

Tuning Parameters in Deep Belief Networks for Time Series Prediction through Harmony Search

Do Ngoc Luu, Nguyen Ngoc Phien, and Duong Tuan Anh

Abstract—There have been several researches of applying Deep Belief Networks (DBNs) to predict time series data. Most of these works pointed out that DBNs can bring out better prediction accuracy than traditional Artificial Neural Networks. However, one of the main shortcomings of using DBNs in time series prediction concerns with the proper selection of their parameters. In this paper, we investigate the use of Harmony Search algorithm for determining the parameters of DBN in forecasting time series. Experimental results on several synthetic and real world time series datasets revealed that the DBN with parameters selected by Harmony Search performs better than the DBN with parameters selected by Particle Swarm Optimization (PSO) or random method in most of the tested datasets.

Index Terms—Deep belief network, parameters, time series prediction, harmony search.

I. INTRODUCTION

Time series prediction is an important area of prediction in which past values of the same variable are collected and analyzed to develop a model describing the underlying relationship. The model is then used to extrapolate the time series in the future. There exist some well-known methods of time series prediction, such as ARIMA, exponential smoothing, artificial neural networks (ANNs), k-nearest-neighbors algorithm and support vector machines (SVMs). Among these methods, ANN is the most popular methods for time series prediction.

Deep neural network models, such as Deep Belief Networks (DBNs), have recently attracted the interest of many researchers in some applications on big data analysis. DBN is generative neural network model with many hidden layers, introduced by Hinton *et al.* [1] along with a greedy layer-wise learning algorithm. The building block of a DBN is a probabilistic model called Restricted Boltzmann Machine

(RBM). DBNs and restricted Boltzmann machines (RBMs) have already been applied successfully to solve many problems, such as classification, dimensionality reduction and image processing.

There have been several researches of applying DBNs to predict time series data in finance ([2]-[7]), meteorology ([8]-[11]), industry ([12]), internet traffic ([13]) and chaotic time series ([14], [15]). Most of these works pointed out that DBNs can bring out better prediction accuracy than traditional ANNs. However, one of the main shortcomings of using DBNs in time series prediction concerns with the proper selection of their parameters, i.e. the number of hidden units, learning rates, and so on. The task of model selection for deep neural networks aims at finding a suitable set of parameters that maximizes some fitness function, such as a classifier's accuracy. Although there exists some works related to using some population-based meta-heuristic techniques in parameter selection for DBNs, such as ([16]-[18]), most of them aim to serve the field of image classification.

In this work, we propose a method of parameter selection for DBNs in time series prediction which is based on Harmony Search algorithm ([19], [20]). This work is inspired by the work of Papa *et al.* in 2016 [18] which proposed a method of parameter selection for DBNs through Harmony Search. However, there are two major points in our work which makes it different from the previous work by Papa *et al.* in 2016.

i) We propose a method of parameter selection for DBNs in time series prediction rather than for DBNs in image classification.

ii) We compare the performance of Harmony Search method to that of Particle Swarm Optimization (PSO) in parameter selection for DBNs in time series prediction.

In experiment, we use three real time series datasets: Sunspots and some financial/economic datasets and one synthetic dataset: Lorenz. Experimental results through four datasets revealed that DBN with parameters selected by Harmony Search performs better than DBN with parameters selected by PSO or by random method in most of the tested datasets for time series prediction.

The remainder of the paper is organized as follows. Section II provides some basic backgrounds about DBN and Harmony Search algorithm. In Section III, the DBN model for forecasting time series and a method of parameter selection for DBN in time series prediction through Harmony Search are introduced. Section IV reports the experiments to evaluate the performance of Harmony Search in parameter selection for DBN model in time series prediction. Finally, Section V gives some conclusions and future work.

Manuscript received January 15, 2020; revised October 2, 2020. This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number C2019-20-17.

D. N. Luu was with the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam (e-mail: luudnbktp@gmail.com).

N. N. Phien is with Center for Applied Information Technology and Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam (e-mail: nguyennhocphien@tdtu.edu.vn).

D. T. Anh is with the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam (e-mail: dtanh@hcmut.edu.vn).

II. BACKGROUND

A. Deep Belief Network

Deep Belief Networks have been proposed by Hinton [1] with remarkable success in image processing and AI areas. DBN models are based on stacking of Restricted Boltzmann Machines (RBMs) ([21]).

RBM is a kind of stochastic artificial neural network with two connected layers: a layer of binary visible units (v , whose states are observed) and a layer of binary hidden units (h , whose states cannot be observed). The hidden units act as latent variables (features) that allow the RBM to model probability distribution over state vectors (see Fig. 1). The hidden units are conditionally independent given visible units. Given an energy function $E(v, h)$ on the whole set of visible and hidden units, the joint probability is given by:

$$p(v, h) = \frac{e^{-E(v, h)}}{Z} \quad (1)$$

where Z is a normalization partition function, which is obtained by summing up the energy of all possible (v, h) configurations.

$$Z = \sum_{v, h} e^{-E(v, h)} \quad (2)$$

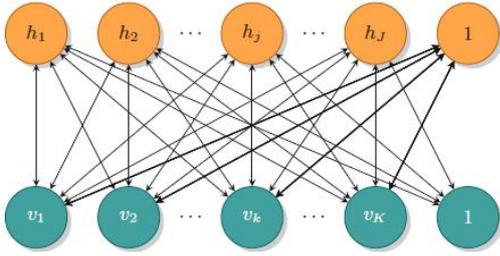


Fig. 1. Restricted Boltzmann Machine (RBM).

For the binary units $h_i \in \{0, 1\}$ and $v_i \in \{0, 1\}$, the energy function of the whole configuration is:

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}) &= -\mathbf{c}\mathbf{v}^T - \mathbf{b}\mathbf{h}^T - \mathbf{h}\mathbf{W}\mathbf{v}^T \\ &= -\sum_{k=1}^K c_k v_k - \sum_{j=1}^J b_j h_j - \sum_{j=1}^J \sum_{k=1}^K W_{jk} v_k h_j \end{aligned} \quad (3)$$

where \mathbf{W} is $J \times K$ matrix of RBM weights, $\mathbf{c} = [c_1, c_2, \dots, c_K]$ is the bias of the visible units and $\mathbf{b} = [b_1, b_2, \dots, b_J]$ is the bias of the hidden units. The marginal distribution over v is:

$$p(v) = \sum_h p(v, h) \quad (4)$$

The posterior probability of one layer given the other is easy to compute by the two following equations:

$$p(h_j = 1 | v) = \prod_j p(h_j = 1 | v) \quad (5)$$

where $p(h_j = 1 | v) = \sigma(b_j + \sum_k W_{jk} v_k)$

$$p(v_k = 1 | h) = \prod_k p(v_k = 1 | h) \quad (6)$$

where $p(v_k = 1 | h) = \sigma(c_k + \sum_j W_{jk} h_j)$

Notice that σ is the sigmoid function. Inference of hidden

factor h given the observed v can be done because h is conditionally independent given v .

A DBN is a generative model with an input layer and an output layer, separated by l layers of hidden stochastic units. This multilayer neural network can be efficiently trained by composing RBMs in such a way that the feature activations of one layer are used as the training data for the next layer.

An energy-based model of RBMs can be trained by performing gradient ascent on the log-likelihood of the training data with respect to the RBM parameters. This gradient is difficult to compute analytically. Markov Chain Monte Carlo methods are well-suited for RBMs. One iteration of the Markov Chain works well and corresponding to the following sampling procedure:

$$v_0 \xrightarrow{p(h_0|v_0)} h_0 \xrightarrow{p(v_1|h_0)} v_1 \xrightarrow{p(h_1|v_1)} h_1$$

where the sampling operations are schematically described. Rough estimation of the gradient using the above procedure is denoted by CD- k , where CD- k represents the Contrastive Divergence algorithm [1] for performing k iterations of the Markov Chain up to v_k .

The weight parameter is updated with the rate of change as shown in the following formula:

$$\nabla W_{jk} = \eta (\langle v_k h_j \rangle_0 - \langle v_k h_j \rangle_k) \quad (7)$$

where η represents the learning rate.

B. Harmony Search

Harmony Search is a meta-heuristic algorithm inspired by the improvisation process of music players. Musicians often improvise the pitches of their instrument searching for a perfect state of harmony ([19], [20]). The main idea is to use the same process adopted by musicians to create new harmonies to obtain a near-optimal solution according to some fitness function. Each possible solution is modeled as a harmony, and each musician corresponds to one decision variable.

Let $HM = (x^1, x^2, \dots, x^N)$ be a set of harmonies that compose the Harmony Memory of size N , such that x^i is an n -dimensional vector: $x^i = (x_1^i, x_2^i, \dots, x_n^i)$. The Harmony Search (HS) algorithm generates in each iteration a new harmony vector x' based on memory considerations, pitch adjustments and randomization (music improvisation). Then, the new harmony vector x' is evaluated in order to be accepted in the Harmony Memory: if x' is better than the worst harmony, the latter is replaced by the new harmony.

The Harmony Search needs two main parameters, which are *harmony memory consideration rate* (HMCR) and *pitch adjustment rate* (PAR). HMCR is the probability of choosing a value from the historic values stored in the Harmony Memory, and $(1 - \text{HMCR})$ is the probability of randomly choosing one feasible value. Notice that for each decision variable, there exists a given set of feasible values. Besides, every component of the new harmony vector x' is examined to determine whether it should be pitch-adjusted or not, which is controlled by the Pitch-Adjustment-Rate (PAR). The pitch adjustment is often used to improve solutions and to escape from local optima. This mechanism shifts the values of some decision variables in the harmony by the

formula:

$$x_j = x_j \pm \text{rand}() * bw \quad (8)$$

where bw is an arbitrary *distance bandwidth* and $\text{rand}()$ is a random number between 0 and 1.

The outline of Harmony Search is given as follows.

Algorithm 1 (Harmony Search)

1. Initialize the harmony memory HM of size N .
 2. Evaluate the fitness of all solutions in HM.
 3. **repeat**
 - a. //improvise a new harmony
 - for** $j = 1$ to n **do** // n : the number of components
 - if** $\text{rand} < HMCR$ //memory consideration
 - $x_j = x_j^a, a \in (1, 2, \dots, N)$
 - if** $\text{rand} < PAR$ //pitch adjustment
 - $x_j = x_j \pm \text{rand}() * bw$
 - endif**
 - else**
 - randomly select x_j in its domain.
 - endif**
 - endif**
 - endfor**
 - b. Evaluate the fitness of x
 - c. Update HM by replacing the worst HM member x_{worst} by x if $f(x)$ is better than $f(x_{worst})$, or disregard x otherwise.
 - d. Update the best harmony vector
 - e. set $t = t + 1$
- until** stopping criterion is satisfied.

To improve the convergence of Harmony Search and overcome some shortcomings of Harmony Search, a new variant of HS, called Global-best Harmony Search (GHS), was proposed by Omran & Mahdavi, 2008 ([22]).

The GHS dynamically updates parameter PAR as follows:

$$PAR(t) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{NI} t \quad (9)$$

where $PAR(t)$ is the pitch adjusting rate at iteration t , PAR_{min} and PAR_{max} are the minimum and the maximum adjusting rate, respectively, t is the iteration variable and NI is the number of iterations.

GHS modifies the pitch adjustment step of HS in order to take advantage of the guiding information of the *best harmony* in the Harmony Memory. Therefore, the distance bandwidth parameter bw is removed from the improvisation step, and the decision variable j of the new harmony is computed using a random component in the best harmony. GHS has the same steps as Harmony Search with the exception that the process of improvising a new harmony is modified as in Algorithm 2.

Due to the guiding information of the *best harmony* in HM, GHS algorithm outperforms HS. The outline of Global-Best Harmony Search is given as follows.

Algorithm 2 (Global-Best Harmony Search)

- for** $j = 1$ to n **do** // n : the number of components
- if** $\text{rand}(0,1) \leq HMCR$ **then**
- Choose a harmony x^a from HM randomly
- $x_j = x_j^a, a \in (1, 2, \dots, N)$
- if** $\text{rand}(0,1) \leq PAR$ **then**
- generate a random integer $k \in [1..n]$

```

// best is the index of the best harmony in HM
x_j = x_k^best
endif
else
x_j = L_j + rand(0,1).(U_j - L_j).
// U_j, L_j stand for the upper and lower bounds of decision
// variable j
endif
endfor

```

III. SELECTING PARAMETERS IN DBN BASED ON HARMONY SEARCH

In this work we use a deep belief network (DBN) with two RBMs as shown in Fig. 2 to be a model for time series prediction. This DBN model was also used by Kuremoto *et al.* in 2014 [23] for time series prediction. The training process for DBNs consists of two stages: a pre-training stage which is a layer-by-layer unsupervised learning using Contrastive Divergence CD-k algorithm and a fine-tuning stage which is a global supervised learning using back-propagation algorithm. These training algorithms make DBN extract abstract features by their multiple layers.

The structure of a deep neural network should be designed to satisfy its processing objective. The number of layers, the number of units of each layer, learning rate and so on need to be selected when the model is applied to a real dataset. Here we adopt Harmony Search as a meta-heuristic to find the optimum numbers of units in input layer and hidden layer and learning rate of RBM.

Suppose the prediction model is given in Fig. 2, in which 2 RBMs are used. The visible layer of RBM1 has n units, the hidden layer of RBM1 has m units and the visible layer of RBM2 also has m units. The hidden layer of RBM2 also has 1 unit, which is the output of the prediction model. Consider the learning rate of RBM η_1 and the learning rate during using back-propagation algorithm η_2 , a harmony vector is designed to be a 4-dimensional vector (n, m, η_1, η_2) where n and m are integers and η_1, η_2 are in the range $(0, 1)$.

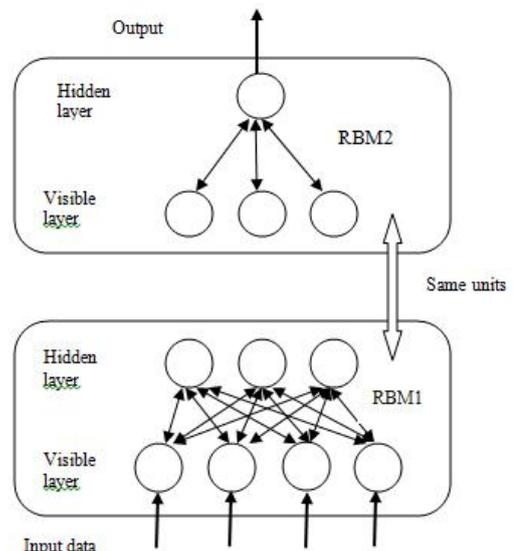


Fig. 2. A prediction model constructed as a deep belief network with two RBMs.

In this study, the mean squared error (MSE) is used as

evaluation criteria for a prediction model. The formula for MSE is given as follows:

$$MSE = \frac{1}{n} \sum_{t=1}^n (\hat{y}_t - y_t)^2 \quad (10)$$

where n is the number of observations, y_t is the actual value in time period t , and \hat{y}_t is the forecast value for time period t .

Here, we propose a method for parameter selection of DBN model in time series prediction which is based on Harmony Search. The summarized Harmony Search algorithm for this purpose on a dataset is shown as follows:

Algorithm 3: (DBN parameter selection through Harmony Search)

For each time series dataset

Step 1: Decide the size of Harmony Memory and limitation of iteration number I .

Step 2: Initialize the Harmony Memory randomly

Step 3: Evaluate each harmony (a set of parameters) using the MSE when applying this set of parameters to the DBN model for forecasting the time series dataset under consideration.

Step 4: Improve a new harmony (i.e. a new set of parameters) based on the current HM and apply it to the DBN model for forecasting the time series dataset under consideration. Update the Harmony Memory if the new harmony brings out the new MSE better than the worst harmony in the HM.

Step 5: Finish the algorithm if the current iteration index k reaches the maximum number of iterations I , else return to Step 4.

In this work, the Harmony Search used in Algorithm 3 is Global-Best Harmony Search.

IV. EXPERIMENTAL EVALUATION

The main purpose of this study is to evaluate the robustness of HS-based method in the context of selecting DBN parameters considering the task of time series prediction. We will compare the proposed HS-based DBN model selection against with random initialization of parameters and Particle Swarm Optimization-based DBN model selection in terms of prediction accuracy. We select Particle Swarm Optimization (PSO) [24] as another competitive method for selecting parameters of DBNs since this meta-heuristic has been applied for the same purpose in the work by Kuremoto *et al.*, 2014 [14]. The HS algorithm we implement in this experiment is Global-Best-Harmony Search.

We implemented the DBN forecasting model with Tensorflow framework (using Python language) [25]. Tensorflow is an open-source software library developed by Google which is very useful in building, training, testing and visualizing deep neural networks. And we conducted the experiments on a Core i7-5500U 2.4 GHz, RAM 8GB PC.

A. Datasets

Here, the tested datasets consist of 3 real world time series datasets and one synthetic time series datasets. All these

datasets are commonly-used by the research community in time series prediction and they are considered as the challenging datasets for prediction. They are described as follows.

1. Monthly sunspot numbers from January of 1949 to March of 1977. (This dataset from the web site: <http://sidc.oma.be>). This time series is in the field of astronomy and is a widely-used benchmark dataset in evaluation of several proposed methods for time series prediction. It consists of 305 data points.

2. Monthly CPI (Consumer Price Index) in Spain from January of 1960 to June of 2005. (This dataset from: <http://www.bde.es/bde/en/>). This time series consists of 535 data points.

3. Monthly exchange rates US dollar/British Pound (USD/GBP) from January 1981 to July of 2005. (This dataset from the web site: <http://www.bde.es/bde/en/>). This time series consists of 295 data points.

4. This chaotic time series dataset is derived from the Lorenz system, given by the three differential equations:

$$\begin{cases} \frac{dx}{dt} = a(y - x) & \frac{dy}{dt} = x(b - z) - y \\ \frac{dz}{dt} = xy - cz \end{cases} \quad (11)$$

where, $a = 10$, $b = 28$, and $c = 8/3$. This time series consists of 1000 data points.

Fig. 3 shows the plots of the three real world datasets. Fig. 4 shows the plot of the synthetic dataset (Lorenz).

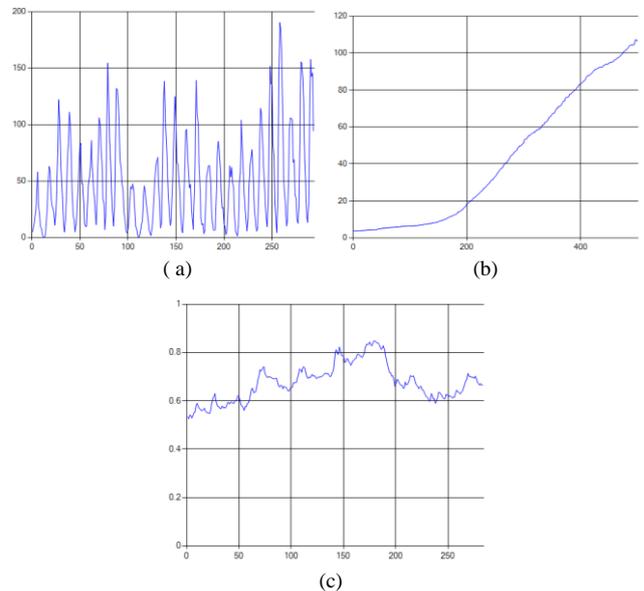


Fig. 3. Three datasets (a) Sunspots, (b) CPI and (c) USD/GBP.

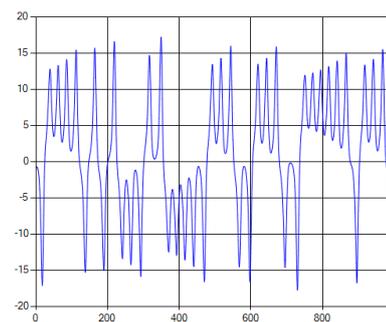


Fig. 4. Lorenz dataset.

Each dataset is divided into two sets: training set and test set. The training set and test set for each of the seven datasets are given in Table I.

TABLE I: TRAINING SET AND TEST SET FOR EACH OF SEVEN DATASETS

Dataset	Length	Training set	Test set
Sunspots	305	288	17
CPI	535	481	54
USD/GBP	295	285	10
Lorenz	1000	800	200

We have set some DBN parameters as follows: number of epochs for RBM to 50, number of iterations for CD algorithm to 2, and batch size in BP algorithm to 32.

TABLE II: PARAMETERS USED IN THE PREDICTION EXPERIMENTS (FOR SUNSPOTS DATASET)

Description	Range	Quantity
Number of iterations for HS to select DBN parameters		1000
Number of units in visible layer of RBM1	$n (1 \leq n \leq 10)$	Given by HS
Number of units in hidden layer of RBM1	$m (1 \leq m \leq 10)$	Given by HS
Learning rate of RBM	$\eta_1 (0.005 \text{ to } 0.2)$	Given by HS
Learning rate of BP	$\eta_2 (0.0008 \text{ to } 0.08)$	Given by HS
Number of epochs for BP		2500

TABLE III: PARAMETERS USED IN THE PREDICTION EXPERIMENTS (FOR CPI DATASET)

Description	Range	Quantity
Number of iterations for HS to select DBN parameters		1000
Number of units in visible layer of RBM1	$n (1 \leq n \leq 10)$	Given by HS
Number of units in hidden layer of RBM1	$m (1 \leq m \leq 10)$	Given by HS
Learning rate of RBM	$\eta_1 (0.001 \text{ to } 0.1)$	Given by HS
Learning rate of BP	$\eta_2 (0.0005 \text{ to } 0.05)$	Given by HS
Number of epochs for BP		1.200

TABLE IV: PARAMETERS USED IN THE PREDICTION EXPERIMENTS (FOR USD/GBP DATASET)

Description	Range	Quantity
Number of iterations for HS to select DBN parameters		2000
Number of units in visible layer of RBM1	$n (1 \leq n \leq 10)$	Given by HS
Number of units in hidden layer of RBM1	$m (1 \leq m \leq 10)$	Given by HS
Learning rate of RBM	$\eta_1 (0.0005 \text{ to } 0.05)$	Given by HS
Learning rate of BP	$\eta_2 (0.00005 \text{ to } 0.005)$	Given by HS
Number of epochs for BP		4000

TABLE V: PARAMETERS USED IN THE PREDICTION EXPERIMENTS (FOR LORENZ)

Description	Range	Quantity
Number of iterations for HS to select DBN parameters		2000
Number of units in visible layer of RBM1	$n (1 \leq n \leq 10)$	Given by HS
Number of units in hidden layer of RBM1	$m (1 \leq m \leq 10)$	Given by HS
Learning rate of RBM	$\eta_1 (0.001 \text{ to } 0.1)$	Given by HS
Learning rate of BP	$\eta_2 (0.003 \text{ to } 0.3)$	Given by HS
Number of epochs for BP		4000

The values of parameters used in the prediction

experiments for Sunspots, CPI, USD/GBP and Lorenz datasets are shown in Table II, Table III, Table IV and Table V respectively.

B. Experimental Results

Through about 3 experiments on each dataset with each method of parameter selection, the best values of the four main parameters n , m , η_1 and η_2 in DBN model for all datasets with three methods of selecting parameters are reported in Table VI.

TABLE VI: BEST PARAMETERS FOR DBN SELECTED BY THE THREE METHODS OVER 4 DATASETS

Dataset	Method of selecting DBN parameters	Best parameters
Sunspots	Random	$n = 5, m = 9, \eta_1 = 0.190985, \eta_2 = 0.053755$
	HS	$n = 9, m = 10, \eta_1 = 0.155583, \eta_2 = 0.066382$
	PSO	$n = 8, m = 6, \eta_1 \eta_2 = 0.063236, \eta_2 = 0.708165$
CPI	Random	$n = 1, m = 8, \eta_1 = 0.054726, \eta_2 = 0.027233$
	HS	$n = 1, m = 3, \eta_1 = 0.098313, \eta_2 = 0.001119$
	PSO	$n = 1, m = 3, \eta_1 = 0.078899, \eta_2 = 0.033510$
USD/GBD	Random	$n = 3, m = 6, \eta_1 = 0.031440, \eta_2 = 0.002913$
	HS	$n = 3, m = 10, \eta_1 = 0.03212, \eta_2 = 0.0049123$
	PSO	$n = 1, m = 1, \eta_1 = 0.044547, \eta_2 = 0.000444$
Lorenz	Random	$n = 9, m = 7, \eta_1 = 0.010804, \eta_2 = 0.016041$
	HS	$n = 5, m = 10, \eta_1 = 0.001539, \eta_2 = 0.028216$
	PSO	$n = 9, m = 7, \eta_1 = 0.010804, \eta_2 = 0.134582$

The experimental results on prediction accuracy based on MSE prediction error over the four datasets for each optimization technique are reported in Table VII.

TABLE VII: PREDICTION ERRORS OF DBN WITH PARAMETERS SELECTED BY THE THREE METHODS OVER 4 DATASETS

Dataset	Random	PSO	Harmony Search
Sunspots	0.007316	0.010027	0.007009
CPI	0.000015	0.000014	0.000014
USD/GBP	0.000785	0.000827	0.000779
Lorenz	0.000006	0.000004	0.000007

From the experimental results in Table VII, we can see that the MSE prediction errors of DBN model with the main parameters selected by Harmony Search are always lower than those of DBN model with the main parameters selected by the random method or by PSO in most of the tested datasets (3 out of 4). As for Lorenz dataset, with the three methods of parameter selection, all the MSE values here are higher than the MSE given by DBN model in the work by Nam, 2017 [15] (MSE = 0.00002) even though this work selected parameters for DBN model by a random search. This is due to the fact that in this work we did not combine

DBN model with chaos theory to deal with Lorenz dataset which is a well-known chaotic time series.

All the experimental results in Table VII imply that Harmony Search can be used as a meta-heuristic method in parameter selection for DBNs in time series prediction.

We also measured the execution times of tuning parameters for DBN model by the two methods: Harmony Search and PSO. The execution times of PSO-based method is higher than Harmony Search-based method. Harmony Search is substantially faster than PSO technique since it does not update all possible solutions at each iteration, but only one.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose and evaluate the use of Harmony Search, a meta-heuristic, in parameter selection of Deep Belief Networks for time series prediction. The parameters here consist of number of units in visible layer of RBM1, number of units in hidden layer of RBM1, learning rate in training RBM and learning rate in Back Propagation algorithm. Experimental results obtained reveal that the DBN method with the main parameters selected by Harmony Search performs better than DBN model with the main parameters selected by the random method or by PSO method in most of the tested datasets. These results imply that Harmony Search can be used as a robust meta-heuristic method in parameter selection for DBNs in time series prediction.

As for future work, we intend to apply Harmony Search in parameter selection of the DBN model with a better architecture proposed by Kuremoto *et al.* in [14] for forecasting chaotic time series. In this future research direction, we will apply chaos theory ([26]-[30]) to determine the number of units in visible layer of RBM1 rather than including it in the set of parameters selected by Harmony Search.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

D. T. Anh proposed the main ideas of the research. D. N. Luu and N. N. Phien implemented the comparative approaches and empirically evaluated the benefits of the proposed approach. D. T. Anh wrote the paper, and all authors had approved the final version.

REFERENCES

- [1] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computing*, vol. 18, pp. 1527-1554, 2006.
- [2] J. Chao, F. Shen, and J. Zhao, "Forecasting exchange rates with deep belief networks," in *Proc. International Conference on Neural Networks*, San Jose, California, USA, 31 July–August 5, 2011, pp. 1259-1266.
- [3] E. Chong, C. Chang, and F. C. Park, "Deep learning network for stock market analysis and prediction: Methodologies, data representation and case studies," *Expert Systems with Applications*, vol. 83, pp. 187-205, Oct. 2017.
- [4] J. H. N. Pena and G. J. Hernandez-Perez, "Price direction prediction on high frequency data using deep belief networks," in *Proc. WEA 2016*, 2016, pp. 74-83.
- [5] F. Shen, J. Chao, and J. Zhao, "Forecasting exchange rate using deep belief networks and conjugate gradient method," *Neurocomputing*, 2015.
- [6] J. Zheng, X. Fu, and G. Zhang, "Research on exchange rate forecasting based on deep belief network," *Neural Computing and Applications*, Springer, 2017.
- [7] C. Zhu, J. Yin, and Q. Li, "A stock decision support systems based on DBNs," *Journal of Computational Information Systems*, vol. 10, no. 2, pp. 883-893, 2014.
- [8] Y. Cheng, X. Zhou, S. Wan, and K. K. R. Choo, "Deep belief network for meteorological time series prediction in the Internet of Things," *IEEE Internet of Things Journal*, vol. 6., no. 3, pp. 4369-4376, 2018.
- [9] C. Li, Y. Bai, and B. Zeng, "Deep learning architecture for daily reservoir inflow forecasting," *Water Resource Management*, vol. 30, pp. 5145-5161, 2016.
- [10] M. Qin, Z. Li, and Z. Du, "Red tide time series forecasting by combining ARIMA and deep belief network," *Knowledge-Based Systems*, vol. 125, pp. 39-52, 2017.
- [11] N. C. Tri, H. N. Duong, H. V. Tran, and V. Snasel, "Predicting daily river runoff by deep belief networks," in *Proc. Int. Conf. on Information and Convergence Technology for Smart Society*, Ho Chi Minh City, Jan. 19-21, 2016.
- [12] W. He, "Deep neural network based load forecast," *Computing Modelling and New Technologies*, vol. 18, no. 3, pp. 258-262, 2014.
- [13] S. Narejo and E. Pasero, "An application of internet traffic prediction with deep neural network," in *Multidisciplinary Approaches to Neural Computing*, A. Esposito *et al.* Eds. Springer, 2018, pp. 139-149, Chapter 14.
- [14] T. Kuremoto, M. Obayashi, K. Kobayashi, T. Hirata, and S. Mabu, "Forecast chaotic time series data by DBNs," in *Proc. 7th Int. Congress on Image & Signal Processing*, 2014.
- [15] T. N. H. Nam, "Chaotic time series prediction based on deep neural networks," Master Thesis, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, 2018.
- [16] K. Liu, L. M. Zhang, and Y. W. Sun, "Deep Boltzmann machines aided design based on genetic algorithms," *Applied Mechanics and Materials*, 2014.
- [17] G. Rosa, J. P. Papa, K. Costa, L. Passos, C. Pereira, and X. Yang, "Learning parameters in deep belief networks using firefly algorithm," *Artificial Neural Networks in Pattern Recognition*, pp. 138-149, 2016.
- [18] J. P. Papa, W. Scheirer, and D. D. Cox, "Fine-tuning deep belief networks using harmony search," *Applied Soft Computing*, vol. 46, September 2016.
- [19] Z. W. Geem and J. H. Kim, "A new heuristic optimization algorithm: harmony search," *SIMULATION*, vol. 76, no. 2, pp. 60-68, 2001.
- [20] Z. W. Geem, "Global optimization using harmony search: theoretical foundations and applications," *Foundations of Computational Intelligence*, vol. 3, pp. 57-73, 2009.
- [21] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1-127, 2009.
- [22] M. G. H. Omran and M. Mahdavi, "Global best harmony search," *Applied Mathematics and Computation*, vol. 198, no. 2, pp. 643-656, 2008.
- [23] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, "Time series forecasting using a deep belief network with restricted Boltzmann machines," *Neurocomputing*, 2014.
- [24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, 1995, pp. 1942-1948.
- [25] S. Pattanayak, *Pro Deep Learning with Tensorflow*, Apress, 2017.
- [26] Z. Liu, "Chaotic time series analysis," *Mathematical Problems in Engineering*, Article ID 720190, pp. 1-31, 2010.
- [27] M. T. Rossenstein, J. J. Colins, and C. J. de Luca, "Reconstruction expansion as a geometry-based framework for choosing proper delay times," *Physica D*, vol. 73, no. 1, pp. 82-98, 1994.
- [28] M. B. Kennel, R. Brown, and H. D. I. Abarbanel, "Determining embedding dimension for phase-space reconstruction using a geometrical construction," *Physical Review A*, vol. 45, no. 6, pp. 3403-3411, 1992.
- [29] A. M. Fraser and H. L. Swinney, "Independent coordinates for strange attractors from mutual information," *Physica D*, vol. 33, p. 1134, 1986.
- [30] F. Takens, "Detecting strange attractors in turbulence," *Dynamical Systems and Turbulence, Warwick 1980*, London, Springer, pp. 366-381, 1981.

Copyright © 2021 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted

use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](#)).



Do Ngoc Luu was born in Quang Ngai, Vietnam in 1990. He received B.Eng. and M. Eng. in computer science from the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology in 2013 and 2019, respectively. Now he works as a software developer for a software company in Ho Chi Minh City, Viet Nam. Mr. Luu's research fields includes machine learning and time series data mining.



Nguyen Ngoc Phien was born in Quang Ngai, Vietnam in 1992. He received B.Eng. in computer science from FPT University in 2014 and M. Eng. in computer science from the Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology in 2017. Now he works as a lecturer for Ton Duc Thang University, Ho Chi Minh City, Viet Nam. Mr. Phien's research fields includes machine learning and time series data mining.



Duong Tuan Anh was born in Quang Ngai, Vietnam in 1953. He received the B.Eng. in electronic engineering from Ho Chi Minh City University of Technology University in 1976. He received his master of engineering and doctorate of engineering in computer science from the School of Advanced Technologies at Asian Institute of Technology, Bangkok, Thailand in 1989 and 1998, respectively. He is currently associate professor of computer science at Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology. His research is in fields of metaheuristics, machine learning and time series data mining. Dr. Anh is currently the head of Time Series Data Mining Research Group in his faculty. He authored more than 100 scientific papers.