

<sup>1</sup>Man Jing

# Application of Time Series Analysis and Regression Algorithm in Assessing the Effectiveness of Monetary Policy in the Era of Digital Economy



**Abstract:** - Monetary policy is the series of actions to manage country's money supply and to attain economic growth. An objection to using interest rates in monetary policy is that the use of interest rates has the problem of time uncertainty. Therefore, this research introduces Conditional Autoregressive Value-at-Risk and Henry Gas Solubility Optimization (Caviar-HGSO) with Deep Long Short Term Memory (DeepLSTM) to monetary policy forecasting to demonstrate its effectiveness. Here, Caviar-HGSO is a hybrid of Caviar and HGSO used to tune DeepLSTM weights. The HGSO algorithm follows the wind state to balance exploration and exploitation in search space while avoiding local optimization. A Caviar model estimates the parameter over time by autoregressive process and determine a parameter by regression analysis. The advantages of these two optimization algorithms lead to better monetary policy forecasts. The experimental results show that Caviar-HGSO\_DeepLSTM performs better regarding MAE, MSE, and RMSE, namely 0.438, 0.103, and 0.351.

**Keywords:** Monetary Policy, Conditional Autoregressive Value-at-Risk, Deep Belief Network, Henry Gas Solubility Optimization, Deep Long Short Term Memory.

## 1. Introduction

Monetary and fiscal policy are the most critical aspects of the government intervention in macroeconomic direction, particularly monetary policy, which is one of an administrative function of central banks worldwide [2]. There are still many reasons for the economic crises and the financial risk events. Their signs are dissimilar in different periods of history. In international process of the financial markets, monetary policies of other countries are significantly affected by the problem of the financial market [11] [1]. Monetary policy is designed to influence the economy's direction of the in a more favorable direction. For example, when economic productivity falls, or unemployment rises, authorities can lower interest rates or increase the money supply to encourage consumption and investment [7].

Although scholars have studied transmission mechanism of the monetary policy [9][3], interest in the monetary policy continues to grow today [10][3], especially in considering its impact on changes in the economy [3]. Monetary policy trend impacts the behavior of the market participants, like general public and the financial institutions, allowing the economy to grow steadily. Predicting the path of monetary policy is theoretically and practically important [6]. The majority of specialists and intellectuals agree with this view point. Monetary policy and financial markets have a strong association, particularly macro-control and economic bubble during crisis period [12]. Therefore, monetary policy prediction research is quite essential. It can detect and prevent financial hazards early [1].

Scholars in monetary policy and the financial markets focus on economic policy effectiveness, transmission mechanisms, and regulatory expectations [15]. Currently, experts use discrete choice models and compound forecast schemes to determine operating direction of monetary policy. With the introduction of new algorithms and ideas, more research is being done to understand complicated financial systems from the standpoint of economic physics and machine learning [16]. Methods and models such as the Dynamic Stochastic General Equilibrium Model (DSGE), Generalized Method of Moments (GMM), machine learning algorithms, and others

<sup>1</sup> School of Economics and Trade, Guangzhou Xinhua University, Guangzhou, Guangdong, 510000, China

\*Corresponding author e-mail: jingman2737@xhsysu.edu.cn

Password 19860904Jingman

<https://exmail.qq.com/login>

Copyright © JES 2024 on-line : [journal.esrgroups.org](http://journal.esrgroups.org)

are useful in regulating monetary policy. Still, their forecast accuracy needs to be higher and solve overfitting issues [17].

The primary aim of this paper is to introduce the method for monetary policy forecasting called Caviar-HGSO. Here, Caviar-HGSO is used to adjust DeepLSTM weights where Caviar and HGSO are combined to form Caviar-HGSO. Next, the features were combined using DBN and Bray-Curtis distance to combine the best features for further processing. In addition, oversampling is also used to refine the data. Based on augmented data, the Caviar-HGSO\_DeepLSTM predicts monetary policy.

The primary contributions of this work are:

Caviar-HGSO is a combination of Caviar and HGSO for monetary policy forecasts. The advantage of these two optimization algorithms is that they produce better prediction results. The final set of active hypothesis weights in CAVIAR depends on the weak learners and the training data. A new physics optimization algorithm called HGSO is presented, following Henry's law behavior.

This research article is structured as follows: Section 2 reviews of literature of some monetary policy predictions; Section 3 provides a brief explanation of Caviar-HGSO; Section 4 presents a results and details of the project; Section 5 summarizes this paper.

## 2. Literature survey

Minrong Lu [1] introduced Back Propagation Neural Network (BPNN) and established a predictive model for monetary policy. Then, considering the importance of the characteristics of the bank index data, expert weights were applied to the BP, and a weighted regression model (WBP) was developed. According to the sequential nature of financial markets, the WBP model was improved, and a weighted regression model (TWBP) was established. The accuracy of the TWBP model was higher, but the model should continue to improve its performance using optimization algorithms.

Maryam Hajipour Sardue *et al.* [2] developed Deep P-V-L: Deep network Variational Auto-Encoder LSTM (VAE-LSTM) forecasting, which significantly improves the forecasting performance of macroeconomic indicators through two deep architectures of VAE and LSTM, unsupervised (representative) learning and observational learning (prediction and real-time prediction). Representational learning using VAE achieved accuracy for time-series prediction models for LSTM architecture. However, new techniques and Artificial Intelligence (AI) were not considered for reconstructing data and traditional data revision.

Betchani H. M. Tchereni *et al.* [3] developed a vector error correction method to study short and the long-run dynamic relationships between variables. Furthermore, fluctuations in stock prices impact the Gross Domestic Product (GDP). The strategy operated the target monetary policy variable more effectively than reserve money. However, the method could have been more responsive to changes in economic policy factors, such as interest rates and money supply.

Georgios Georgiadisy and Arnaud Mehl [4] believe that the importance of two specific aspects of the financial world determines their impact on monetary policy effectiveness: the size of the external fund of the economy, which determines the effects of international economic changes on the domestic market. In developing countries, the impact of foreign exchange is higher than the impact of global financial cycle; due to the global financial, effectiveness of monetary policy increases. However, this approach needs to examine the potential difference in the importance of the impact of foreign exchange on the public, household, business, and financial sectors.

Cheng Yang and ShuhuaGuo [5] introduced Recurrent Neural Networks and Gated Recurrent Units (GRU-RNN) to forecast inflation. In addition, the Consumer Price Index (CPI) was used as an inflation indicator, and various indicators related to the economy were used as CPI eigenvectors. The trained GRU-RNN was designed to optimize the network parameters using historical data as input. The CPI vectors were used as input to obtain CPI forecast values. GRU-RNN does a good job predicting China's inflation rate, but it needs to design feature vectors that describe inflation better or CPI to train the network.

Chuanxin Qiu *et al.* [6] used a Random Forest (RF) to quantify and anticipate Bank of China's monetary policy given 16 macroeconomic factors. The method was tested against a Support Vector Machine (SVM), CART decision tree, and a Neural Network (NN), and it gained higher forecast accuracy when determine a direction of central bank's monetary policy. However, it does not consider the Targeted Medium-Term Lending Facility (TMLF) to determine the better accuracy.

Natascha Hinterlang [7] applied Artificial Neural Networks (ANN) to predict the federal funds rate better than linear and nonlinear Taylor's laws with different methods. Specifically, the “within quartile” parameter produces the most minor error of the squared error of approximation (one over the top four quartiles). The method is robust to different time periods, but it can only be determined locally and there is no economic interpretation of the parameters.

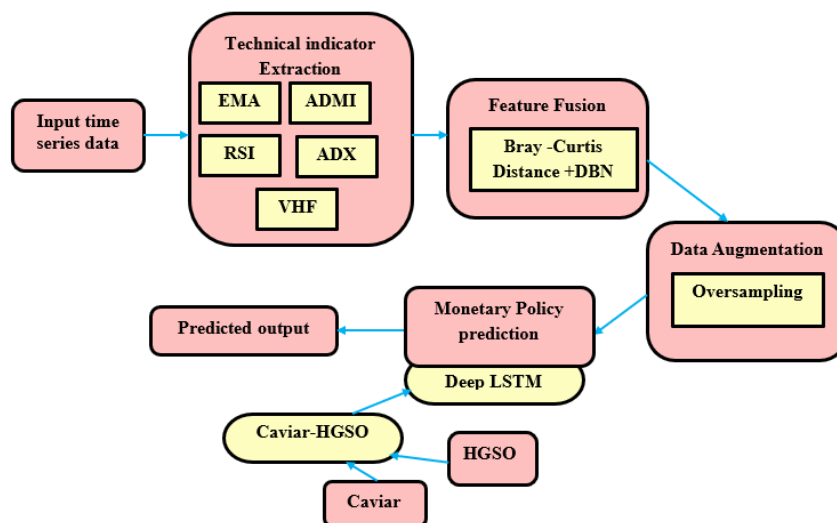
Szabolcs Deak *et al.* [8] assign models by predictive performance than the best-practice model in line with their future goals. These scales were used to resolve the policy design issue of choosing an optimal and robust Taylor-type interest rate policy to model the uncertainty in DSGE (with no uncertainty, financial or not). In addition, the cost level rule has good stability and robustness. However, the selection of weights significantly affects the active optimization rule.

The following issues are detected in the relevant work:

Low-income nations confront additional hurdles in determining monetary policy transmission channels due to their undeveloped institutional structures. Monetary policy primarily concerns with achieving high and the sustained economic growth while keeping inflation low. A policy target for inflation is consistent with a desired pace of the output and the employment growth. Tightening monetary policy above market expectations increases the likelihood of experiencing severe systemic stress levels. Resolving the problem of national sovereignty over monetary policy, and the need for global engagement in crisis prevention remain vital challenges.

### 3. Proposed methodology

This section describes the Caviar-HGSO\_DeepLSTM model for monetary policy prediction. The steps for prediction include technical indicator extraction, feature fusion, data augmentation, and monetary policy prediction. Initially, technical indicators such as Average Directional Index (ADX), Exponential Moving Average (EMS), Relative Strength Index (RSI), Average Directional Movement Index (ADMI), and VHF [18] are retrieved from time series data. After the technical indications are recovered, the features are fused using Bray Curtis distance and DBN [20], and the data is enhanced through oversampling. Finally, Caviar-HGSO predicts monetary policy by DeepLSTM [12]. The Caviar-HGSO combines Caviar [21] and HGSO [14]. Figure 1 shows the block of Caviar-HGSO\_DeepLSTM for monetary policy prediction.



**Figure 1.** Schematic view of Caviar-HGSO\_DeepLSTM for Monetary policy prediction

### 3.1 Acquisition of the input data

Time series are essential to the material derived from scientific and financial applications. Time series data includes more extensive data; as the amount of data increases, numerical and continuous values can represent the data, so predictions must be made in monetary policy analysis. Time series reports use verified data as a basis for evaluating future actions. Let's assume that the data set contains a lot of time series information, and the mathematical formula is:

$$L = \{I_1, I_2, \dots, I_l, \dots, I_k\}; 1 \leq l \leq k \tag{1}$$

where,  $L$  is dataset,  $k$  is total data, and  $I_l$  is time-series data located at the index  $l$  in the dataset.

### 3.2 Extracting Technical Indicators

Input data  $I_l$  is given to a technical indicator extractor module to capture the technical elements in the input data. However, technical feature extraction improves the accuracy of monetary policy forecasting. Technical indicators derived from time series data include ADMI, RSI, EMA, ADX, VHF, and SMI [18], as explained below:

*RSI*: It validates the quantity of new profits received against the value of new losses, thus explaining the price weakness or strength.

$$q_1 = 100 - \frac{100}{(1 + q_1(\delta^+) / q_1(\delta^-))} \tag{2}$$

where,  $q_1$  is RSI output.

*EMA*: This falls into the Weighted Moving Average (WMA) category, which allocates a weight value to each data value in time series information. Here, the cost of the previous weighted values is reduced, which is determined as:

$$q_2 = \sum_{g=0}^{d-1} X_g M_{t-g} \tag{3}$$

where,  $d$  is the length of input window,  $M_t$  is the day  $r$  close price,  $X_g$  is the weight, and  $q_2$  is the output of EMA.

*ADMI*: It refers to the strength or weakness of the current value of the monetary policy. The calculation depends on a combination of negative and positive movement indicators, measured over recent days and the input window length. Furthermore, this is also called,

$$q_3 = 100 * (B_d^+ - B_d^-) / (B_d^+ - B_d^-) \tag{4}$$

$$B_d^+ = 100 * q_2(\delta^+) / C_d \tag{5}$$

$$B_d^- = 100 * q_2(\delta^-) / C_d \tag{6}$$

$$\delta^+ = \max(M_t - M_{t-1}, 0) \tag{7}$$

$$\delta^- = \min(M_t - M_{t-1}, 0) \tag{8}$$

$$C_d = q_2(\max(HM_t - LM_t, |HM_t - M_{t-1}|, |LM_t - M_{t-1}|)) \tag{9}$$

where,  $\delta^+$  and  $\delta^-$  are positive and negative directional movement,  $HM_t$  and  $LM_t$  is high and low price on a day  $t$ ,  $M_t$  is closing price per day  $t$ ,  $|\cdot|$  shows absolute value. The ADMI output is  $q_3$ .

ADX: This feature is measured in positive and negative shift values, so it is expressed as:

$$q_4 = \frac{\text{sum}(B^+ - B^-)}{(B^+ + B^-) / d} \tag{10}$$

where,  $q_4$  is ADX output.

VHF: It defines the policy values that indicate the congestion or trending phase that is displayed as  $q_5$ . Therefore, technical indicators derived from these data are designated as  $q = \{q_1, q_2, q_3, q_4\}$ , respectively.

### 3.3. Feature fusion

The DBN [20] with a Bray-Curtis distance is used for feature fusion considering account the feature vector  $q$  as input. It is also employed to calculate a distance between binary and categorical data.

- Bray-Curtis distance for sort the features

Features obtained from time series data are ranked based on Bray-Curtis distance based on number of features selected and expressed as,

$$Z = \sum_{l=1}^n \frac{\alpha}{a} Z_l \tag{11}$$

$$l = n - \frac{J}{b} \tag{12}$$

$$b = \frac{J}{n}; 1 \leq a \leq n \tag{13}$$

where,  $J$  is feature numbers,  $n$  is features selected,  $\alpha$  is ideal parameter.

-Construct  $\alpha$  using DBN

In DBN [20], the size and features of the training material are used. All properties in the DBN are passed to get the specified value. Next, the ideal factor value  $\alpha$  is found by DBN. During training, the ideal parameter values are expressed as:

$$\alpha = L(c_l, \beta_l) \tag{14}$$

Among them,  $\beta_l$  represents the average  $c_l$  of the same class,  $c_l$  is a data record, and  $L$  is Bray Curtis distance.

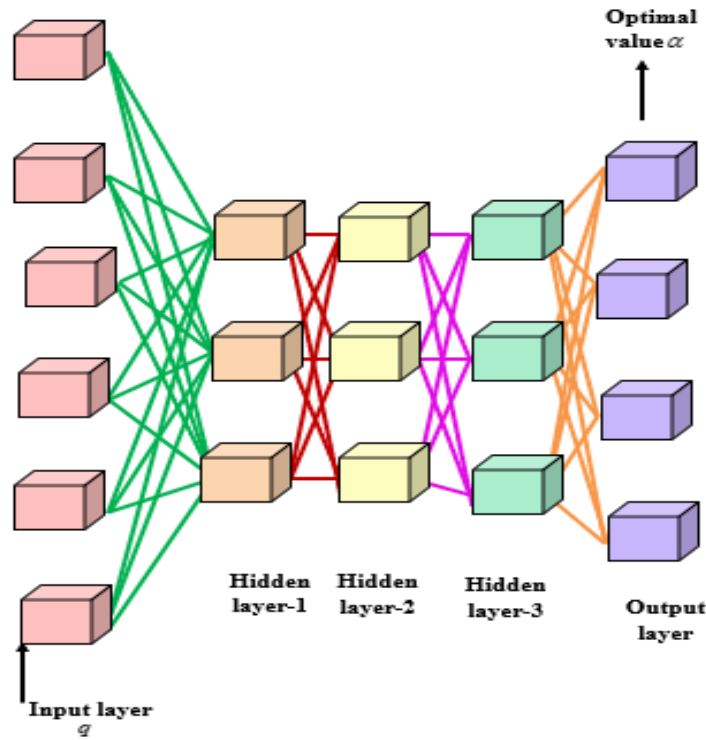
The distance between image  $\beta_l$  and target  $c_l$  is expressed by:

$$L(l, g) = \frac{\sum_{d=0}^{k-1} |Z_{l,d} - Z_{g,d}|}{\sum_{d=0}^{k-1} |Z_{l,d} + Z_{g,d}|} \tag{15}$$

where,  $d$  is variable index,  $k$  is variable,  $g$  is data,  $g$  is class. The selected features of Bray-Curtis distance are called as  $\sigma$ .

(i) *DBN model*

DBN classifier is used to determine ideal parameter values. DBN consists of multilayer perceptron (MLP) and finite element Boltzmann machine (RBM) layers. MLP layers consist of hidden, input, and output layers, while RBM layers consist of visible and hidden layers. Visible layers are connected to hidden layers. Figure 2 depicts the DBN model.



**Figure 2.** Structure of DBN

The input to visible and a hidden layers of RBM 1 is modeled by,

$$H^1 = \{H_1^1, H_2^1, \dots, H_X^1, \dots, H_n^1\}; 1 \leq X \leq n \tag{16}$$

$$G^1 = \{G_1^1, G_2^1, \dots, G_Z^1, \dots, G_k^1\}; 1 \leq Z \leq k \tag{17}$$

where,  $H_X^1$ ,  $G_Z^1$ , and  $k$  is RBM-1 visible neuron  $X$ , hidden neuron  $Z$ , and total hidden neuron. Let  $S$  and  $M$  are visible and hidden layer biases and RBM-1 layer biases is given by,

$$S^1 = \{S_1^1, S_2^1, \dots, S_X^1, \dots, S_k^1\} \tag{18}$$

where,  $S_X^1$  is visible neuron bias. The RBM-1 weight vector is,

$$F^1 = \{F_{X,Z}^1\}; 1 \leq X \leq n; 1 \leq Z \leq k \tag{19}$$

where,  $F_{X,Z}^1$  is weight of visible neuron  $X$  and hidden neuron  $Z$ . Consider the RBM-1 output hidden layer evaluated by biases and weights associated with each visible neuron, given as follows:

$$J_Z^1 = \delta \left[ G_Z^1 + \sum_P (H_X^Z) F_{X,Z}^1 \right] \tag{20}$$

where,  $\delta$  is activation function. Therefore, the output by RBM1 is given as,

$$N^1 = \{N_Y^1\}; 1 \leq Y \leq n \tag{21}$$

Next, the output of RBM1 is fed into the detection layer of RBM2. Therefore, the input layer of RBM2  $H^2$ . Next the hidden layer RBM 2 is given  $G^2$ , RBM2 weight vector is called  $F^2$ , hidden neuron output  $Z$  is termed by  $G_Z^2$  and the bias linked with hidden neuron  $Z$  is  $G_u^2$ . Therefore, the hidden layer result is shown as  $G^2$ .

Therefore, considering the weights  $F^\varphi$  to evaluate the output vector, the result of hidden layer is expressed by,

$$H_s = \sum_{Y=1}^Z F_{Ye}^\varphi * L_Z \tag{22}$$

where,  $F_{Ye}^\varphi$  is weight among  $e^{th}$  output neuron and  $Y^{th}$  hidden neuron, and output of hidden layer is expressed as  $H_s$ . The outcome of DBN is displayed by  $\alpha$ .

### 3.4 Data augmentation

The output of feature fusion  $\alpha$  is given to data augmentation to generate more training samples using oversampling. The process of generating a sample size by considering class labels of sample. Using category labels, a data is divided into more samples. The advantage of data augmentation is that it maximizes the amount of data, which also helps to higher the prediction accuracy of monetary policy. Since, deep learning classifiers are more efficient when processing more training samples, a data augmentation method is used to generate multiple samples using an oversampling technique. However, the results of the augmented data are considered as  $T$ .

### 3.5 Monetary Policy Prediction by Caviar-HGSO\_DeepLSTM

Here, augmented data  $T$  is considered to determine the monetary policy using DeepLSTM trained with Caviar-HGSO. Caviar-HGSO is a hybrid of Caviar [21] and HGSO [14] and is used for monetary policy forecasting. The benefits of Deep LSTM networks is improved processing speed. A Deep LSTM model structure is explained in the next section.

#### Structure of Deep LSTM

LSTM is the enhanced RNN architecture to learn long-term dependencies. Deep LSTM [12] consists of LSTM neurons that contain input gates, forget gates, units, output gates, and output responses. Input and output gates are used to pass messages between units. The memory unit has a self-linked recurrent edge of weight, which ensures that the gradient must go through many times without exploding or disappearing. Therefore, Deep LSTM models solve the complexity due to gradient loss during RNN processing. For each neuron, the recursive calculation of activation units is given as:

$$x_o = \kappa(Q_{vx}v_m + Q_{ix}i_{m-1} + Q_{fx}q_{m-1} + a_x) \tag{23}$$

$$f_m = \kappa(Q_{vf}v_m + Q_{if}i_{m-1} + Q_{qf}q_{m-1} + a_f) \tag{24}$$

$$q_m = f_m \otimes q_{m-1} + x_m \otimes \tanh(Q_{vq}v_m + Q_{iq}i_{m-1} + Q_{qx}q_{m-1} + a_q) \tag{25}$$

$$g_m = \kappa(Q_{vg}v_m + Q_{ig}i_{m-1} + Q_{qg}q_m + a_g) \tag{26}$$

$$i_m = g_m \otimes \tanh q_m \tag{27}$$

where,  $\otimes$  is element wise multiplication,  $\kappa(v)$  determines sigmoid function,  $Q_{qx}$  is weight matrix among input  $v_m$  and input gate  $x_m$ ,  $q_m$  is cell,  $f_m$  is forget gate, and  $i_m$  be output response. Therefore, the predicted monetary policy results are called  $D$ , where DeepLSTM weights are adjusted to Caviar-HGSO. The DeepLSTM structure is given in figure 3.

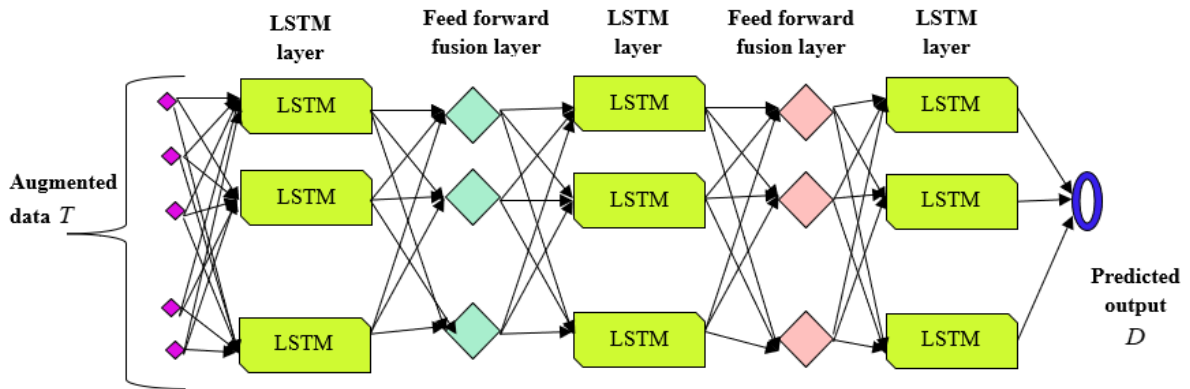


Figure 3. Deep LSTM structure

*Training of DeepLSTM*

The Deep LSTM [13] training by Caviar-HGSO, which is modeled by mixing Caviar and HGSO is explained here. Generally, Caviar models are based on autoregressive schemes for calculating parameters by the regression quantile. The advantage of Caviar is that it produces better results for the time series data. HGSO algorithm is developed based on principle of the Henry's Law and is used to solve different optimization problems. HGSO algorithm simulates degassing behavior to stabilize the mine and explore the search space while avoiding the limited area. The steps of the HGSO are given below.

*Initialization:* The amount of population size  $T$  (gas) and gas location are calculated based on the following equation.

$$K_h(p+1) = K_{min} + v \times (K_{max} - K_{min}) \tag{28}$$

where,  $K_h$  is gas position  $h$  in population  $T$ ,  $v$  is random number between  $[0, 1]$ ,  $K_{max}$  and  $K_{min}$  is boundary limit of issue.

*Fitness calculation:* The fitness function calculates the ideal solution using an error function, as seen in the following equation:

$$FIT = \frac{1}{I} \sum_{v=1}^I [D - \gamma_v]^2 \tag{29}$$

where,  $D$  is DeepLSTM output,  $\gamma$  is target output, and  $I$  is training samples.

*Clustering:* Divide the population agents into smaller groups of equal size, equal to the total gas in the atmosphere. Therefore, each cluster has the same gas classes and the same Henry's values ( $T_f$ ).



*Evaluation:* Each cluster group is analyzed to identify a best gas by relative properties of other gases in their subset. These gases are regulated to obtain good gas for the entire population.

*Renew Henry coefficient:* During this stage, Henry coefficient is upgraded in ideal place when achieving optimal gas. Therefore, Henry's coefficient updated equation is shown below.

$$N_f(p+1) = N_f(p) \times \exp(-G_f \times (1/q(p) - 1/q^v)) \tag{30}$$

$$q(p) = \exp(-p/iter) \tag{31}$$

where,  $N_f$  is Henry coefficient of  $f$  cluster,  $q$  be temperature,  $q^v$  is the constant,  $iter$  is the total iteration count.

*Solubility updation:* Once Henry's coefficient is updated, the gas solution is updated using the following equation.

$$R_{h,f}(p) = K \times N_f(p+1) \times H_{h,f}(p) \tag{32}$$

where,  $R_{h,f}$  is gas solubility  $h$  in cluster  $f$ ,  $H_{h,f}$  represents the partial pressure, and  $B$  is constant.

*Update location:* The gas location is renewed as,

$$K_{h,f}(p+1) = K_{h,f}(p) + V \times n \times \mu \times (K_{h,best}(p) - K_{h,f}(p)) + V \times n \times \varpi \times (R_{h,f}(p) \times K_{best}(p) - K_{h,f}(p)) \tag{33}$$

$$K_{h,f}(p+1) = K_{h,f}(p)(1 - V \times n(\mu + \varpi)) + V \times n \times \mu \times K_{h,best}(p) + V \times n \times \varpi \times R_{h,f}(p) \times K_{best}(p) \tag{34}$$

However, a convergence speed of HGSO is slow, and this solution consumes a lot of memory. To overcome this problem, the Caviar algorithm is combined with the HGSO algorithm to obtain an optimal solution. According to Caviar, the updated equation is:

$$K_{h,f}(p) = \varepsilon_0 + \varepsilon_1 * K_{h,f}(p-1) + \varepsilon_2 * K_{h,f}(p-2) + \varepsilon_1 f(K_{h,f}(p-1) + \varepsilon_2 K_{h,f}(p-2)) \tag{35}$$

Substitute equation (35) to (34),

$$K_{h,f}(p+1) = \left[ \varepsilon_0 + \varepsilon_1 * K_{h,f}(p-1) + \varepsilon_2 * K_{h,f}(p-2) + \varepsilon_1 f(K_{h,f}(p-1) + \varepsilon_2 K_{h,f}(p-2)) \right] (1 - V \times n(\mu + \varpi)) + V \times n \times \mu \times K_{h,best}(p) + V \times n \times \varpi \times R_{h,f}(p) \times K_{best}(p) \tag{36}$$

Thus, the above equation is the final updated Caviar-HGSO equation that predicts the monetary policy, thus shows the effectiveness.

*Computation of feasibility:* To determine the best option, re-evaluate the fit using equation (29).

*End:* Repeat all the above procedures until the ideal solution is found. Table 1 shows pseudocode of the Caviar-HGSO.

**Table 1.** Pseudocode of Caviar-HGSO

<b>Input:</b> $K_h$
<b>Output:</b> $K_{h,f}(p+1)$

Initialization the gas population
Sort gases into different types
Compute whole cluster $f$
Achieve ideal gas $K_{h,best}$ and the ideal searching agent $K_{best}$
<b>While</b> $P < iter$ do
for each search agent do
Locations of all search agents are renewed
<b>End for</b>
Update Henry coefficient of each gas by equation (30) and (31)
The solubility of each gas type is updated by equation (32)
Renew ideal gas $K_{d,best}$ , and best search agent $K_{best}$ by Equation (36)
<b>end while</b>
$p = p + 1$
<b>Return</b> $K_{best}$

DeepLSTM is used to train the Caviar-HGSO, which is designed by combining the advantages of Caviar and HGSO. Therefore, Caviar-HGSO is used to determine the optimal solution to predict the effectiveness of monetary policy.

#### 4. Results and discussion

The below section analyzes results and information on Caviar-HGSO\_DeepLSTM for monetary policy prediction.

##### 4.1 Experimental setup

Caviar-HGSO\_DeepLSTM is tested using PYTHON tool running Windows 10 operating system.

##### 4.2 Dataset description

The National Bureau of Statistics of China [19] is a data set used to predict monetary policy effectiveness. Sample data are collected from 2000 to 2018, which includes currency, macro index, stock market and credit market data. The sample data collected is related to changes in monetary policy. It mainly collects China's monetary policy, macroeconomic indicators and so on for years. The Caviar-HGSO\_DeepLSTM system uses three performance measures, like MSE, MAE, and RMSE for evaluation:

*MAE*: It refers to the error factor calculated from the actual and forecast values using the following expression:

$$MAE = \frac{1}{I} \sum_{v=1}^I [D - \gamma_v] \tag{37}$$

*MSE*: It is evaluated by subtracting expected from predicted value. Next, the MSE expression can be calculated using Equation (29).

*RMSE*: The square root of the mean error value is defined and calculated by the following expression:

$$RMSE = \frac{1}{I} \sum_{v=1}^I [D - \gamma_v]^2 \tag{38}$$

##### 4.3 Sample outcome

Figure 4 shows the results of Caviar-HGSO\_DeepLSTM for predicting real data compared to reconstructed and generated data.

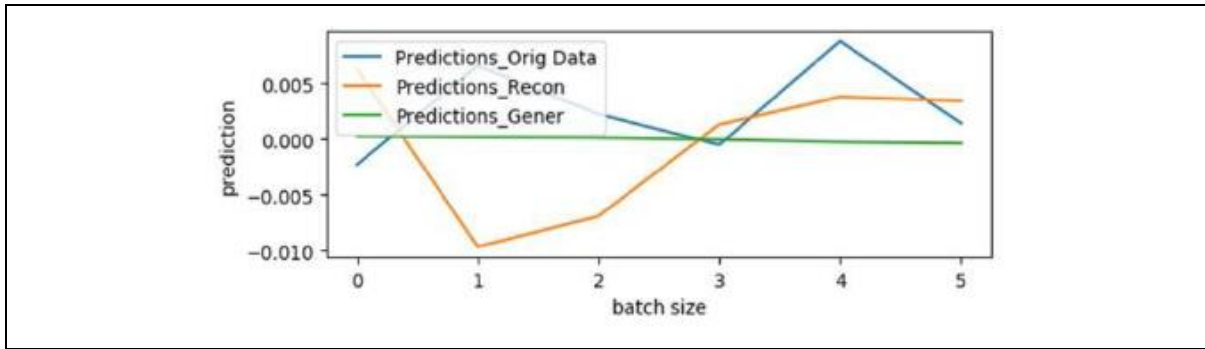


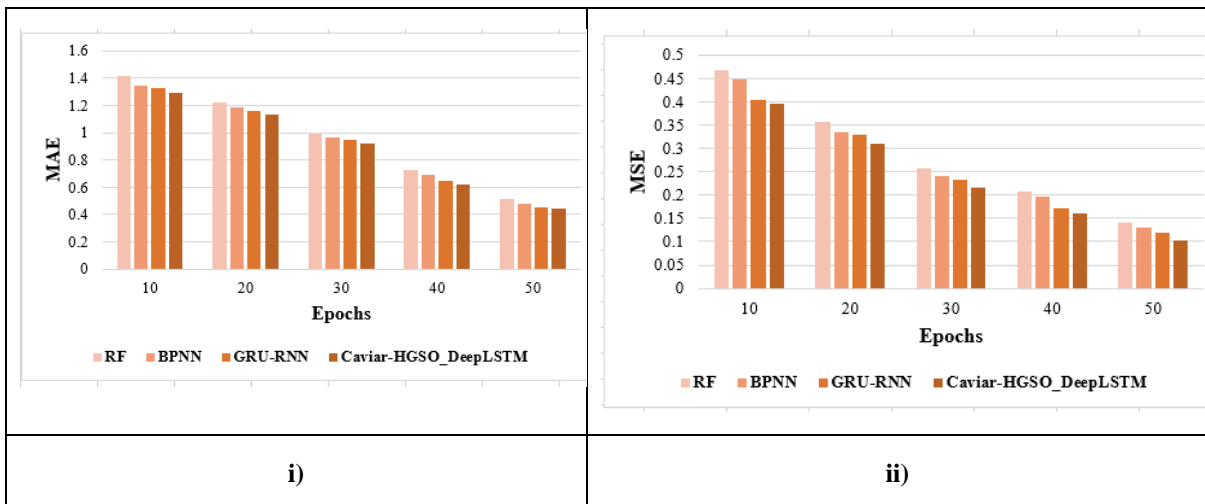
Figure 4. Estimated output

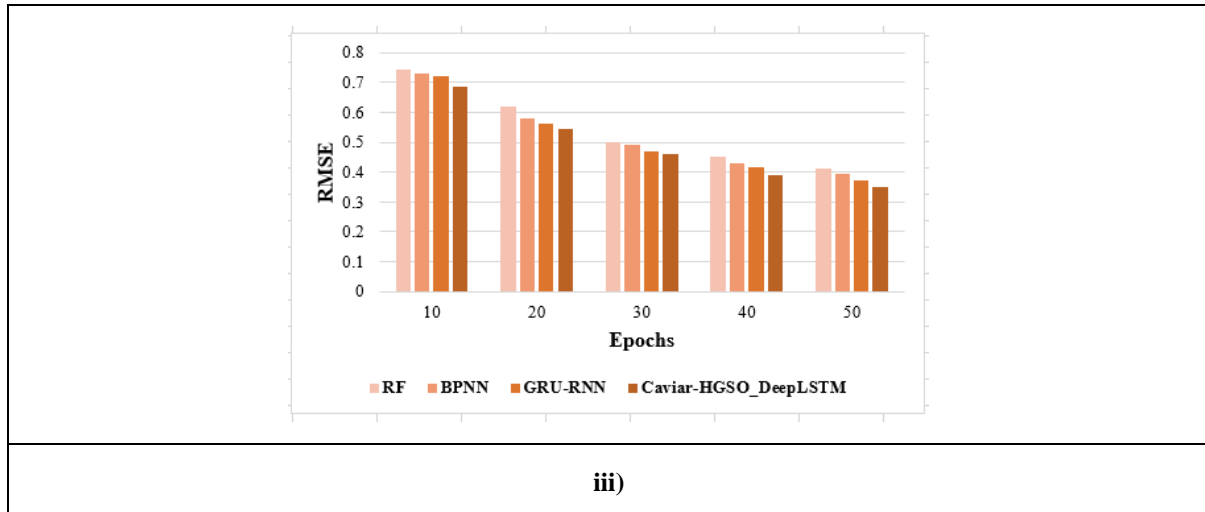
#### 4.4 Comparative methods

The performance increase of Caviar-HGSO\_DeepLSTM is examined in contrast to established approaches, such as RF [6], BPNN [1], and GRU-RNN [5].

#### 4.5 Comparative assessment

Figure 5 depicts a comparison of Caviar-HGSO\_DeepLSTM performance utilizing several criteria. Figure 5i) depicts the MAE evaluation of Caviar-HGSO\_DeepLSTM following epoch changes. Caviar-HGSO\_DeepLSTM achieves an MAE of 1.291, RF of 1.412, BPNN of 1.342, and GRU-RNN of 1.323 for 10<sup>th</sup> epochs. As a result, Caviar-HGSO\_DeepLSTM achieved the lowest MAE among the methods tested in terms of epochs. The MSE analysis is displayed in Figure 5ii). If the epochs are 20, the MSE value measured by Caviar-HGSO\_DeepLSTM is 0.310, while standard approaches have MSE values of 0.356, 0.334, and 0.329. Figure 5iii) depicts the RMSE-based evaluation after adjusting the epochs. The RMSE value obtained by Caviar-HGSO\_DeepLSTM is 1.462, whereas the RMSE values for typical techniques, such as RF, BPNN, and GRU-RNN, are 0.501, 0.490, and 0.471 for epochs=30. The proposed technique can retain the minimal MSE at each epoch, demonstrating the robustness of its prediction. In particular, when noise interference grows more severe, the benefits of the suggested method become more considerable, demonstrating its effectiveness for the data. As a result, the estimates for building costs are more accurate. The results demonstrate that Caviar-HGSO\_DeepLSTM is a decent predictor of monetary policy.





**Figure 5.** Comparative evaluation of Caviar-HGSO\_DeepLSTM, i) MAE ii) MSE iii) RMSE

#### 4.6 Comparative Study

Table 1 compares the assessment methods used for Caviar-HGSO\_DeepLSTM. The MAE, MSE, and RMSE of Caviar-HGSO\_DeepLSTM are 0.438, 0.103, and 0.351, respectively. Other approaches, such as RF, BPNN, and GRU-NN, produce MAE values of 0.517, 0.482, and 0.452, MSE values of 0.142, 0.1313, and 0.119, and RMSE of 0.411, 0.392, and 0.372, respectively. The table below compares Caviar-HGSO\_DeepLSTM to RF, BPNN, and GRU-NN in terms of MAE, RMSE, and MSE. Caviar-HGSO\_DeepLSTM outperforms other models, indicating that it is accurate. Simultaneously, the prior model's error is controlled, resulting in a very good Caviar-HGSO\_DeepLSTM. It can be observed that the monetary policy effectiveness prediction performs better.

**Table 1.** Comparative discussion

Metrics	RF	BPNN	GRU-NN	Caviar-HGSO_DeepLSTM
MAE	0.517	0.481	0.452	<b>0.438</b>
MSE	0.142	0.1312	0.119	<b>0.103</b>
RMSE	0.411	0.392	0.372	<b>0.351</b>

#### 5. Conclusion

This research introduces Caviar-HGSO\_DeepLSTM as a monetary policy prediction model. Caviar-HGSO is a combination of two optimization algorithms, Caviar and HGSO, which alter DeepLSTM weights to accurately predict monetary policy. Furthermore, a feature fusion is done using DBN and the Bray-Curtis distance, which combines the best characteristics for optimal prediction. DBN can be used to address unsupervised learning tasks by reducing the dimensionality of features, as well as supervised learning tasks by building classification or regression models. The estimation results are improved by Caviar-HGSO\_DeepLSTM. Furthermore, the test results reveal that Caviar-HGSO\_DeepLSTM outperforms in terms of MSE, RMSE, and MAE (values of 0.103, 0.351, and 0.438, respectively). The performance of financial market indicators is most evident in the banking market. Monetary policy can encourage economic development and prevent financial risks.

#### References

[1] Minrong Lu, "A monetary policy prediction model based on deep learning", Neural Computing and Applications, vol.32, no.10, pp.5649-5668, 2020.

- [2] Maryam Hajipour Sardui, Mohammadali Afshar Kazemi, Mahmood Alborzi, Adel Azar and Ali Kermanshah, "P-V-L Deep: A Big Data Analytics Solution for Now-casting in Monetary Policy", *Journal of Information Technology Management*, vol.12, no.4, pp.22-62, 2020.
- [3] Betchani H. M. Tchereni, Ahmed Makawa, and Fredrick Banda, "Effectiveness of the Asset Price Channel as a Monetary Policy Transmission Mechanism in Malawi: Evidence from Time Series Data", *International Journal of Economics and Financial Issues*, vol.12, no.5, pp.160, 2022.
- [4] Georgios Georgiadisy and Arnaud Mehl, "Financial Globalisation and Monetary Policy Effectiveness", *Journal of International Economics*, vol.103, pp.200-212, 2016.
- [5] Cheng Yang and Shuhua Guo, "Inflation Prediction Method Based on Deep Learning", *Computational Intelligence and Neuroscience*, 2021.
- [6] Chuanxin Qiu, Tong Li, and Xuelin Qiu, "Research on Forecasting of China's Monetary Policy Based on Random Forest Algorithm", *Journal of Statistical Mechanics: Theory and Experiment*, vol.1549, no.3, 2020.
- [7] Natascha Hinterlang, "Predicting Monetary Policy Using Artificial Neural Networks", 2020.
- [8] Szabolcs Deak, Paul Levine, Afrasiab Mirza, and Joseph Pearlman, "Designing Robust Monetary Policy Using Prediction Pools", 2019.
- [9] Bernanke, B.S., Gertler, M., "Inside the black box: The credit channel of monetary policy transmission. *Journal of Economic Perspectives*", vol.9, no.4, pp.27-48, 1995.
- [10] Cesa-Bianchi, A., Thwaites, G., Vicendoa, A., "Monetary policy transmission in the United Kingdom: A high frequency identification approach", *European Economic Review*, vol.123, pp.103375, 2020.
- [11] Lin EMH, Sun EW, Yu MT, "Systemic risk, financial markets, and performance of financial institutions", , vol.262, no.2, pp.579–603, 2018.
- [12] Platen E, Rebolledo R, "Principles for modelling financial markets. *Journal of Applied Probability*", vol.33, no.3, pp.601-613, 1996.
- [13] Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L. and Xie, X., "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks", In *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.
- [14] Hashim, F.A., Houssein, E.H., Mabrouk, M.S., Al-Atabany, W. and Mirjalili, S., "Henry gas solubility optimization: A novel physics-based algorithm", *Future Generation Computer Systems*, vol.101, pp.646-667, December 2019.
- [15] Lengnick M, and Wohltmann HW, "Optimal monetary policy in a new Keynesian model with animal spirits and financial market", *Journal of Economic Dynamics and Control*, vol.64, pp.148-165, 2016.
- [16] Laurent L. Pauwels and Andrey Vasnev, "Forecast combination for discrete choice models: predicting FOMC monetary policy decisions", *Empirical Economics*, vol.52, pp.229-254, 2017.
- [17] Ganda, F., "The environmental impacts of financial development in OECD countries: a panel GMM approach", *Environmental science and pollution research*, vol.26, no.7, pp.6758-6772. 2019.
- [18] Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A. and Li, Y., "Forecasting price movements using technical indicators: Investigating the impact of varying input window length", *Neurocomputing*, vol.264, pp.71-88, 2017.
- [19] Statistical data: National bureau of statistics of China taken from, "<http://www.stats.gov.cn/>", accessed on March 2024.
- [20] Bc. Ján Vojtech, "Deep Neural Networks and Their Implementation", Thesis, Charles University in Prague, 2016.
- [21] Engle, R.F. and Manganelli, S., "CAViaR: Conditional autoregressive value at risk by regression quantiles," *Journal of business & economic statistics*, vol.22, no.4, pp.367-381, 2004.