Enhancing Neural Network with Particle Swarm: A House Price Prediction Case

Yaotong Liang *

Department of Mathematics, Nanjing University, Nanjing, China

* Corresponding author: 211840071@smail.nju.edu.cn

Abstract. This study delves into the utilization of the Particle Swarm Optimization (PSO) algorithm to discover the best parameters of Long Short-Term Memory (LSTM) model with the aim of housing price prediction. Employing the Ames Housing dataset, our investigation underscores the remarkable efficacy of PSO in mitigating prediction errors within the LSTM framework, thereby enhancing overall predictive performance. The symbiotic integration of PSO with LSTM facilitates the discernment of intricate data patterns, particularly in scenarios involving high-dimensional and nonlinear optimization conundrums. The findings illuminate PSO's ability to expediently traverse and identify global optima. The outcomes unequivocally underscore the PSO-LSTM model's superior attributes, characterized by accelerated convergence, diminished errors, and exceptional proficiency. In summation, this investigation underscores PSO's promise as a potent technique for optimizing machine learning models, notably exemplified by its integration with the LSTM framework.

Keywords: Particle Swarm Optimization, Long Short-Term Memory, housing price prediction, optimization, machine learning.

1. Introduction

Forecasting property values presents a crucial and formidable undertaking within both the real estate sector and the wider economic landscape [1]. Accurate forecasting of house prices is vital for various stakeholders, including homebuyers, sellers, investors, policymakers, and financial institutions. It empowers them to make informed decisions, manage risks, and devise effective strategies.

The traditional housing price models commonly employ the hedonic house price model, which evaluates the relationship between house prices and various features [2]. This model has proven to be efficient, since it necessitates vital data including prices and accurate specification of functional relationships. By utilizing the hedonic price approach, we can calculate the distinct impacts of each housing attribute on housing prices while keeping all other variables unchanged [3]. While valuable insights are offered into predicting housing prices, it also comes with specific limitations.

On one hand, the hedonic price model typically assumes linear relationships, which might not capture complex nonlinear connections [4]. In reality, house prices are often influenced by multiple factors in combination, and these factors do not always exhibit simple linear relationships. On the other hand, spatial heterogeneity also exerts a significant impact on the prediction results [5].

The advent of artificial neural network models has offered new prospects for housing price predictions. Artificial neural network models are sophisticated machine learning methods that emulate the structure of the human brain's neural network [6]. They effectively harness large-scale datasets and adeptly handle nonlinear relationships, endowing them with powerful predictive capabilities in housing price forecasts.

Moreover, deep learning facilitates intricate function approximation by employing a nonlinear network that can extract crucial features even from limited samples [7]. Renowned for their capacity to grasp extended temporal relationships within data, Long Short-Term Memory (LSTM) networks have become prominent, which can lead to noticeable improvements [8]. Nevertheless, the prediction outcomes are directly influenced by Long Short-Term Memory models' parameters. Historically, training these models heavily depended on manual parameter tuning rooted in experience, leading to subpar generality and prediction efficacy [9].

This paper introduces a house price prediction model, namely Particle Swarm Optimization-Long Short-Term Memory (PSO-LSTM), which has the potential to achieve superior accuracy in estimating house prices [10]. Firstly, we establish an LSTM neural network model with parameter initialization. Next, various parameters including iterations, batch size and neuron number in each layer are optimized by Particle Swarm Optimization algorithm. Subsequently, dataset is divided into training and test set to train and fine-tune the model. Ultimately, we compare the errors of LSTM, PSO-LSTM, and other traditional machine learning models.

2. Method

In this thesis, PSO-LSTM is utilized for house price prediction. Firstly, an LSTM model is constructed, and PSO algorithm is employed to tune its parameters. This model reduces prediction errors caused by inappropriate parameters in deep learning models, and the simultaneous adjustment of parameters enables faster convergence.

2.1. LSTM neural network

The LSTM belongs to a distinctive category known as Recurrent Neural Networks (RNNs). RNNs are a set of networks specifically developed to process sequential data. LSTM stands out as an enhanced iteration of RNNs, introduced to tackle the challenges of vanishing and exploding gradients that often plague conventional RNNs.

In traditional RNNs, the hidden state at each time step can only retain short-term historical information, making capturing extended temporal relationships within lengthier sequences a challenging endeavor. The limitation means that RNNs may struggle to effectively remember important information from earlier time steps, thus impacting their performance on certain tasks.

The design of the LSTM neural network aims to tackle these challenges. It introduces a special type of unit called the LSTM cell, which incorporates three crucial gated mechanisms: input, forget and output gate. Each gate controls the information flow within the network and allows it to remember and forget information over extended periods. These gates are governed by sigmoid activation functions that produce values between 0 and 1.

2.1.1. Forget Gate

Forget Gate (Fg) plays a pivotal role in deciding which information to discard from the preceding cell state (C). It is computed as follows:

$$Fg_t = \sigma\left(h_{t-1} * Wm_f + i_t * Um_f + \mathrm{b_f}\right) \tag{1}$$

Where:
- Wm_f represents the weight matrix corresponding to the previous hidden state,
- Um_f represents the weight matrix associated with the current input,
- $\mathrm{h_{t-1}}$ represents the output from the prior time step,
- $\mathrm{i_t}$ represents the current input,
- $\mathrm{b_f}$ represents the bias vector.

2.1.2. Input Gate and Cell Update

Input Gate (Ig) assumes the responsibility of determining the fresh data that will be integrated into the cell state. Furthermore, a prospective cell state (C') is computed by taking into account the present input and the preceding hidden state:

$$Ig_t = \sigma(h_{t-1} * W_i + i_t * U_i + b_i) \tag{2}$$

$$C_t' = tanh(h_{t-1} * W_C + i_t * U_C + b_C) \tag{3}$$

Where:
- Wm_i, Wm_C, Um_i, and Um_C represent weight matrices,
- b_i and b_C represent bias vectors.

2.1.3. Cell State Update

The cell state undergoes an update process that involves factoring in the influence of the forget gate and the candidate cell state:

$$C_t = C_{t-1} * f_t + C_t' * I_t \tag{4}$$

2.1.4. Output Gate

Finally, Output Gate (Og) regulates the information flow from the cell state which contributes to the generation of the ultimate prediction (H) at the present time step:

$$Og_t = \sigma(h_{t-1} * W_o + x_t * U_o + b_o) \tag{5}$$

$$H_t = tanh(C_t) * Og_t \tag{6}$$

Where:
- W_o and U_o represent weight matrices,
- b_o represents the bias vector.

The detailed structure of LSTM is shown in the following Fig. 1.

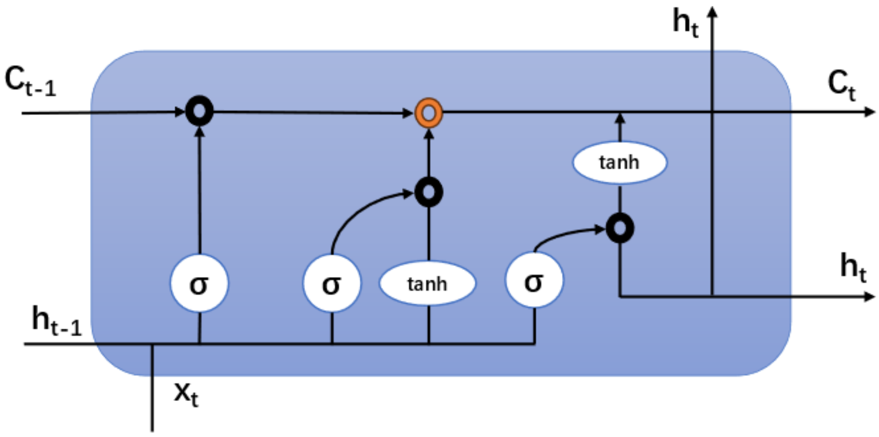

Figure 1. The structure diagram of the LSTM unit

2.2. Standard PSO algorithm

PSO draws its roots from the communal behavior observed in bird flocks, offering a population-centered optimization approach. It mimics the collective intelligence exhibited by bird flocks to explore the optimal solution within a specified search area. This paper introduces PSO algorithm as a powerful tool to optimize the LSTM neural network for house price prediction.

The PSO algorithm commences with the initialization of a population composed of particles, each representing a possible solution within the designated exploration range. All particles are dispersed at random across the exploration space and possess distinct positions and velocities. The particle's position denotes a particular configuration of parameters for the LSTM model, while the velocity signifies the direction of movement toward improved solutions. Subsequently, the fitness of each particle is appraised through the computation of its objective function value, specifically the Mean Absolute Error (MAE) between projected house prices and actual house prices. A reduced MAE indicates higher fitness for the particle.

Throughout the optimization procedure, all particles modify both their velocities and positions grounded in the prevailing location, the most favorable position attained thus far (referred to as individual best, pb), and the optimum position achieved by particles within the entire population (termed global best, gb). The velocity update formula for particle i at the subsequent time step ti+1 is expressed as follows:

$$v_i^{(t+1)} = w * v_i^{(t)} + ac_1 * random * \left(pb_i - x_i^{(t)}\right) + ac_2 * random * \left(gb_i - x_i^{(t)}\right) \quad (7)$$

Here, $v_i^{(t+1)}$ presents the updated velocity of the particle i, w stands for the inertia weight, ac_1 and ac_2 denote acceleration coefficients, $random$ signifies a random number ranging from 0 to 1, pb_i signifies the finest individual position of particle i while $x_i(t)$ reflects the present position, and gb_i captures the most exceptional global position reached by all particles.

Following the adjustment of velocities for all particles, the corresponding positions of these particles are subsequently modified. If a particle's new position is outside the boundaries of the search space, it is clipped to stay within the valid range. The PSO algorithm iteratively performs velocity and position updates for a predefined number of iterations or until a termination criterion is met. This process allows the particles to converge towards the optimal solution in the search space, with aligns with the identification of the optimal parameter configuration for the LSTM model.

By utilizing the Particle Swarm Optimization to tune the parameters of LSTM model, issues can be addressed of inappropriate parameter settings that may lead to larger prediction errors. Furthermore, the parallel optimization of multiple particles expedites convergence, bolstering predictive efficacy of proposed PSO-LSTM approach, thus contributing to precise and efficient house price forecasting.

2.3. LSTM Model Optimized by PSO

The construction of the LSO-LSTM model involves the following steps:

1. Parameter and Particle Population Initialization:

Initialize the parameters in the LSTM network, including neurons number in each layer. a cluster of particles is generated at random, with each particle denoting a distinct parameter configuration for the LSTM model. These particles' positions and velocities are distributed randomly across the exploration domain, representing different candidate solutions.

2. LSTM Model Training and Particle Fitness Calculation:

Train the LSTM model for all particles and compute their levels of fitness, which is evaluated by comparing the model's predictions on the training data with the corresponding ground truth values. This paper uses Mean Absolute Error (MAE).

3. Individual and Population Best Fitness Determination:

Determine the best fitness for each particle as well as the overall best fitness for the entire particle population. The individual best fitness is updated by comparing each particle's fitness value, while the global best fitness is determined based on the best fitness attained by any member within the particle population.

4. Particle Velocity and Position Update:

Revise the velocity and position based on the optimal performance of each particle and the most exceptional overall achievement. The velocity update determines the direction of movement in the search space, while the position update represents the new position of each particle. This way, the particle population gradually converges towards the global best solution.

5. Check for Maximum Generation Times:

Verify if the predefined maximum iteration count has been attained. If not, continue with Steps 2 to 4, repeating the optimization process. If the maximum generation times have been reached, use the optimized parameters to construct the final LSTM model and train it on the dataset.

By following these steps, the LSO-LSTM model, through the collaboration of the particle population and fitness optimization, efficiently searches for better LSTM model parameters, thereby enhancing the accuracy and performance of house price prediction.

The combination is shown below in Fig. 2.

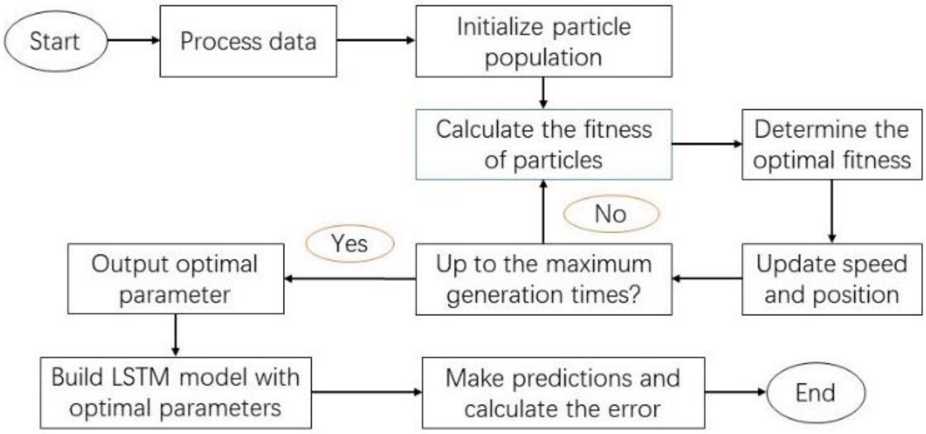

Figure 2. The PSO-LSTM flowsheet

3. Data and Results

3.1. Data Processing

In this study, we obtained a CSV copy of the data from https://www.kaggle.com/prevek18/ames-housing-dataset. Data processing includes the following steps: Firstly, dealing with missing values. For certain columns and specific values, designated strategies were utilized for filling in the absent data using the ‘fillna’ method. In specific cases, conditional statements and mapping functions were utilized for filling in the absent data. For instance, for rows with non-missing values in the "Garage Type" column and missing values in the "Garage Qual" column, the value in the "Garage Type" column was set as "No Garage". Secondly, handling outliers. Specifically, five data points with GrLivArea (above-ground living area in square feet) greater than 4000 were removed to mitigate the impact of outliers on data statistics. Next, categorical variables underwent conversion into one-hot encoding to facilitate model utilization. Subsequently, the dataset was divided into training subsets and test subsets, maintaining a ratio of 3:1, to assess the model's performance.

3.2. LSTM model construction:

This paper constructed an LSTM neural network consisting of 4 layers. In the initial LSTM layer, 64 neurons were designated, while the subsequent layer consisted of 16 neurons. Both layers were equipped with the 'relu' activation function for data fitting. Mean Absolute Error (MAE) was used as the evaluation metric for the model. For predicted values (y) and their corresponding true values (y_i), MAE was calculated as follows for all n samples:

$$MAE = \frac{\Sigma_{i=1}^{n}|\hat{y}_i - y_i|}{n} \tag{8}$$

The fitting curve loss curve and the comparison plot between predicted values and true values are shown below in Fig. 3.

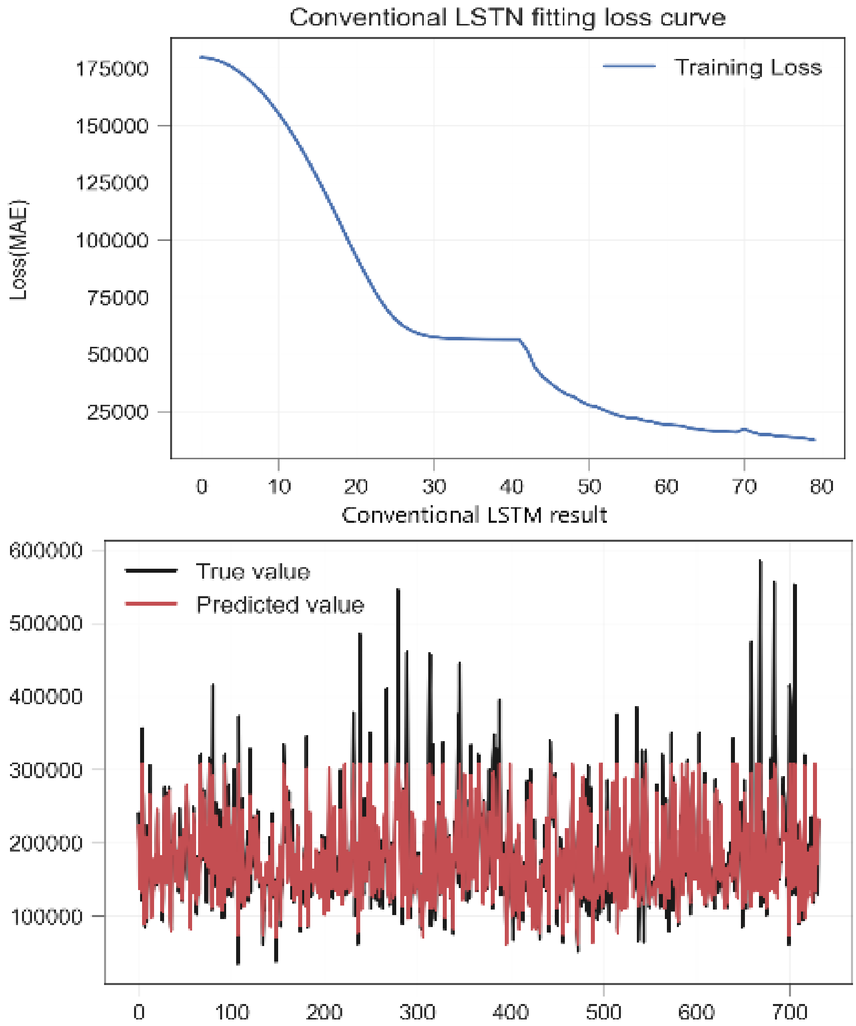

Figure 3. Results of conventional LSTM model

3.3. Particle Swarm Optimization

After PSO, we obtained optimized parameter data, which includes iteration, batchsize, neuron number of two LSTM layers, neuron number of the fully connected layer, and the learning rate. These optimized parameters were used for building the LSTM model for data fitting. The study sets the maximum iteration of the PSO model to 20. The allowable range for the six parameters is defined by the subsequent upper and lower bounds (See Table 1):

Table 1. The upper and lower bounds for the six parameters

	Iteration	Batchsize	Neurons (First layer)	Neurons (Second layer)	Neurons (FC layer)	Learning rate
Upper bound	60	100	100	100	100	0.01
Lower bound	10	10	10	10	10	0.001

For the parameters c1, c2, and w in the PSO algorithm, this study follows the work of Marini and Walczak [11] and sets c1 = c2 = 2 and w = 0.7. The results are shown in Fig. 4 and Fig. 5.

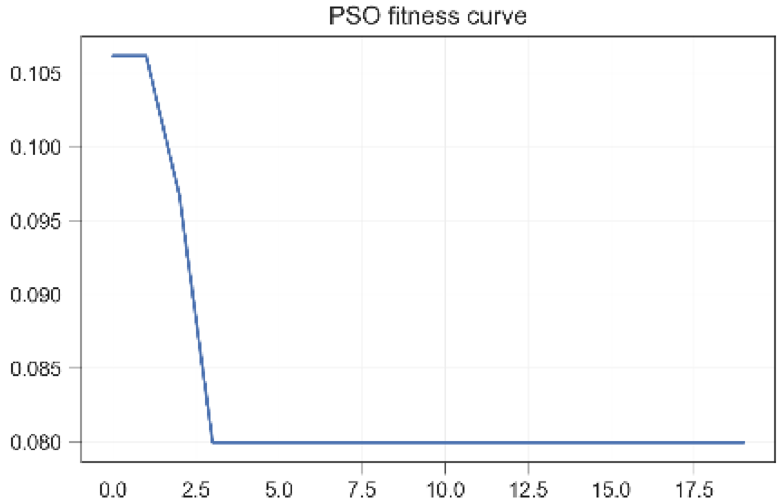

Figure 4. Fitness curve of PSO

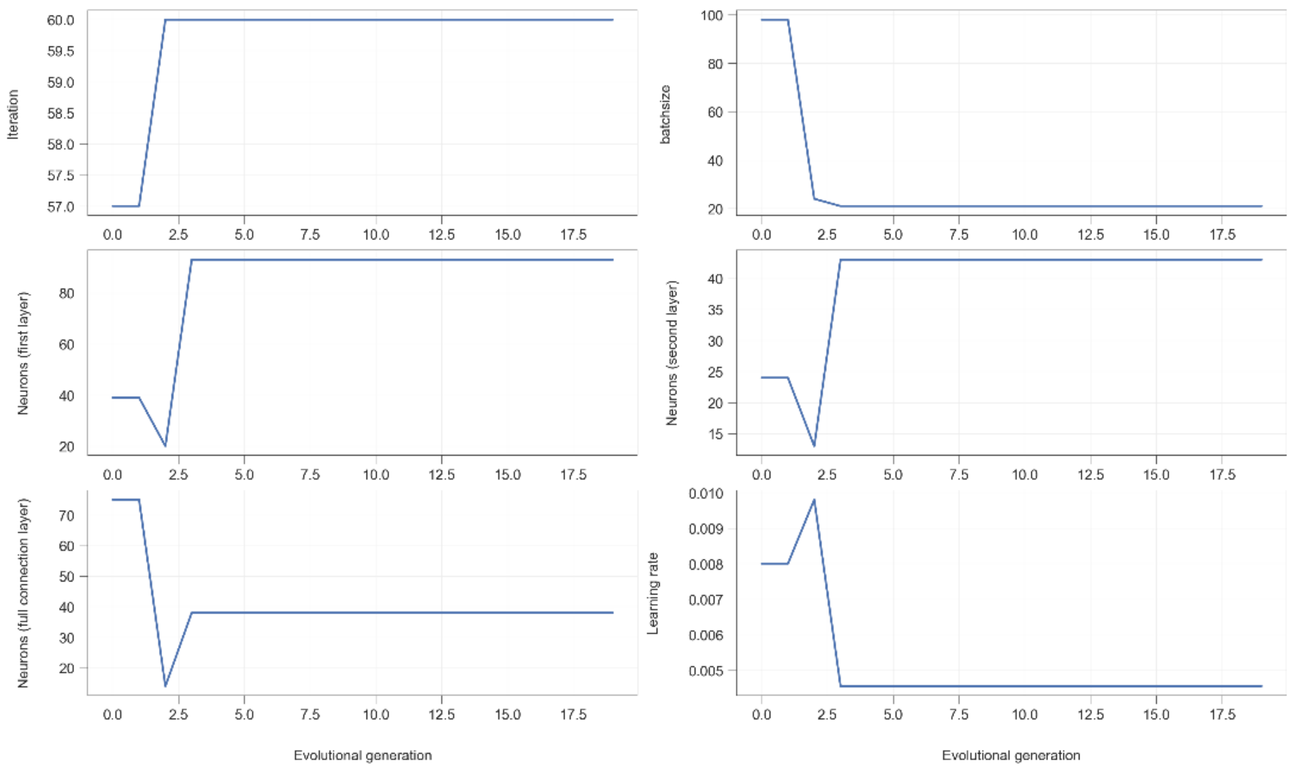

Figure 5. Iterative process of parameters

The above represents the optimization outcome of the PSO model. From PSO fitness curve graph, it can be observed that the model converged within the maximum iteration.

3.4. PSO-LSTM model construction

After PSO, this paper obtained the optimized parameter data in Table 2:

Table 2. The optimal value of parameters

Iteration	Batchsize	Neurons (first layer)	Neurons (second layer)	Neurons (full connection layer)	Learning rate
61	21	93	41	38	$4.55*10^{-3}$

In this paper, these parameters were used to fit the LSTM model. The results are as follows in Fig. 6:

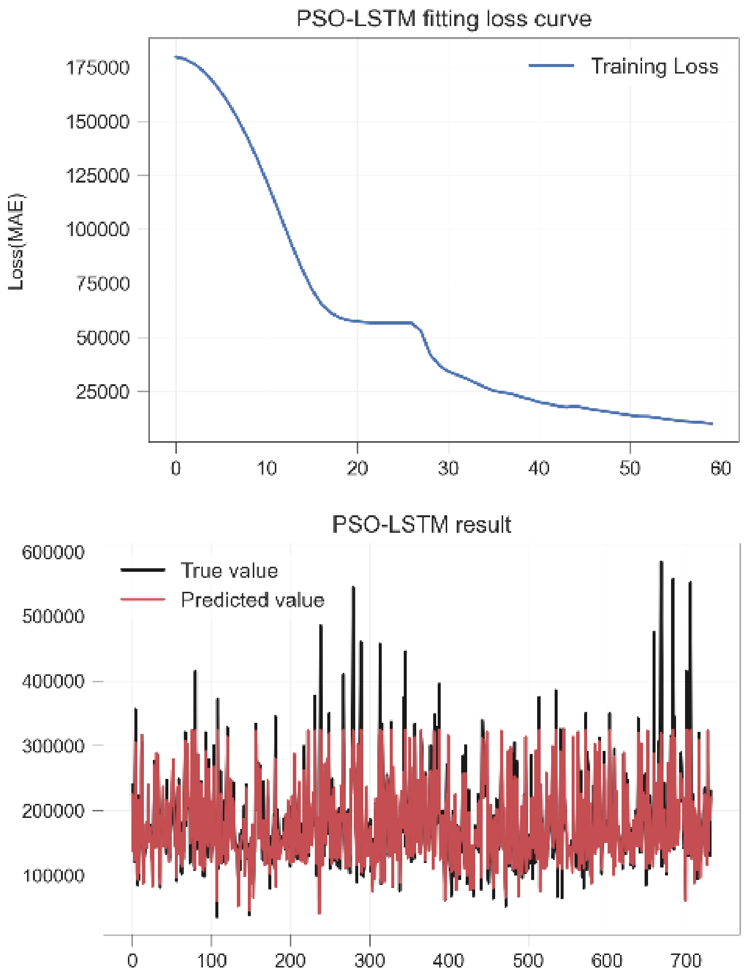

Figure 6. Results of PSO-LSTM model

3.5. Comparison of different models

Prior to Particle Swarm Optimization, various machine learning models were employed for fitting, including Support Vector Regression, Extreme Gradient Boosting, Decision Tree, Random Forest, Ridge Regression and K-Nearest Neighbors. MAE was utilized as the evaluation metric, and the error results for different models were obtained. The results are as follows in Fig. 7.

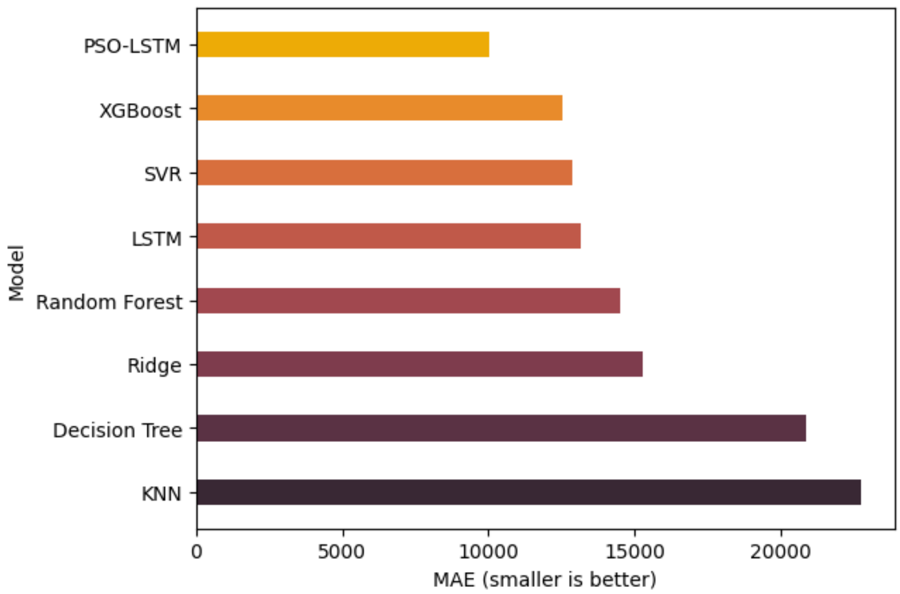

Figure 7. Comparison of different models' performance

This study compared the fitting results of different models, and the outcomes indicated a notable reduction in the error of the LSTM model through the application of the PSO algorithm, culminating in a substantially diminished MAE in contrast to all alternative models.

4. Conclusion

This research delved into the utilization of Particle Swarm Optimization to fine-tune the parameters of the LSTM model for housing price prediction. The findings underscore the efficacy of PSO in diminishing the LSTM model's error and elevating its predictive capabilities. This conclusion highlights the significance of PSO as a powerful optimization technique in the field of machine learning.

While the current study successfully demonstrated the effectiveness of PSO in optimizing the LSTM model, there are still several areas for further exploration and investigation:

1. Adaptive Particle Swarm Optimization (APSO): The study showed the potential of APSO in further optimizing the LSTM model. Future research could focus on exploring the benefits of APSO in other machine learning models and comparing its performance against traditional PSO and other optimization algorithms.

2. Generalization across Datasets: The current study utilized the Ames Housing Dataset for housing price prediction. Further research should validate PSO-optimized LSTM network’s performance across diverse datasets to assess its generalization ability.

3. Versatility: PSO can be easily combined with other machine learning algorithms, providing a versatile approach for enhancing model performance across various applications. Its ability to efficiently fine-tune parameters makes it a valuable tool in a wide range of machine learning tasks.

In conclusion, our study highlights the effectiveness of Particle Swarm Optimization in optimizing the LSTM model for housing price prediction. PSO's ability to efficiently search for global optimal solutions and its potential for further enhancement through adaptive approaches offer promising opportunities for future research and practical applications in machine learning and optimization domains.

References

[1] Börsch-Supan, A. Household formation, housing prices, and public policy impacts. Journal of Public Economics, 1986, 30 (2): 145 - 164.

[2] Goodman, A. C. Hedonic prices, price indices and housing markets. Journal of urban economics, 1978, 5 (4): 471 - 484.

[3] Chau, K. W., & Chin, T. L. A critical review of literature on the hedonic price model. International Journal for Housing Science and its applications, 2003, 27 (2): 145 - 165.

[4] Fletcher, M., Gallimore, P., & Mangan, J. Heteroscedasticity in hedonic house price models. Journal of Property Research, 2000, 17 (2): 93 - 108.

[5] Can, A. Specification and estimation of hedonic housing price models. Regional science and urban economics, 1992, 22 (3): 453 - 474.

[6] Limsombunchao, V. House price prediction: hedonic price model vs. artificial neural network. 2004, https: //researcharchive.lincoln.ac.nz/handle/10182/5198.

[7] Heddam, S. Development of air–soil temperature model using computational intelligence paradigms: artificial neural network versus multiple linear regression. Modeling Earth Systems and Environment, 2019, 5 (3): 747 - 751.

[8] Chen, X., Wei, L., & Xu, J. House price prediction using LSTM. arXiv preprint arXiv, 2017: 1709.08432.

[9] Poon, H., & Domingos, P. Sum-product networks: A new deep architecture. In 2011 IEEE International Conference on Computer Vision Workshops, 2011: 689 - 690.

[10] Gao, X., Guo, Y., Hanson, D. A., Liu, Z., Wang, M., & Zan, T. Thermal error prediction of ball screws based on PSO-LSTM. The international journal of advanced manufacturing technology, 2021, 116 (5-6): 1721 - 1735.

[11] Marini, F., & Walczak, B. Particle swarm optimization (PSO). A tutorial. Chemometrics and Intelligent Laboratory Systems, 2015, 149: 153 - 165.