Particle Swarm Optimization-based CNN-LSTM Networks for Forecasting Energy Consumption

Tae-Young Kim
Department of Computer Science
Yonsei University
Seoul, Republic of Korea
taeyoungkim@yonsei.ac.kr

Sung-Bae Cho
Department of Computer Science
Yonsei University
Seoul, Republic of Korea
sbcho@yonsei.ac.kr

***Abstract*—Recently, there have been many attempts to predict residential energy consumption using artificial neural networks. The optimization of these neural networks depends on the trial and error of the operator that lacks prior knowledge. They are also influenced by the initial values of the model based on the gradient algorithm and the size of the search space. In this paper, different kinds of hyperparameters are automatically determined by integrating particle swarm optimization (PSO) to CNN-LSTM network for forecasting energy consumption. Our findings reveal that the proposed optimization strategy can be used as a promising alternative prediction method for high prediction accuracy and better generalization capability. PSO achieves effective global exploration by eliminating crossover and mutation operations compared to genetic algorithms. To verify the usefulness of the proposed method, we use the household power consumption data in the UCI repository. The proposed PSO-based CNN-LSTM method explores the optimal prediction structure and achieves nearly perfect prediction performance for energy prediction. It also achieves the lowest mean square error (MSE) compared to conventional machine learning methods.**

Keywords—Particle swarm optimization, Convolutional neural network, Long short-term memory, Deep learning

I. INTRODUCTION

Over the past several decades, energy consumption is accelerating rapidly due to fundamental changes in the industry and the economy [1]. With this trend, electricity consumption forecasting has become an integral step in the automated management of power systems. Electricity energy must be consumed at the same time as it is generated in the power plant due to its expensive storage costs [2]. Excessive estimation of energy consumption leads to unnecessary idle capacity, while under estimation would increase the operating costs of the vendor and cause potential energy interruption [3]. Electric energy consumption prediction includes several time series variables. In addition, the electric power demand represents various patterns including irregular time series components. Therefore, it is difficult to predict electrical energy consumption using classical prediction methods [4]. It is also difficult to model variables because it involves complex nonlinear patterns between variables gathered from various sensors [5]. Recently, neural network technology is recognized to be more useful than existing statistical prediction model because it can easily map nonlinear function.

Many kinds of neural networks have been proposed to predict energy consumption, but inadequate network structures are inefficient for practical applications. Techniques to search appropriate hyperparameters are manual and very complex. In addition, these neural networks can nonlinearly map complex energy consumption variables, but it is difficult to remember previous memories. Classical neural networks therefore suffer from performance degradation due to overfitting as the data becomes larger and more complex over time.

TABLE 1. TYPES OF HYPERPARAMETERS

Model	*Hyperparameter*	*Range*
Convolutional neural network	Learning rate	10^{-5}~10^{-1}
	No. of filters	2^0~2^8
	Filter size	2~7
	Pooling size	2~7
	Dropout rate	0.5~0.9
Recurrent neural network	Learning rate	10^{-5}~10^{-1}
	Cell type	GRU, LSTM
	Layer size	2^0~2^8
	Dropout rate	0.5~0.9

Recently, conventional MLP-based neural networks have been further improved by convolutional neural networks (CNN) and recurrent neural networks (RNN). CNN learns the weights of feature maps consisting of layers, extracts high-level features of the input data, and preserves the relationships between the different variables [6]. Similarly, an RNN has a circular linking structure and stores dynamic features in a hidden memory. It updates transient information over time to ensure continuity [7]. In energy consumption, CNN extracts input features surrounded by temporal contexts in order to reduce spectrum variations of variables. The LSTM layer stores changes in time series data over time and models complex energy consumption trends. Also, a few fully connected DNN layers, used in conjunction with CNN-LSTM, transform multi-dimensional space to generate predicted power consumption [8]. However, the complexity of neural networks using CNN and LSTM increases the number of hyperparameters that need to be adjusted. It is also influenced by the initial values of the model based on the

978-1-7281-2153-6/19/$31.00 ©2019 IEEE

gradient algorithm and the size of the search. Table 1 shows the hyperparameters used for CNN and LSTM neural networks. Figure 1 briefly represents the search space according to the hyperparameter of the neural network. The x-axes and y-axes in Figure 1 can represent the number of filters or the number of hidden units. The z-axis represents the performance according to each hyperparameter combination.

Manually adjusting the hyperparameter is undoubtedly one of the most important steps in the machine learning workflow [9]. The technique of retrieving hyperparameters requires tremendous computational resources, so it is almost impossible to reproduce experiments [10]. The difficulty of finding the appropriate combination of hyperparameters lies in a complex interaction between the parameters. The conventional naïve search method is to set a reasonable range for each hyperparameter and try different values for each. This is called grid search. It skips the search space slightly for each iteration [11]. However, it is likely to skip between certain intervals, which is likely to produce high performance values. Random search works on existing continuous and non-discriminatory functions to solve the disadvantages of grid search [12]. Random search is more likely to find a global optimum compared to a grid search. However, increasing the number of independent variables can degrade the performance of model.

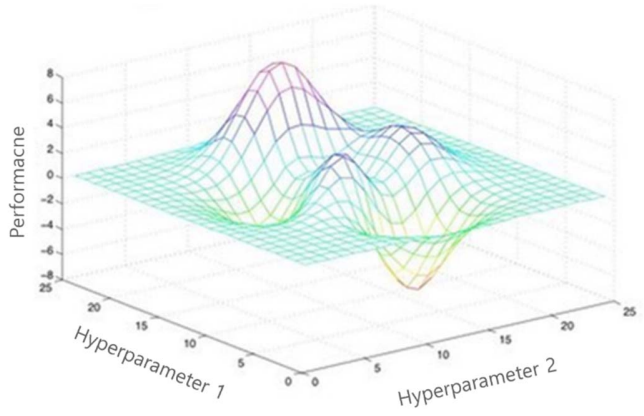


Fig. 1. Search space according to hyper parameters

Recently, a genetic algorithm was used to solve the search problem. It is a meta-heuristic optimization algorithm inspired by natural evolution [13]. Selection and mutation operations of genetic algorithms based on the concept of survival of the fittest and evolution are used to optimize the hyperparameters of neural networks. Genetic algorithms aim to simulate the evolution of optimal hyperparameter encoding over several generations. Genetic algorithms evaluate the fitness of each hyperparameter over the generations and choose the optimal hyperparameter. Genetic operators such as crossover and mutation are then used to evolve selected encodings and generate new encodings. The genetic algorithm is returned to the solution of the optimization problem by selecting the best hyperparameter encoding in the generated population. It can improve the variability of hyperparameter encoding by crossover and mutation operations, but it has many redundancies, and low computational efficiency.

In this paper, we propose a hyperparameter selection method using PSO to optimize the structure of CNN-LSTM. PSO is an algorithm for simulating social behavior. It is a population-based heuristic algorithm that formats and expresses the movement of organism in a bird flock or fish school [14]. PSO eliminates crossover and mutation operations of the GA algorithms to reduce computational costs and reach robust, fast global optimization. It also saves computing resources because it uses a simple heuristic operation for computation.

In this study, we optimize the number of CNN kernels, the number of hidden units in LSTM and the number of units in fully connected layer of CNN-LSTM model for electricity energy consumption prediction using PSO. It is the first application of PSO to optimize the CNN-LSTM model for electric energy consumption prediction. We encode each hyperparameter into a genetic form, taking into account the number of all cases. We have observed through experiments that the CNN-LSTM structure for exploring through the PSO achieves the highest performance in the problem of forecasting energy consumption. The performance of the CNN-LSTM model optimized on the basis of PSO is higher than 10% compared with the model in which the hyperparameters are not properly tuned. The dataset we have experimented in this paper is the dataset of household power consumption provided in the UCI repository.

The remainder of this paper is organized as follows: Section II presents the relevant works of structural optimization in neural networks for predicting electricity consumption. Section III describes the method for selecting CNN-LSTM of learning hyperparameters for forecasting energy consumption using PSO. Section IV shows the experiment results and discussion, and finally Section V concludes this paper and discusses future works.

II. Related Works

A. Search Algorithm

Energy consumption prediction requires a learning process and various neural network structures depending on the dataset. The neural network learning is set a group of hyperparameter so that we can adjust the behavior of model for a given problem. Therefore, we can have various neural network structures by adjusting hyper parameters, and finding the best combination of parameters in a hyperparameter can be considered as a problem. The process of exploring hyperparameters requires manual engineering, but it is time-consuming and tedious. As the choice to design the architecture of neural networks increases, the existing manual methods have been replaced by search algorithms. Typical algorithms for finding hyperparameters are grid search and random search. Figure 2 shows the hyperparameter search methods of manual search, grid search, and random search [15, 16]. Grid search is a hyperparameter adjusting method that searches for combinations of parameters within a range specified by a grid and evaluates the neural network model systematically [11]. Grid search is based on a priori knowledge, analyzing the problem and determining the scope of the grid, so it is possible to search efficiently. However, there is a disadvantage that the optimal hyperparameter combination may be outside the specified range. To overcome the disadvantages of grid search, the idea of random search was

proposed [12]. Random search is different from grid search. Thus, the range of discrete values provided to explore hyperparameter is broader. It provides a statistical distribution for each hyperparameter that can randomly sample values. It also efficiently searches for optimal hyperparameter combinations within a predetermined range compared to the greedy algorithm. However, simple random sampling is intended as a non-bias approach, but sample selection bias can occur.

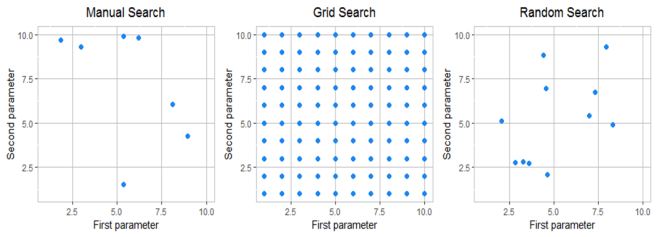


Fig. 2. Search algorithm for hyperparameter space

B. Genetic Algorithm

TABLE 2. RELATED WORKS BASED ON GENETIC ALGORITHM

Author	***Year***	***Method***	***Dataset***
Bouktif et al [20].	2018	Long short-term memory	Electricity energy consumption
Li et al [21].	2015	Artificial neural network	Building energy consumption
Jung et al [22].	2015	Support vector machine	Building energy consumption
Uzlu et al [23].	2014	Artificial neural network	Electricity energy consumption
Boithias et al [24].	2012	Artificial neural network	Electricity energy consumption
Qing et al [25].	2010	Radial basis function	Electricity energy consumption

Genetic algorithm provides an opportunity to randomly search the hyperparameter space in the neural network models for electricity energy prediction and searches for optimal parameter encoding based on previous results [27]. There are two categories of previous studies using genetic algorithms with neural networks [17]. The first category is to replace the back propagation used in the learning process of neural networks based on GA [18]. This method searches the optimal weight when the structure of the neural network is simple, but produces lower performance than the back propagation method as the neural network becomes complicated [28]. It consists of genetic algorithms applied to optimize the large hyperparameter space that can be found in complex deep neural networks [19].

Figure 3 shows the optimization of neural networks using genetic algorithms. The simple neural networks have been shown to use genetic algorithms to empirically evaluate and adjust the hyperparameter to optimize the performance. Table 2 shows an optimization study based on genetic algorithms for forecasting power demand.

To date, evaluations using genetic algorithms to optimize hyperparameters have focused only on small data sets and simple methods, making it difficult to identify the usefulness of genetic algorithms for more difficult problems. The effect of this method of finding a series of hyperparameters in a small neural network suggests that it can be used in deeper and more complex networks.

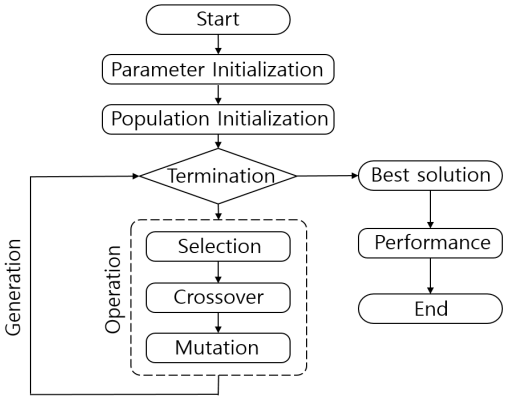


Fig. 3. Optimization step using genetic algorithm

Bouktif *et al*. proposed a model of predicting short-term load with machine learning and LSTM-based neural networks [20]. They solved the problem using GA to find optimal time lag and window size to improve short-term prediction accuracy. Li *et al*. suggested a model of optimized ANN for time-dependent prediction of building power consumption [21]. They applied genetic algorithms to adjust the weights and thresholds of artificial neural network structures. They used principal component analysis (PCA) to select key input variables and a search algorithm to simplify the model structure. Jung *et al*. proposed a squared support vector machine strategy. They made an important contribution to the prediction of building energy consumption in the way suggested [22]. They also used existing coded genetic algorithms to determine the parameters. Uzlu *et al.* proposed an artificial neural network model combined with a genetic algorithm to estimate Turkey's energy consumption [23]. They also attempted to outperform the performance of backpropagation algorithm using optimization algorithm. Boithias *et al*. suggested a GA-based method to optimize the architecture and training parameters [24]. Qing *et al*. used genetic algorithm and RBF neural network to predict power consumption and optimize parameters of RBF neural network by introducing genetic algorithm [25].

Conventional genetic algorithms have crossover and mutation operations, so that hyperparameter combinations can be optimized. However, hyperparameter combinations are very large and complex, so that the amount of computation increases. Therefore, it is necessary to search for hyperparameters efficiently, but also to optimize the computational complexity.

III. The Proposed Method

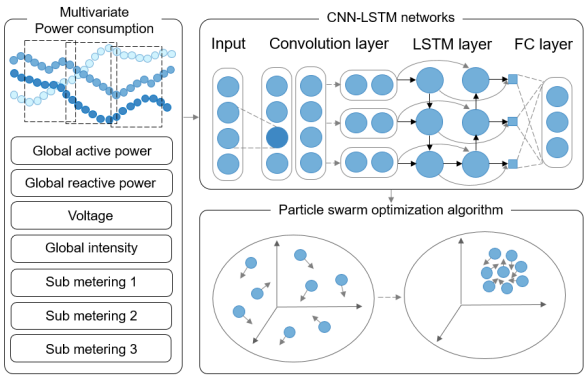


Fig. 4. The overall structure of the proposed method

A. Overview

Figure 4 shows the overall architecture for predicting household power consumption using a PSO-based CNN-LSTM model. The PSO-based CNN-LSTM learns input data that has been pre-processed by the sliding window algorithm in advance. The spatial properties of a multivariate time series variable are extracted from the convolution and pooling layers of CNN and passed to LSTM layer. The LSTM layer models non-regular time information using transmitted spatial features. At this time, the hyperparameters constituting CNN-LSTM are optimized by using PSO. We encode the number of CNN kernels, the hidden unit of LSTM, and the number of units of fully connected layer that consist of CNN-LSTM. The PSO finds the optimal hyperparameter structure while moving the encoded particles. Finally, the CNN-LSTM method can generate predicted electrical energy consumption in fully connected layers.

B. CNN-LSTM Networks

CNN-LSTM is a linear combination of the CNN layer for the correlation between variables and the LSTM layer for temporal feature modeling. The preprocessed power demand data are passed through a layer composed of convolution and pooling and automatically extracts important features from multivariate that affect power consumption. $x_i^0 = \{x_1, x_2, \cdots, x_n\}$ is the global active power representing the power consumption data, and n represents the 60-minute unit pre-processed by the sliding window algorithm. Equation (1) is the result of the feature vector y_{ij}^1 output from the first convolutional layer, y_{ij}^1 is calculated by output feature vector x_{ij}^1 of the previous convolution layer, b_j^1 represents the bias for the j^{th} feature map, w is the weight of the convolution kernel, m is the index value of the convolution filter, and σ is the activation function. Equation (1) also represents the result of the feature vector y_{ij}^l output from the l^{th} convolutional layer.

Features extracted from the convolution layer are passed to the pooling layer for sampling. In particular, because the pooling layer chooses the largest value, the amount of computation required to predict power demand can be significantly reduced. Equation (2) represents the operation of the max-pooling layer. R is the pooling size, and T is the stride that decides how far to move pooling region.

$$y_{ij}^1 = \sigma(b_j^1 + \sum_{m=1}^{M} w_{m,j}^1 x_{i+m-1,j}^0) \quad (1)$$

$$p_{ij}^l = \max_{r \in R} y_{i\times T+r,j}^{l-1} \quad (2)$$

CNN-LSTM method represents the correlation between energy consumption variables through the CNN layer and delivers the noise-canceled features to the LSTM layer. The LSTM layer consists of an input gate, a forget gate and an output gate as shown in Equations (3), (4), and (5). Each gate is represented by the i, f, and o. The LSTM layer can be used to easily model long-term electric energy consumption. Each gate unit controls a continuous value between 0 and 1, so that the hidden value of the LSTM cell, h_t is updated every t steps. Equations (6) and (7) are used to determine cell states and hidden states. These are represented by c and h. σ is an activation function. The term p_t is the output of CNN and used as an input to determine the state of the LSTM cell. W is a weight matrix, and b is a bias vector.

$$i_t = \sigma(W_{pi}p_t + W_{hi}h_{t-1} + W_{ci} \circ c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{pf}p_t + W_{hf}h_{t-1} + W_{cf} \circ c_{t-1} + b_f) \quad (4)$$

$$o_t = \sigma(W_{po}p_t + W_{ho}h_{t-1} + W_{co} \circ c_t + b_o) \quad (5)$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \sigma(W_{pc}p_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$h_t = o_t \circ \sigma(c_t) \quad (7)$$

Finally, the last layer of CNN-LSTM consists of fully connected layers to generate electric energy consumption. $h^l = \{h_1, h_2, \ldots h_l\}$, where l is the number of units in LSTM. We set l to 60 to generate a 60-minute power consumption. Equation (8) shows the equations used in the fully connected layer, where σ is the activation function, w is the weight of the i^{th} node for layer $l-1$ and the j^{th} node for layer l, and b_i^{l-1} represents a bias.

$$d_i^l = \sum_j w_{ji}^{l-1} (\sigma(h_i^{l-1}) + b_i^{l-1}) \quad (8)$$

C. Particle Swarm Optimizaion

In this paper, we use PSO algorithm that automatically search hyperparameters to improve the efficiency of power demand prediction. The PSO is an algorithmic representation of the movement of particles. It searches for a space using different particles and updates the result when an optimal position is found by a specific particle. We set the kernel size, LSTM, and output size of fully connected layer as hyper parameters. 8 bits are assigned to each hyperparameter, and a chromosome composed of 32 bits in total is used. Figure 5 shows the encoding of the chromosome. The fitness function is evaluated using Equation (9)

$$Fitness\ function = 1 - \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (9)$$

The PSO searches for hyperparameter space using particles that swarm like birds flock. In general, each particle moves in the direction of optimizing the hyperparameters. The heuristic PSO algorithm tracks the movement of all particles and ultimately aims to find the optimal position of all the particles in the hyperparameter space. We place particles at arbitrary positions in the hyperparameter space and assign a small initial velocity. The speed of each particle is affected by the speed and direction of the other particles. We use the PSO algorithm to update the motion of the particles over the generations. Each particle changes its velocity, inertia, and position to find the optimal position. We also set the objective function to optimize electric energy consumption. Over the generations and time, each particle is brought together to optimize the CNN-LSTM model. The PSO algorithm keeps track of the best past location of each particle so that neighboring particles can be influenced to navigate to a good location. We update each particle using Equation (10).

$$v_i(t+1) = (c_1 \times rand() \times (p_i^{best} - p_i(t))) + (c_2 \times rand() \times (p_g^{best} - p_i(t))) + v_i(t) \quad (10)$$

$$p_i(t+1) = p_i(t) + v_i(t) \quad (11)$$

$v_i(t+1)$ is the updated velocity for the i^{th} particle. The p_i^{best} is the best individual position with a uniformly random variable. p_g^{best} is the best position of all particles. The c_1 and c_2 are the acceleration coefficients for the personal and global best positions, respectively. A particle's position is updated using Equation (11). $p_i(t)$ is the i^{th} particle's position at time t. Figure 6 provides a pseudocode of the PSO algorithm for maximizing the fitness function.

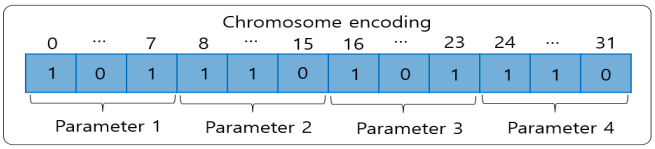


Fig. 5. Chromosome representation of hyperparameters

Algorithm: Particle Swarm Optimization Algorithm

Input: $Problem_{size}$, $Population_{size}$
Output: P_{g_best}
$Population \leftarrow 0$
$P_{g\_best} \leftarrow 0$
For ($i = 1$ *to* $Population_{size}$)
  $P_{velocity} \leftarrow Random\ Velocity()$
  $P_{position} \leftarrow Random\ Position(Population_{size})$
  $P_{p\_best} \leftarrow P_{position}$
  If ($Cost(P_{p\_best}) \leq Cost(P_{g\_best})$)
    $P_{g\_best} \leftarrow P_{p\_best}$
While (!StopCondiction())
  For ($P \in Population$)
    $P_{velocity} \leftarrow Update\ Velocity(P_{velocity}, P_{g\_best}, P_{p\_best})$
    $P_{position} \leftarrow Update\ Position(P_{position}, P_{velocity})$
  If ($Cost(P_{p\_best}) \leq Cost(P_{g\_best})$)
    $P_{p\_best} \leftarrow P_{position}$
    If ($Cost(P_{p\_best}) \leq Cost(P_{g\_best})$)
      $P_{g\_best} \leftarrow P_{p\_best}$
return P_{g_best}

Fig. 6. Pseudo code for particle swarm optimization

IV. Experiments

A. Dataset and Experimental Setting

In this paper, we evaluated the PSO-based CNN-LSTM method using the dataset of individual household electric power consumption in the UCI machine learning repository [10]. This dataset represents the amount of power consumption collected per minute. It also includes several variables that affect power demand. Each of the variables represents a time series and affects power consumption prediction. We remove 25,979 missing values from 2,075,259 measurements in data preprocessing. The data is preprocessed with a sliding window algorithm for 60 minute unit prediction. The preprocessed data includes electricity consumption in the kitchen, laundry room, electric water heater and an air-conditioner as well as date and time variables. To use preprocessed data as input, the training and test data are organized in two dimensions, including multivariate variables. We used the preprocessed training and test data to evaluate the PSO-based CNN-LSTM model. Table 3 shows the variables for each household power consumption dataset.

Table 3. The Features of Power Consumption Dataset

Category	**#**	***Attribute***
Time information	1	Day
	2	Monte
	3	Year
	4	Hour
	5	Minute
Sensor information	6	Global active power
	7	Global reactive power
	8	Voltage
	9	Sub metering 1
	10	Sub metering 2
	11	Sub metering 3

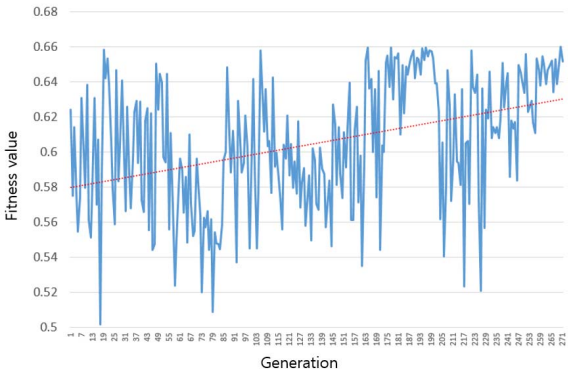


Fig. 7. Fitness value per generation

B. PSO Performance Evaluation by Generation

In order to show the usefulness of the proposed algorithm, the fitness using PSO was measured in each generation. For the experiment, a total of 270 generations were run and the gene of encoding the hyperparameters of CNN-LSTM was optimized. Figure 7 shows the fitness with PSO in the problem of forecasting residential power consumption in each generation. The number of generations for PSO increases the fitness for

predicting power demand. In the early generations, the fitness value gets declined, but as the generation goes by, the fitness value increases. We used the average fitness of each particle to obtain the fitness value.

C. Performance Comparison with Other Methods

In order to verify the usefulness of forecasting the power demand using the proposed PSO based CNN-LSTM, we conducted a performance comparison experiment using the machine learning methods [26]. We also compared the performance of the conventional CNN-LSTM structure manually tuned to verify that the PSO algorithm stably optimizes the CNN-LSTM model. We performed 10-fold cross-validation by dividing the preprocessed data for experiments. Figure 8 shows the performance through 10-fold cross-validation. We evaluated the performance using the MSE metric. The proposed method outperforms the conventional machine learning methods. The PSO-based CNN-LSTM is followed by a linear regression model (LR) and a random forest regression (RF) with low MSE. Figure 8 also shows a more stable performance distribution than the manually adjusted CNN-LSTM structure. We confirm that the PSO-based CNN-LSTM method is useful for estimating the actual residential power demand.

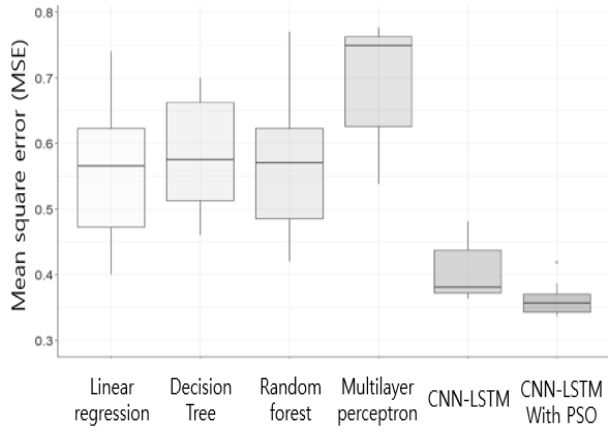


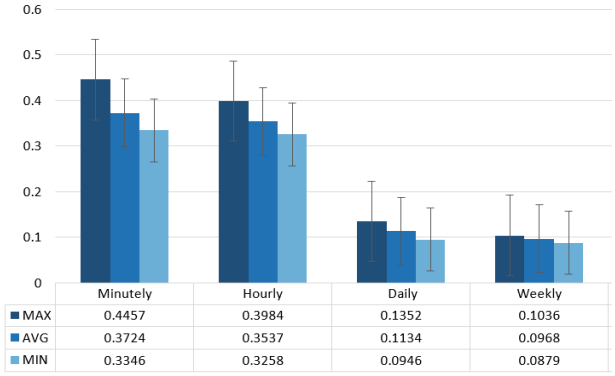
Fig. 8. The accuracy of 10-fold cross-validation

D. Performance Optimization based on Time Resolution

We checked the optimization performance of PSO-based CNN-LSTM with time resolution. We have aggregated for minutely, hourly, daily, and weekly resolution. Figure 9 shows the performance of the PSO-based CNN-LSTM with time resolution. The max value in this figure shows the predicted performance of CNN-LSTM optimized by PSO. The min value represents the lowest performance that the non-optimized model represents. Figure 9 shows that the minimum performance is about 10% difference compared to the optimized performance. The lower the resolution, the smaller the performance that can be optimized, but the trend is the same.

E. Prediction Results for PSO-based CNN-LSTM Network

We compared the proposed method with the linear regression model to qualitatively evaluate the prediction results of the proposed PSO-based CNN-LSTM method. Figure 10 shows the predictions of the PSO-based CNN-LSTM prediction and the linear regression analysis. Figure 10 also represents that the CNN-LSTM optimized by PSO models local features better than the linear regression model. Therefore, it can be confirmed that the proposed method accurately predicts irregular power consumption.



	Minutely	Hourly	Daily	Weekly
MAX	0.4457	0.3984	0.1352	0.1036
AVG	0.3724	0.3537	0.1134	0.0968
MIN	0.3346	0.3258	0.0946	0.0879

Fig. 9. Optimize performance based on resolution

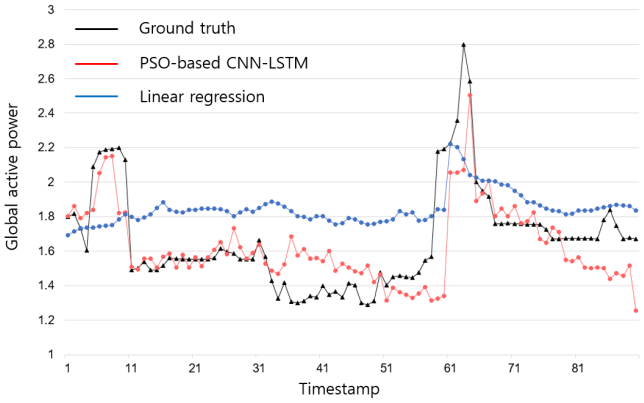


Fig. 10. Graph of power consumption prediction for performance comparison

F. Performance in terms of Chromosomes

Figure 11 shows the expected results of the CNN-LSTM with PSO network according to the chromosomes. We used the PySwarm library to manipulate genes by moving the particles from generation to generation. The x-axis in Figure 11 represents the MSE and y-axis represents the encoding of the chromosomes. The visualization shows that the chromosomes converge according to the objective function. As the generation increases, the encoded hyper parameter structure converges. Depending on the type of gene, the performance can vary by up to 15%. Therefore, we can confirm that PSO is effective.

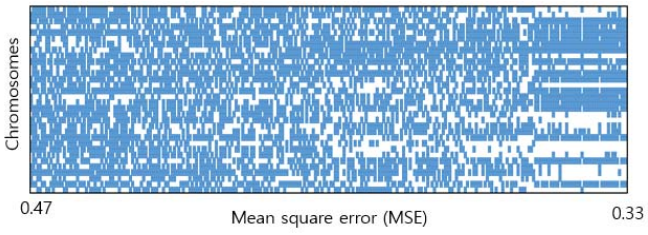


Fig. 11. Chromosome change by performance

V. CONCLUSION

In this paper, we propose a PSO-based CNN-LSTM neural network for predicting residential power demand efficiently. In order to verify the usefulness of the proposed method, we conducted experiments using the dataset of household power consumption in the UCI repository and achieved the lowest RMSE compared with the conventional methods. We used the proposed method to search for optimal hyperparameters of CNN-LSTM neural networks. In addition, the utility of the method was verified by analyzing the power consumption chromosomes influencing the optimal CNN-LSTM neural network learning. Future research will apply an evolutionary algorithm to determine the activation function that converts the input signal of the neural network into the output signal.

ACKNOWLEDGMENT

This research was supported by Korea Electric Power Corporation (Grant number: R18XA05).

REFERENCES

[1] E. M. Oliveira, and F. L. C. Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776-788, 2018.

[2] C. Deb, F. Zhang, J. Yang, S. E. Lee, and K. W. Shah, "A review on time series forecasting techniques for building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 74, pp. 902-924, 2017.

[3] H. Ibrahim, A. Ilinca, and J. Perron, "Energy storage systems—Characteristics and comparisons," *Renewable and sustainable energy reviews*, vol. 12, no. 5, pp. 1221-1250, 2008.

[4] M. I. Ahmad, "Seasonal decomposition of electricity consumption data," *Review of Integrative Business and Economics Research*, vol. 6, no. 4, pp. 271-275, 2017.

[5] X. Lü, T. Lu, C. J. Kibert, and M. Viljanen, "Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach," *Applied Energy*, vol. 144, pp. 261-275, 2015.

[6] C. A. Ronao, ans S.-B. Cho,"Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235-244, 2016.

[7] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222-2232, 2017.

[8] T.-Y. Kim, and S.-B. Cho, "Predicting the household power consumption using CNN-LSTM hybrid networks," *In Int. Conf. on Intelligent Data Engineering and Automated Learning*, pp. 481-490, 2018.

[9] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of recurrent network architectures," *In Int. Conf. on Machine Learning*, pp. 2342-2350, 2015.

[10] T. Domhan, J. T. Springenberg, and F. Hutter, "Speeding up automata networks by extrapolation of learning curves," *In IJCAI* vol. 15, pp. 3460-3468, 2015.

[11] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *In Advances in Neural Information Processing Systems*, pp. 2546-2554, 2011.

[12] J. Bergstra, and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, pp. 281-305, 2012.

[13] M. Suganuma, S. Shirakawa, and T. Nagao, "A genetic programming approach to designing convolutional neural network architectures," *In Proc. of the Genetic and Evolutionary Computation Conference*, pp. 497-504, 2017.

[14] B. Qolomany, M. Maabreh, A. Al-Fuqaha, A. Gupta, and D. Benhaddou, "Parameters optimization of deep learning models using particle swarm optimization," *In Int. Wireless Communications and Mobile Computing Conference*, pp. 1285-1290, 2017.

[15] C. Li, Z. Ding, D. Zhao, J. Yi, and G. Zhang, "Building energy consumption prediction: An extreme deep learning approach," *Energies*, vol. 10, no. 1-20, pp. 1525. 2017.

[16] P. A. Gonzalez, and J. M. Zamarreno, "Prediction of hourly energy consumption in buildings based on a feedback artificial neural network,". *Energy and Buildings*, vol. 37, no. 6, pp. 595-601, 2005.

[17] S. R. Young, D. C. Rose, D. C., T. P. Karnowski, S. H. Lim, and R. M. Patton, "Optimizing deep learning hyperparameters through an evolutionary algorithm," *In Proc. of the Workshop on Machine Learning in High-Performance Computing Environments*, pp. 1-5, 2015.

[18] P. Verbancsics, and J. Harguess, "Image classification using generative neuro evolution for deep learning," *In IEEE Conf. on Applications of Computer Vision*, pp. 488-493, 2015.

[19] E. Cantú-Paz, and C. Kamath, "An empirical comparison of combinations of evolutionary algorithms and neural networks for classification problems," *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 915-927, 2005.

[20] S. Bouktif, A. Fiaz, A. Ouni, and M. Serhani, "Optimal deep learning LSTM model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches," *Energies*, 2018.

[21] K. Li, C. Hu, G. Liu, and W. Xue, "Building's electricity consumption prediction using optimized artificial neural networks and principal component analysis," *Energy and Buildings*, vol. 108, pp. 106-113, 2015.

[22] H. C. Jung, J. S. Kim, and H. Heo, "Prediction of building energy consumption using an improved real coded genetic algorithm based least squares support vector machine approach," *Energy and Buildings*, vol. 90, pp. 76-84, 2015.

[23] E. Uzlu, M. Kankal, A. Akpınar, and T. Dede, "Estimates of energy consumption in Turkey using neural networks with the teaching–learning-based optimization algorithm," *Energy*, vol. 75, pp. 295-303, 2014.

[24] F. Boithias, M. El Mankibi, and P. Michel, "Genetic algorithms based optimization of artificial neural network architecture for buildings indoor discomfort and energy consumption prediction," *In Building Simulation*, vol. 5, no. 2, pp. 95-106, 2012.

[25] Q.-W. Zang, Z.-H. Xu and J. Wu, "Prediction of electricity consumption based on genetic algorithm-RBF neural network,". *In Int. Conf. on Advanced Computer Control*, vol. 5, pp. 339-342, 2010.

[26] T.-Y. Kim, and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems with Applications*, vol. 106, pp. 66-76, 2018.

[27] Y.-G. Seo, S.-B. Cho, X. Yao, "The impact of payoff function and local interaction on the N-player iterated prisoner's dilemma," *Knowledge and Information Systems*, vol. 2, no. 4, pp. 461-478, 2000.

[28] S.-B. Cho, and K. Shimohara, "Evolutionary learning of modular neural networks with ge-netic programming," *Applied Intelligence*, vol. 9, no. 3, pp. 191-200, 1998.