9 | | | STACK–PCA | 20 | 0 |
| | Nondecomposed | 7 | | | STACK–BOXCOX | 10 | 0 |
| | | Cost | Kernel | | Two-days-ahead | # of Committees | # of Instances |
| SVR | $IMF_1$ | 1 | Linear | Cubist | VMD–STACK–CORR | 1 | 5 |
| | $IMF_2$ | 1 | | | VMD–STACK–PCA | 20 | 5 |
| | $IMF_3$ | 1 | | | VMD–STACK–BOXCOX | 1 | 0 |
| | $IMF_4$ | 1 | | | STACK–CORR | 10 | 0 |
| | $IMF_5$ | 1 | | | STACK–PCA | 20 | 0 |
| | Nondecomposed | 0.25 | | | STACK–BOXCOX | 10 | 0 |
| | | # of Hidden Units | Weight Decay | | Three-days-ahead | # of Committees | # of Instances |
| NNET | $IMF_1$ | 1 | 1e-04 | Cubist | VMD–STACK–CORR | 1 | 5 |
| | $IMF_2$ | 5 | 0.1 | | VMD–STACK–PCA | 20 | 5 |
| | $IMF_3$ | 5 | 0.1 | | VMD–STACK–BOXCOX | 1 | 0 |
| | $IMF_4$ | 3 | 0 | | STACK–CORR | 10 | 5 |
| | $IMF_5$ | 1 | 1e-04 | | STACK–PCA | 10 | 0 |
| | Nondecomposed | 5 | 0.1 | | STACK–BOXCOX | 20 | 0 |

proposed models compared to (i) different preprocessing algorithms applied in the layer-1 phase; (ii) VMD models without STACK method; (iii) STACK models without decomposition and different preprocessing algorithms; and (iv) the models applied directly to the dataset.

### B. Performance indicators

To evaluate the performance of the proposed models, relative root mean square error (RRMSE) (9), symmetric mean absolute percentage error (sMAPE) (10) and absolute percentage error (APE) (11) were performed, and they are described as follows

$$\text{RRMSE} = \frac{\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{\frac{1}{n}\sum_{i=1}^{n} y_i}, \tag{9}$$

$$\text{sMAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{(|y_i| + |\hat{y}_i|/2)}\right|, \tag{10}$$

$$\text{APE} = \frac{|y_i - \hat{y}_i|}{y_i}, \tag{11}$$

where $n$ is the number of observations, $y_i$ and $\hat{y}_i$ are the *i*-th observed and predicted values, respectively.

### C. Diebold-Mariano test

With the objective of comparing the forecast errors of the models, a Diebold-Mariano (DM) test [38] is conducted. The DM test will verify whether the forecasting errors are lower in relation to each other. A hypothesis test is given by (12), where null hypothesis ($H_0$) says that there is no difference between the forecasting errors of the models compared, and the alternative hypothesis ($H_1$) says that forecasting error of the model proposed is lower than the compared one. The hypothesis test can be defined as follows,

$$H : \begin{cases} H_0 : \epsilon_i^P = \epsilon_i^C \\ H_1 : \epsilon_i^P < \epsilon_i^C, \end{cases} \tag{12}$$

and statistic of DM test is given by (13),

$$\text{DM} = \frac{\frac{\sum_{n}^{i=1}[d_i]}{n}}{\sqrt{\frac{\text{var}(d_i)}{n-1}}}, \tag{13}$$

where $d_i = L(\epsilon_i^P) - L(\epsilon_i^C)$, $L$ is a loss function that estimates the accuracy of each model, $\epsilon_i^P$ and $\epsilon_i^C$ are the error of the proposed model and the compared model, respectively, var$(d_i)$ is an estimate for the variance of $d_i$. By using the hypothesis defined, the aim is to know if the errors for the proposed model are lower than the compared model. If the null hypothesis is rejected, it is possible to say that there is statically evidence that there is a reduction in the errors of the proposed model regarding the compared model at the $\alpha$ level of significance.

## IV. RESULTS

The performance measures results of the models for predicting one, two and three-days-ahead are shown on Table IV. The best results are stated in bold. The models were listed in the Table IV to facilitate comprehension further. For the one-day-ahead prediction, model (C) presented a better performance in both indicators. On two-days-ahead forecasting, models (A) and (B) presented better results in sMAPE and RRMSE, respectively. Finally, for three-days-ahead, models (A) and (K) were more accurate according to sMAPE and RRMSE, respectively. Further, Figure 4 illustrates a violin plot of the fourteen models APE for the three forecasting horizons. By this plot, it shows that model (A) presented error signals with less variability.

Most of the VMD models (models (D) to (G)) presented better results than stand-alone machine learning models (models (K) to (N)). In contrast, STACK models (models (H) to (J)) outperformed all VMD models. However, the combination of

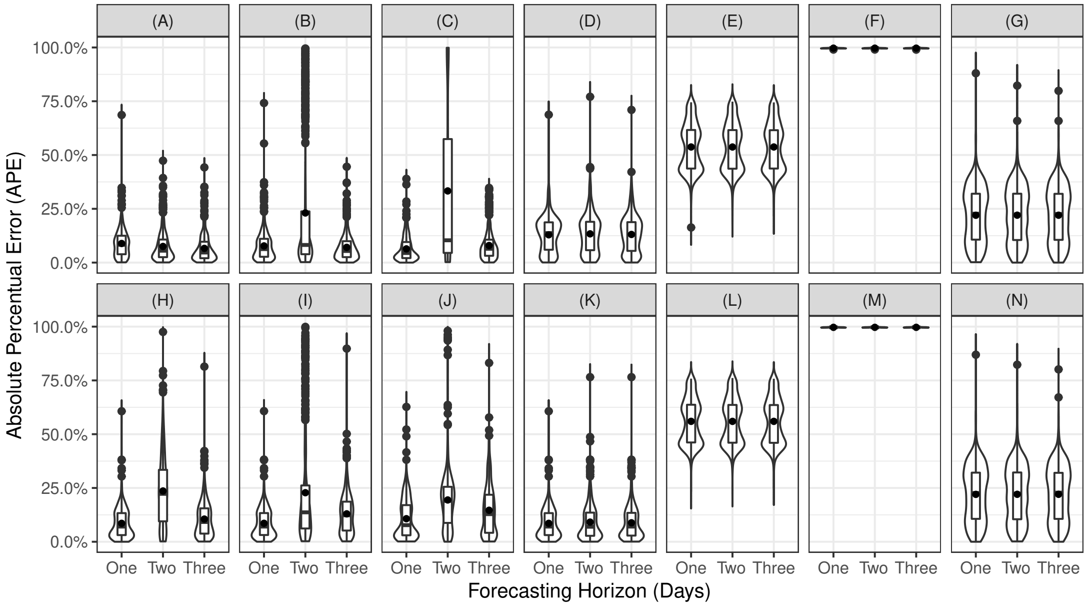



Fig. 4. Violin plot for the models APE

TABLE IV
PERFORMANCE MEASURES RESULTS OF THE MODELS

| Model | Forecasting Horizon | | | | | |
|---|---|---|---|---|---|---|
| | One-day-ahead | | Two-days-ahead | | Three-days-ahead | |
| | sMAPE | RRMSE | sMAPE | RRMSE | sMAPE | RRMSE |
| (A) VMD–STACK–CORR* | 0.0835 | 0.0936 | **0.0660** | 0.0943 | **0.0762** | 0.1060 |
| (B) VMD–STACK–PCA* | 0.0735 | 0.0915 | 0.0708 | **0.0934** | 0.2310 | 0.3812 |
| (C) VMD–STACK–BOXCOX* | **0.0626** | **0.0800** | 0.0836 | 0.1003 | 0.5766 | 0.7108 |
| (D) VMD–$k$-NN** | 0.1189 | 0.1244 | 0.1201 | 0.1274 | 0.1224 | 0.1304 |
| (E) VMD–SVR** | 0.7506 | 1.5222 | 0.7499 | 1.5187 | 0.7497 | 1.5184 |
| (F) VMD–NNET** | 1.9870 | 355.3495 | 1.9870 | 354.8142 | 1.9870 | 354.7863 |
| (G) VMD–GLM** | 0.2153 | 0.2867 | 0.2151 | 0.2858 | 0.2151 | 0.2858 |
| (H) STACK–CORR* | 0.0804 | 0.0935 | 0.0976 | 0.1099 | 0.2033 | 0.1954 |
| (I) STACK–PCA* | 0.0806 | 0.0936 | 0.1185 | 0.1263 | 0.2169 | 0.3395 |
| (J) STACK–BOXCOX* | 0.0985 | 0.1131 | 0.1315 | 0.1596 | 0.3608 | 0.5100 |
| (K) $k$-NN** | 0.0804 | 0.0935 | 0.0824 | 0.0985 | 0.0850 | **0.1028** |
| (L) SVR** | 0.7925 | 1.6495 | 0.7917 | 1.6458 | 0.7916 | 1.6456 |
| (M) NNET** | 1.9871 | 359.3776 | 1.9871 | 358.8427 | 1.9871 | 358.8427 |
| (N) GLM** | 0.2154 | 0.2863 | 0.2152 | 0.2854 | 0.2152 | 0.2854 |

**Note:** *Cubist as meta-learner; **BoxCox as pre-processing.

VMD + STACK (models (A) to (C)) presented a significant improvement compared to others. This combination creates a powerful model, due to the ensemble-learning principle of divide-to-conquer, where the final model (ensemble on this case) seeks to overcome the performance of a base model that operates in isolation, and the characteristics of VMD. Even though some VMD models presented bad performance (e.g. model (F)), the combination of the weak models and then performing a strong model make a relevant difference in the model accuracy when comparing VMD-STACK and VMD models. Yet, it was observed that dimensionality reduction techniques (i.e. CORR and PCA) did not significantly improve the predictions, once that the difference between models (A), (B) and (C) performances, and between models (H), (I) and (J) performance are almost irrelevant.

Analysing the performance results, it is possible to calculate the percentage reduction of the indicators in relation to the more accurate model (bold results in Table IV).

Firstly, comparing the VMD–STACK models: (i) for forecasting horizon of one day, in relation to sMAPE and model (C) as reference, the models (A) and (B) were reduced in 25.03 and 14.83%, respectively. In relation to RRMSE and model (C), the reduction of model (A) were 14.53% and model (B) 12.57%; (ii) for two days of forecasting horizon, in relation to sMAPE and model (A), the models (B) and (C) respectively reduced in 6.78% and 21.05%. In relation to RRMSE and model (B), the models (A) and (C) were reduced 0.95% and 6.88%; and (iii) for three-days-ahead predictions, in relation to sMAPE and model (A), the reduction of models (B) and (C) were 67.01% and 86.78%. And, in relation to RRMSE and model (K) as reference, the models (A), (B) and (C) were reduced in 3.02, 73.03 and 85.54%, in the order.

Second, comparing the VMD models: (i) for one-day-ahead forecasting, in relation to sMAPE and model (C), the models (D) to (G) reduced in the range of 47.35%–96.85%, being model (D) the smaller and (F) the bigger. In relation to RRMSE and model (C), the models reduction were in the range of 35.69% and 99.98%; (ii) for two-days-ahead, in relation to sMAPE and model (A), the reduction range varies from 45.05% (model (D)) to 96.68% (model (F)). For RRMSE and model (B) as reference, the models reduction varies from 26.69% to 99.97%; and (iii) for three-days-ahead, in relation to sMAPE and model (A), the VMD models reduction varies in a range of 37.75% to 96.17%. And in relation to RRMSE and model (K), model (D) presented the smaller reduction of 21.17% and model (F) the bigger of 99.97%.

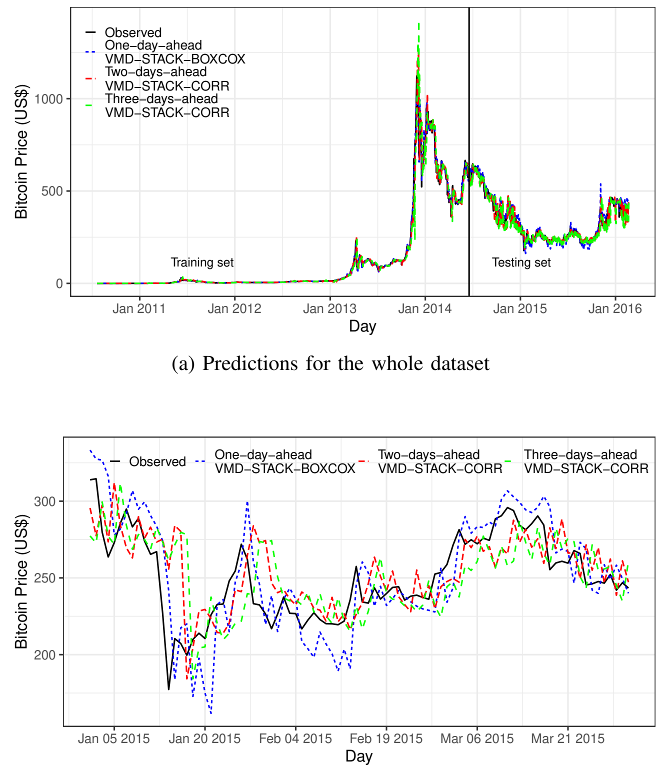
Third, comparing the STACK models: (i) for one-day-ahead predictions, sMAPE and model (C), the reduction of STACK models were 22.14%, 22.33% and 36.45%, respectively. For RRMSE, the reductions were 14.44%, 14.53% and 29.27%; (ii) for two-days-ahead, sMAPE and model (A), models (H), (I) and (J) reduced in 32.38%, 44.30% and 49.81%, respectively. For RRMSE and model (B), the reductions of the models were reduced in 15.01%, 26.05% and 41.48%; and (iii) for three-days-ahead, sMAPE and model (A), the models reduced in 62.52%, 64.87% and 78.88%. And for RRMSE and model (K) the reductions were 47.39%, 69.72% and 79.84%.

Last, comparing the machine learning models, and knowing that in all forecasting horizons and performance indicators the model (K) and the model (M) were the best and the worse of the machine learning models, respectively. Knowing this: (i) for predictions of one-day-ahead, in relation to sMAPE and model (C), the models reductions were in a range of 22.14% and 96.85%. In relation to RRMSE, the reductions were in a range of 14.44% and 99.98%; (ii) for two-days-ahead, sMAPE and model (A) as reference, the reductions were in a range 19.90%–96.68%. And for RRMSE and model (B), the reductions were in a range 5.18%–99.97%; and (iii) for three-days-ahead, sMAPE and model (A), the models reductions were in a range of 10.35% to 96.17%. And for RRMSE and model (K) as reference, the range of the reductions were 63.98%–99.97%.

In this respect, based on the metrics results, the predictions for each forecasting horizon was chosen, where model (C) for one-day-ahead, and model (A) for two and three-days-ahead. Figure 5a illustrates the predictions for the whole dataset, and Figure 5b is a window of the dataset illustrating the samples from January 1st, 2015 to March 31st, 2015, where there are the Observed is the 'Closing Price' (black line), the predictions for One-day-ahead (blue dotted line), Two-days-ahead (red dotted line), and Three-days-ahead (green dotted line). The plots show that the proposed models indeed are accurate and they have learned the dynamism of the time-series.

Moreover, DM tests were conducted to compare the proposed models ((A), (B) and (C)) with the other models for each forecasting horizon from one to three-days-ahead. Table V presents the DM-values, calculated as (13).

Furthermore, DM test results shows that, for one-day-ahead forecasting, model (A) is statistically equal to models (B), (H), (I) and (K), model (B) is statistically equal to model (J), and model (C) is statistically equal to models (H), (I) and (K). For two-days-ahead forecasting, models (A), (B), (C) and (K) are statistically equal. And, for three-days-forecasting, models (A) and (K), and models (B) and (C) are statistically equal. It is important to emphasize that even though errors of some models are statistically the same, the results obtained by using them are not.




(a) Predictions for the whole dataset



(b) Predictions for samples from Jan. 1st 2015 to Mar. 31st 2015

Fig. 5. Bitcoin price predictions for multistep-ahead

## V. Conclusion

Cryptocurrencies' price forecasting interest, in specific the Bitcoin price, has been growing in the last years, due to the boom of the cryptocurrency market. This influences to emerge many studies in the field to develop accurate prediction models. In this context, this study proposed a novel heterogeneous decomposition-ensemble learning model by using VMD and STACK with different preprocessing algorithms to forecast Bitcoin price multistep-ahead. These models were built by decomposing data into IMFs, training them using diverse weak models, recomposing the data, preprocessing with three different algorithms, and then training the base (weak) models with a meta (strong) one. The proposed models are namely VMD–STACK–CORR, VMD–STACK–PCA and VMD–STACK–BOXCOX.

To evaluate the effectiveness of this approach, the VMD–STACK models performance were compared to VMD models, STACK models and base models performance. Further, DM tests were conducted to evaluate the forecasting errors of the VMD–STACK approach regarding the other models. The results indicate that VMD–STACK approach performs better than the techniques individually applied and the base learners. In all forecasting horizons presented in this paper, the proposed methodology showed to be the most accurate.

As future researches, it is proposed to (i) test different base models, as well as its quantity, and different meta-model; (ii) optimize the hyperparameters of base and meta learners; (iii) optimize the number of IMFs to be decomposed; (iv) decompose time series using different decomposition method; and (v) enlarge the forecasting horizon to over 3 days ahead.

## References

[1] A. Greenberg, "Crypto currency," April 2011. [Online]. Available: https://www.forbes.com/forbes/2011/0509/technology-psilocybin-bitcoins-gavin-andresen-crypto-currency.html#45c1577b353e

[2] K. Sagona-Stophel, "Bitcoin 101: How to get started with the new trend in virtual currencies," *White Paper, Thompson Reuters*, 2013.

TABLE V
DIEBOLD-MARIANO TEST RESULTS

| Model | (A) | (B) | (C) | (D) | (E) | (F) | (G) | (H) | (I) | (J) | (K) | (L) | (M) | (N) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | One-day-ahead | | | | | | | | | | | | | |
| (A) | - | -1.1808 | 3.1101* | -7.3984* | -12.0835* | -16.5659* | -8.3376* | 1.8446 | 1.8281 | -3.7099* | 1.8446 | -12.2978* | -16.5617* | -8.3427* |
| (B) | - | - | 3.0229* | -5.8826* | -12.0814* | -16.5654* | -8.3103* | 1.9772** | 1.9679** | -1.6512 | 1.9772** | -12.2958* | -16.5612* | -8.3153* |
| (C) | - | - | - | -8.0185* | -12.0883* | -16.5674* | -8.3892* | -0.9842 | -0.9961 | -4.9253* | -0.9842 | -12.3022* | -16.5632* | -8.3948* |
| | Two-days-ahead | | | | | | | | | | | | | |
| (A) | - | 0.2929 | -0.2391 | -5.5822* | -6.9657* | -9.5660* | -4.8075* | -2.5678** | -4.5894* | -6.8380* | -0.9036 | -7.0893* | -9.5635* | -4.8115* |
| (B) | - | - | -0.4671 | -5.6608* | -6.9654* | -9.5658* | -4.8066* | -2.8868* | -4.9356* | -6.9963* | -1.1455 | -7.0891* | -9.5634* | -4.8106* |
| (C) | - | - | - | -4.6239* | -6.9646* | -9.5655* | -4.7984* | -1.9949** | -3.9650* | -6.4008* | -0.5822 | -7.0882* | -9.5631* | -4.8024* |
| | Three-days-ahead | | | | | | | | | | | | | |
| (A) | - | -13.8752* | -7.8720* | -3.5211* | -5.4017* | -7.4190* | -3.7026* | -7.0480* | -27.4156* | -13.8618* | 0.1871 | -5.4978* | -7.4171* | -3.7051* |
| (B) | - | - | -1.5640 | 13.9756* | -2.1187** | -6.4100* | 10.3861* | 13.5846* | 5.7783* | -35.6064* | 13.8949* | -2.4337** | -6.4087* | 10.4037* |
| (C) | - | - | - | 7.8447* | -17.5507* | -7.6499* | 8.9298* | 7.6728* | 4.0853* | -12.5259* | 7.8699* | -6.6571* | -7.6474* | 8.9214* |

**Note:** *1% significance level; **5% significance level.


[3] R. Adcock and N. Gradojevic, "Non-fundamental, non-parametric Bitcoin forecasting," *Physica A: Statistical Mechanics and its Applications*, vol. 531, no. 121727, 2019.

[4] A. Phillip, J. Chan, and S. Peiris, "On generalized bivariate student-t gegenbauer long memory stochastic volatility models with leverage: Bayesian forecasting of cryptocurrencies with a focus on bitcoin," *Econometrics and Statistics*, 2018.

[5] M. Yu, "Forecasting bitcoin volatility: The role of leverage effect and uncertainty," *Physica A: Statistical Mechanics and its Applications*, vol. 533, no. 120707, 2019.

[6] G. Köchling, P. Schmidtke, and P. N. Posch, "Volatility forecasting accuracy for bitcoin," *Economics Letters*, no. 108836, 2019.

[7] X. Sun, M. Liu, and Z. Sima, "A novel cryptocurrency price trend forecasting model based on lightgbm," *Finance Research Letters*, 2018.

[8] G. S. Atsalakis, I. G. Atsalaki, F. Pasiouras, and C. Zopounidis, "Bitcoin price forecasting with neuro-fuzzy techniques," *European Journal of Operational Research*, vol. 276, no. 2, pp. 770–780, 2019.

[9] D. C. Mallqui and R. A. Fernandes, "Predicting the direction, maximum, minimum and closing prices of daily bitcoin exchange rate using machine learning techniques," *Applied Soft Computing*, vol. 75, pp. 596–606, 2019.

[10] D. Konstantin and Z. Dominique, "Variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 531–544, 2014.

[11] M. M. Rahman, M. I. H. Bhuiyan, and A. B. Das, "Classification of focal and non-focal eeg signals in vmd-dwt domain using ensemble stacking," *Biomedical Signal Processing and Control*, vol. 50, pp. 72–82, 2019.

[12] G. Baldini, G. Steri, R. Giuliani, and F. Dimc, "Radiometric identification using variational mode decomposition," *Computers & Electrical Engineering*, vol. 76, pp. 364–378, 2019.

[13] Y. Liu, C. Yang, K. Huang, and W. Gui, "Non-ferrous metals price forecasting based on variational mode decomposition and lstm network," *Knowledge-Based Systems*, no. 105006, 2019.

[14] S. Lahmiri, "Long memory in international financial markets trends and short movements during 2008 financial crisis based on variational mode decomposition and detrended fluctuation analysis," *Physica A: Statistical Mechanics and its Applications*, vol. 437, pp. 130–138, 2015.

[15] ——, "A variational mode decompoisition approach for analysis and forecasting of economic and financial time series," *Expert Systems with Applications*, vol. 55, pp. 268–273, 2016.

[16] Y. Jiang, H. Nie, and J. Y. Monginsidi, "Co-movement of asean stock markets: New evidence from wavelet and vmd-based copula tests," *Economic Modelling*, vol. 64, pp. 384–398, 2017.

[17] R. Bisoi, P. Dash, and A. Parida, "Hybrid variational mode decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis," *Applied Soft Computing*, vol. 74, pp. 652–678, 2019.

[18] M. H. D. M. Ribeiro, V. H. A. Ribeiro, G. Reynoso-Meza, and L. S. Coelho, "Multi-objective ensemble model for short-term price forecasting in corn price time series," in *International Joint Conference on Neural Networks (IJCNN)*, Budapest, Hungary, July 2019, pp. 1–8.

[19] M. H. D. M. Ribeiro and L. S. Coelho, "Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series," *Applied Soft Computing*, vol. 86, no. 105837, 2020.

[20] S. R. Moreno, R. G. Silva, M. H. D. M. Ribeiro, N. Fraccanabbia, V. C. Mariani, and L. S. Coelho, "Very short-term wind energy forecasting based on stacking ensemble," in *14th Brazilian Computational Intelligence Meeting (CBIC)*, Belém, Brazil, November 2019, pp. 1–7.

[21] B. K. King, "Time Series in R," June 2016. [Online]. Available: https://github.com/ByronKKing/Time-Series-R/blob/master/bitcoin-2016-02-21.csv

[22] M. Firdaus Isham, M. Salman Leong, M. Hee Lim, and Z. Asrar Ahmad, "Variational mode decomposition: mode determination method for rotating machinery diagnosis," *Journal of Vibroengineering*, vol. 20, no. 7, 2018.

[23] N. ur Rehman and H. Aftab, "Multivariate variational mode decomposition," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 6039–6052, 2019.

[24] G. Li, X. Ma, and H. Yang, "A hybrid model for forecasting sunspots time series based on variational mode decomposition and backpropagation neural network improved by firefly algorithm," *Computational Intelligence and Neuroscience*, vol. 2018, no. 3713410, 2018.

[25] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[26] T. M. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[28] Y.-Y. Chen, Y.-H. Lin, C.-C. Kung, M.-H. Chung, I. Yen *et al.*, "Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes," *Sensors*, vol. 19, no. 9, 2019.

[29] M. Ahmad, K. Alam, S. Tariq, S. Anwar, J. Nasir, and M. Mansha, "Estimating fine particulate concentration using a combined approach of linear regression and artificial neural network," *Atmospheric Environment*, vol. 219, no. 117050, 2019.

[30] F. Rosenblatt, "Principles of neurodynamics: perceptrons and the theory of brain mechanisms," Cornell Aeronautical Lab Inc Buffalo NY, Tech. Rep., 1961.

[31] J. A. Nelder and R. W. M. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[32] O. Idais, "A note on locally optimal designs for generalized linear models with restricted support," *Statistics & Probability Letters*, vol. 159, no. 108692, 2020.

[33] X.-W. Liu and D.-G. Lu, "Survival analysis of fatigue data: Application of generalized linear models and hierarchical bayesian model," *International Journal of Fatigue*, vol. 117, pp. 39–46, 2018.

[34] H. Nguyen, X.-N. Bui, Q.-H. Tran, and N.-L. Mai, "A new soft computing model for estimating and controlling blast-produced ground vibration based on hierarchical k-means clustering and cubist algorithms," *Applied Soft Computing*, vol. 77, pp. 376–386, 2019.

[35] G. E. Box and D. R. Cox, "An analysis of transformations," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 26, no. 2, pp. 211–243, 1964.

[36] M. Kuhn and K. Johnson, *Applied predictive modeling*. Springer, 2013, vol. 26.

[37] J. E. Jackson, *A user's guide to principal components*. New York, USA: John Wiley & Sons, 1991.

[38] F. X. Diebold and R. S. Mariano, "Comparing predictive accuracy," *Journal of Business and Economic Statistics*, vol. 13, no. 3, pp. 253–263, 1995.