Enhancing Electrical Load Forecasting Accuracy in Poultry Farms: A Hybrid Approach Integrating Fuzzy Signature Particle Swarm Optimization and Neural Network

Lailis Syafaah*, Novendra Setiawan, Amrul Faruq, Mauridhi Hery Purnomo

Abstract— **Power systems are encountering technical challenges due to increased electrical load variability and the integration of renewable energy technologies, which necessitate accurate short-term electrical load forecasting for effective dispatching instructions, spot market management, and system anomaly detection. This paper introduces a novel hybrid method, the Fuzzy Signature Particle Swarm Optimization Neural Network (FSPSO-NN), designed to enhance electrical load forecasting models. The Fuzzy Signature is utilized to dynamically optimize the inertia weight of the Particle Swarm Optimization (PSO) algorithm, incorporating various variables within the PSO to assess performance during execution. This method is specifically implemented for forecasting the electrical load of poultry farms using an Internet of Things (IoT) data logger system. Experimental validation against a range of PSO adjustment algorithms and conventional Artificial Neural Network (ANN) approaches demonstrates that the proposed FSPSO-NN algorithm significantly improves forecasting accuracy, achieving a Root Mean Square Error (RMSE) of 0.0864 on normalized data. These results indicate that FSPSO-NN offers superior performance and accuracy compared to existing methods, making it a valuable tool for modern power system management.**

***Index Terms*— Fuzzy Signature; Particle Swarm Optimization; Neural Network; Forecasting; Electrical Load.**

I. INTRODUCTION

RENEWABLE energy power facilities harness inherent natural resources like wind, hydro, and solar radiation to generate electrical power. These power plants leverage the environmental conditions associated with these renewable sources, utilizing the kinetic energy of wind, the gravitational force of flowing water, and the radiant energy from the sun to produce electricity. The combination of several types of renewable energy power plants is called a hybrid renewable energy power plant [1]. Although environmentally friendly, renewable energy power plants have a very fluctuating level of electricity production depending on the surrounding natural phenomena [2]. Therefore, an appropriate energy resource management plan is needed to be able to balance the electrical power generated with the demand for power consumed so that fluctuating electrical energy production can be overcome [3]. One way to be able to balance electrical power with demand power in conditions of fluctuating electricity production is with accurate load forecasting in the future. Load forecasting aims to identify patterns of electricity consumption by processing historical data on electricity consumption shown in the daily load curve. Load forecasting aims to identify patterns of electricity consumption by processing historical data on electricity consumption shown in the daily load curve [4]. Load forecasting is divided into three classifications, including short-term load forecasting (STLF), medium-term load forecasting (MTLF), and long-term load forecasting (LTLF) [5, 6]. Of the three classifications, short-term load forecasting is the most popular method used [7]. Short-term load forecasting is used in this study because changes in daily electricity consumption have significant variations.

Artificial intelligence computation has been widely used for forecasting problems. Such as in [8], long-term forecast of energy consumption by using Artificial neural network (ANN) with load, temperature, wind speed, humidity, solar flux, and energy generation per hour as input. The output from this model is reported in mean absolute percentage error (MAPE) and root mean square error, where the result shows the value of forecasted sample close to the actual value. Furthermore, prediction of solar energy and load helps to fulfil economic benefits. Moreover, [9] made the load predictions by using ANN and Ensemble Models. The results from the proposed model analysed and compared based on MAPE, mean absolute error (MAE), and daily peak error forecast. The research shows that the ANN has a better performance

Manuscript received December 6, 2023; revised October 16, 2024.

This work was supported Kedaireka, Indonesian Ministry of Education, Culture, Research, and Technology in 2022 Matching Fund Program.

Lailis Syafaah is an Associate Professor of the Electrical Engineering Department, Faculty of Engineering, Universitas Muhammadiyah Malang, Indonesia (lailis@umm.ac.id).

Novendra Setyawan is a Lecturer in the Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Malang, Indonesia (novendra@umm.ac.id).

Amrul Faruq is a Lecturer in the Department of Electrical Engineering, Faculty of Engineering, Universitas Muhammadiyah Malang, Indonesia (faruq@umm.ac.id).

Mauridhi Hery Purnomo is a Professor of Artificial Intelligence and Deep Learning in Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia (hery@ee.its.ac.id)

when a new data test is applied. However, the main limitation of using ANN is the weighting, and the properties of ANN must be appropriately selected so that the performance of ANN can be optimized. The optimizing performance of ANN using an optimization algorithm has been proposed in the past. In [10] combined ANN using Particle Swarm Optimization (PSO) in robotic vision approach. Jaya optimization algorithm is used in ANN weight learning method for time series data forecasting have been proposed in [11]. The modified PSO with a time-varying coefficient is proposed in [12] to adjust ANN weighting. The other optimization algorithm such as Genetic Algorithm, Ant Colony Optimization, and Simulated Annealing is used to improve the performance of ANN. Compared to the different optimization algorithms, the main reason for implementing PSO as an optimization algorithm of ANN is the simplicity and several set parameters. Although many previous works have reported on the use of PSO to optimize ANN, it is believed that the accuracy of the previously proposed methods can still be improved if the PSO get the optimal weight of ANN. Since the conventional PSO algorithm didn't provide local and global search mechanism through the learning process in each iteration, another PSO variant is proposed with inertia parameter introduction [13–15].

The inertia parameter update mechanism improved the PSO to obtain rapid convergence. These characteristics parameters are advantageous to complex optimization problems that use many parameters and have difficulty obtaining analytical solutions. Various update mechanisms of inertia parameter in PSO optimization methods have been proposed. The Adaptive Inertia Weight PSO (AIWPSO) using the percentage of success in every iteration is proposed in [16]. The early and most popular update mechanism is Linear Decreasing Weight PSO (LDWPSO), that using linear decrease of iteration as the update parameter of inertia [17]. The Adaptive PSO (APSO) is proposed by A. Djoewahir and etc using the comparison of global best and personal best parameter in PSO [18]. The early introduction of Fuzzy Signature as the inertia weight adjustment strategies of PSO is proposed in [19] because of simplicity and multivariable capability. In previous study the development of ANN using conventional PSO optimization [10, 11] is not enough to reach global optimum to obtain the forecasting accuracy compared to the PSO with the adaptive inertia update mechanism [16].

This paper proposed an optimization algorithm of Neural Network using Fuzzy Signature PSO (FSPSO) for electrical load of poultry farm. The advantages of FSPSO is the improvement of search ability through Fuzzy Signature inertia update mechanism that measures the condition of PSO particles through many variables which not provided by other PSO variant [19]. In this paper, the best type of inertia update mechanism of PSO method combined with ANN could be compared and identified, which may improve the electrical load forecasting accuracy.

II. Artificial Neural Network

The neuron serves as the fundamental building block of Artificial Neural Networks (ANNs), establishing connections with other neurons to form a network. Each individual neuron comprises components such as inputs, input weights, activation functions, and outputs. This paper focuses on elucidating the relationship between inputs and outputs, which mirrors the connection between previously recorded electrical loads and their forecasted future trends. This connection is established using an ANN model, trainable through real-world experiences by means of sample data during the training process.

In the realm of intricate system characterization, numerous neurons can be interlinked to constitute a network structure featuring multiple layers, termed a multi-layer neural network (MLNN). Within an MLNN, input data denoted as Xi,d undergoes processing via a set of j-neurons situated in hidden layers. These layers encompass essential elements like weighting factors (Whj,i) and biases (Bhj), as expounded upon in (1).

The sigmoid function is employed as the activation function for the hidden layer, as depicted in (2). Subsequently, the subsequent layer of the multi-layer neural network (MLNN) receives inputs from the preceding layer, accompanied by specific weighting and bias terms, exemplified in (3). This activation function can be consistently applied, mirroring the function utilized in the previous layer, as illustrated in (4).

$$Zh_{j,d} = \sum_{i=1,j=1}^{I,J} Wh_{j,i} X_{i,d} + Bh_j \tag{1}$$

$$Yh_{j,d} = f(Zh_{j,d}) = \frac{1}{1 + exp(-Zh_{j,d})} \tag{2}$$

$$Zo_{k,d} = \sum_{k=1,j=1}^{K,J} Wo_{k,j} Yh_{j,d} + Bo_k \tag{3}$$

$$Yo_{k,d} = f(Zo_{k,d}) = \frac{1}{1 + exp(-Zo_{k,d})} \tag{4}$$

The learning procedure for the multi-layer neural network (MLNN), as defined by equations 1 and 3, involves optimizing the network's neurons by adjusting the weights (Wh (j, i), Wo (k, j)), as well as the biases (Bhj, Bhk). This optimization process is deemed successful when the MLNN's output closely approximates the desired target output. To enable efficient learning in a Multi-Layer Neural Network (MLNN), the Backpropagation method is employed, which leverages the gradient descent algorithm. This combination lies at the heart of training neural networks. The gradient descent algorithm is an optimization technique that iteratively adjusts the weights within the network to minimize the error between the predicted and actual output values.

III. Particle Swarm Optimization

Particle Sarm Optimization (PSO), initially introduced by Kennedy and Eberhart [20], emulates the social dynamics observed in biological swarm behaviors, reminiscent of the coordinated movements seen in flocks of birds or schools of fish. In this simulation, each bird or fish is symbolized as a particle, mirroring their flight or swimming patterns during biological activities. As the swarm collectively navigates, for instance, during food search, an exchange of information takes place within the social context. This amalgamation of shared knowledge and individual experience enhances the ongoing search process.

The PSO algorithm commences with the initialization of each particle's position and velocity, denoted by (5) and (6), respectively.

$$x_{i,d} = \begin{bmatrix} x_{i1} & x_{i2} & \dots & x_{id} \end{bmatrix} \quad (5)$$

$$v_{i,d} = \begin{bmatrix} v_{i1} & v_{i2} & \dots & v_{id} \end{bmatrix} \quad (6)$$

where xid ∈ [lb, ub] is particle position, vi is particle velocity, d ∈ D, lb is the lower boundary of search with the D-dimension, and ub is the upper limit of the search with the D-dimension. Several trajectories change the position of a particle with a certain speed based on the experience of the particle and the group. In the PSO, the particle position moves according to the mechanism in (7) and (8).

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 r_1 \left(p_{id}(t) - x_{id}(t)\right) + c_2 r_2 \left(g_d(t) - x_{id}(t)\right) \quad (7)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t) \quad (8)$$

where c_1 and c_2 are non-negative acceleration constants, r1 and r2 are two random numbers in range [0,1], p_{id} is personal best that is the experience of individual particles and g_d is global best that is the experience of particle groups. At first, w is a parameter with a constant value according [21]. However, the increase in PSO search performance is less than optimal. As it develops, there are mechanisms for changing the settings of inertia depending on varying conditions that occur in swarm learning. The weight change mechanism aims to regulate the distribution and search capabilities of the PSO algorithm. In [17], they propose the linear decreasing function to changes the parameter. However, better distribution and search are maintained by changing the weights adaptively, as shown in a study carried out by [19]. Even the w is changed linearly or adaptively, its varies in range (0 < w ≤ 1).

IV. Proposed Fuzzy Signature Neural Network

Fuzzy signature is a simple multivariable of fuzzy logic system which is introduced by Koczy [22]. The generalized form of vector fuzzy is used in fuzzy signature that can be represented as the vector in (9) or a tree structure such as Fig. 1.

$$\boldsymbol{x} = \begin{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} \\ \begin{bmatrix} x_{21} \\ \begin{bmatrix} x_{221} \\ x_{222} \\ x_{223} \end{bmatrix} \\ x_{23} \end{bmatrix} \\ \begin{bmatrix} x_{31} \\ x_{32} \end{bmatrix} \end{bmatrix} \quad (9)$$

Equation (9), $[x_{11} \quad x_{12}]$ and $[x_{31} \quad x_{32}]$ is a higher-level sub-group from structure which is $\boldsymbol{x_1}$ and $\boldsymbol{x_3}$. The $[x_{221} \quad x_{222} \quad x_{223}]$ is the lowest level that will be combined into $\boldsymbol{x_{22}}$. The sub-group or branch was computed first than connected until the higher level in structure which is $\boldsymbol{x} = [\boldsymbol{x_1} \quad \boldsymbol{x_2} \quad \boldsymbol{x_3}]$. The combination of the sub-group was done using some aggregation functions such as max, min, and mean functions. Consider that a_1 is the aggregation function that was resulting $\boldsymbol{x_1}$; hence, $\boldsymbol{x_1} = a_1(\boldsymbol{x_{11}}, \boldsymbol{x_{12}})$. In combining the sub-group, aggregation function could be similar or different from one another. Usually, the $min, max,$ and $mean$ function is used as the aggregation function such as in [23], using that function in compute the sub-group.

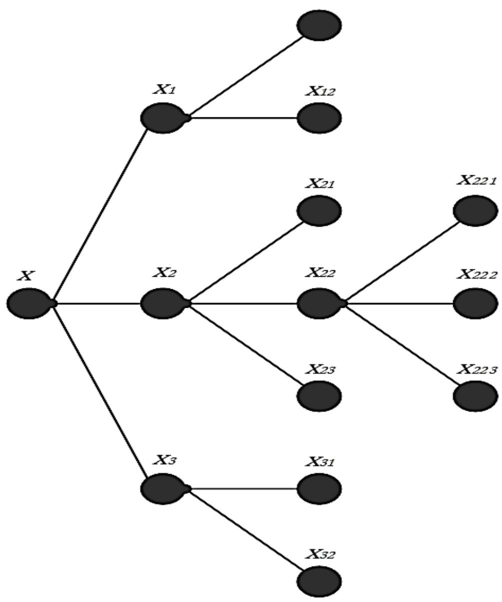

Fig. 1. The tree structure of the fuzzy signature

The structure of the fuzzy signature inertia adaptation function is described in Fig. 2 and was obtained using (1).

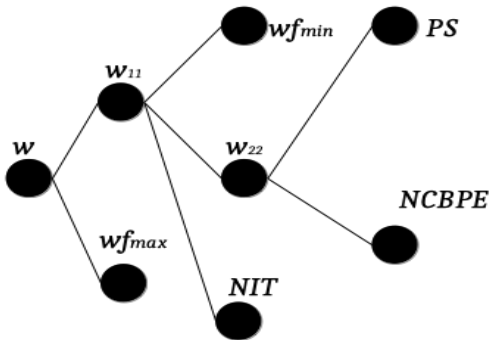

Fig. 2. The structure of the fuzzy signature as inertia weight adaptation function.

In this structure, three aggregation functions were used, which are max, min, and mean. The inertia weight w is the higher level of the structure, which was adapted according to the value of the feedback parameter. The adaptation of w was constrained with w_{fmin}, and w_{fmax}.

$$\boldsymbol{w} = \left[\begin{array}{c} \boldsymbol{w_{fmax}} \\ \left[\begin{array}{c} \boldsymbol{w_{fmin}} \\ \left[\begin{array}{c} \boldsymbol{NIT} \\ \left[\begin{array}{c} \boldsymbol{NCBPE} \\ \boldsymbol{PS} \end{array}\right] \end{array}\right] \end{array}\right] \end{array}\right] \tag{1}$$

Equation (1), the percentage of success (PS) is calculated using (2)

$$PS(t) = \frac{1}{N}\sum_{i=1}^{N} SC_i(t) \tag{2}$$

where SC_i is the count of particle which has the best position to minimize the objective function [16]. the successful count (SC_i) of particles is obtained in the following function:

$$SC_i(t) = \begin{cases} 1, & f(x_{id}(t)) < f(P_{id}(t)) \\ 0, & else \end{cases} \tag{3}$$

The NCBPE is the normalization of current best performance evaluation (CBPE) that measures the best fitness value by the most recent best candidate solution. As parameter feedback, CBPE is normalized to the following (4).

$$NCBPE = \frac{CBPE - CBPE_{min}}{CBPE_{max} - CBPE_{min}} \tag{4}$$

Then the NIT is the linear decreasing of iteration value that can be described in (5):

$$NIT = 1 - \left(\frac{it}{maxit}\right) \tag{5}$$

where the it is the current iteration number and $maxit$ is the maximum iteration value.

Consider that a_i is the aggregation function at i level of fuzzy signature structure, hence the process of equation (1) can be described in (6).

$$\begin{gathered} w = a_1(w_{fmax}; w_{12}) \\ w_{12} = a_2(w_{fmin}; w_{22}) \\ w_{22} = a_3(NIT; a_4(NCBPE, PS)) \end{gathered} \tag{6}$$

The aggregation function in the lowest level is a_4, where it is chosen as the mean function because of the $NCBPE$ and PS have the same behaviour that measure the performance of particle. Then according [17], in the end iteration the inertia value must be smaller to give better performance, hence in the next step of aggregation function is min function with the iteration factor NIT. The last two aggregation function is a_1 and a_2 is the min and max function to limit the inertia weight according of boundary w_{fmax} and w_{fmin}.

Neural Network's Training Using FSPSO

In order to evaluate the performance, the particle of PSO, in this paper Root Mean Square Error (RMSE) is used as evaluation function. Consider that y_t and y_n is the output target and the output of neural network from equation (2), hence the RMSE can be write in (8):

$$RMSE = \sqrt{\sum_{d=0}^{D}\sum_{n=0}^{N}(y_t - y_n)^2} \tag{7}$$

where d is the number of data and N is the number of output neuron. In transformer fault diagnosis the algorithm of conventional ANN or Various PSO in this paper is to minimize the MSE.

FSPSO-ANN Algorithm
start *initialize* c_1, c_2, *lower and upper bound;* *initialize* w_{fmin}, w_{fmax}, *and* w*;* *for i=1 to number of particles*   *initialize particle.*   *evaluate particle using equation (7);* $p_{i,d} = x_{i,d}$ *;*   *select* g_d *for minimize objective function;* *end for* *for it from 1 to maximum iteration*   *SC=0;*   *for i from 1 to number of particles*   *for d from 1 to D*

update velocity of particle using equation.
update position of particle using equation
evaluate particle using equation (7);
end for
end for
update P_{id} and G_d;
measure using equation (2), (4), and (5);
update inertia using (6);
end for
optimal solution = G_d ;
end

Fig. 3. Pseudo code of FSPSO-ANN

V. Result And Discussion

The performance of proposed FSPSO algorithm for ANN weight optimisation is investigated. The inertia weight of FSPSO-NN is updated using various measured parameters, such as Percentage Success (PS) of particle in each iteration, Best Performance Evaluation (NCBPE), and the linear decreasing iteration factor (NIT). The first experimental testing was done with various parameter of c_1 and c_2 in equation (7). The test using 10 input layers, 1 hidden layer with 10 neuron, and 1 output layer of NN architecture. The number of FSPSO particle is 20 particles, with 121 problem dimensions according the NN architecture. The range of inertia weighting of proposed FSPSO-NN such as w_{fmin} set into 0 and w_{fmax} set into 1 int the aggregation of equation (6) to ensure the optimum c_1 and c_2 parameters.

Table 1 presents the result of FPSO-NN when the parameters c_1 and c_2 are varied from 0.5 into 2. These parameters are crucial as the influence the behaviour and performance of the FPSO-NN. The goal was to determine the most suitable values for c_1 and c_2 to optimise network's performance. In this result, the suitable parameter of $c_1 = 1.5$ and $c_2 = 1.5$ or $c_2 = 1.2$. To further refine the optimization process, an additional test was conducted by varying of w_{fmin} which is the parameter of aggregation function of fuzzy signature. This aggregation function plays a significant role in combining the fuzzy rules and influences the final input of the fuzzy system. Table 2 displays the result from the test where each parameter configuration was run 10 times to ensure statistical reliability. The result indicates that the optimal parameter of FPSO-NN is $c_1 = 1.5$ and $c_2 = 1.5$ with the $W_{fmin} = 0.5$ and $W_{fmax} = 1$ or $c_1 = 1.5$ and $c_2 = 2$ with the $W_{fmin} = 0.1$ and $W_{fmax} = 1$. These configurations provided the best performance metrics, indicating a robust convergence to the global optimum. According to the study by [24], the c_2 is the g_d acceleration parameter of the PSO. This acceleration factor helps particles to converge more rapidly to the global best solution by adjusting their velocities based on cognitive and social components. The results in Table 3 is proved that if the c_2 set into 2 as the maximum, a lower value of W_{fmin} is required to ensure the particles efficiently converge to the global optimum. This finding aligns with the concept that higher acceleration factors can lead to faster convergence but may also require fine-tuning of other parameters to prevent premature convergence or oscillations around the optimal solution [24].

TABLE I

THE FSPSO-NN RESULT WITH VARIOUS c_1 AND c_2

Parameter		Mean RMSE
c1	**c2**	
1.5	0.5	0.089741
1.5	1.5	**0.087498**
1.5	2.0	**0.086840**
0.5	1.5	0.089013
2.0	1.5	0.087944
2.0	2.0	0.088651
0.5	0.5	0.089861

Fig. 4 illustrates the outcomes of various inertia configurations used in the optimization process. The Figure reveals that combining inertia adaptation with Neighbourhood-based Cooperative Binary Particle Excitation (NCBPE) and Particle Swarm (PS) techniques significantly accelerates particle convergence during the initial iterations. This indicates that the convergence towards an optimal solution occurs rapidly in the early stages of the algorithm's execution. The integration of a small weight of linear decrease weight adjustment in the final iterations, as observed in Fig. 4, implies a gradual reduction in inertia weight as the algorithm progresses. This linear decrease helps fine-tune the particle movement to ensure more precise convergence to the desired solution. This approach aligns with the findings by [16], who demonstrated that decreasing inertia weight linearly over time can significantly enhance the convergence speed and accuracy of PSO. Furthermore, the Figure indicates that incorporating NCBPE with the PS technique not only accelerates convergence in the initial stages but also maintains robust performance throughout the optimization process. NCBPE leverages the cooperative behaviour of particles within a neighbourhood, allowing them to share information and converge more efficiently.

Overall, Fig. 4 highlights the effectiveness of the proposed inertia configuration in optimizing the convergence speed and accuracy of the PSO process. The dynamic adjustment mechanism, which balances rapid convergence in the initial phases with meticulous refinement towards the end, proves to be a robust approach for achieving optimal solutions.

In the last test experiment, the FSPSO-NN compared with others optimization algorithm including Adaptive Inertia Weight PSO (AIWPSO) [17], Adaptive PSO (APSO) [19], Linear Decreasing Weight PSO (LDWPSO) [22], Genetic Algorithm [26] and the conventional PSO [17]. In Fig. 5 showed that the FSPSO outperform compared with others in early iteration followed by AIWPSO. The FSPSO take the advantage of inertia adaptation in the early iteration that comparable with the AIWPSO. The LDWPSO strategies showed the advantages of linear decreasing the inertia in the last iteration and the FSPSO too. The aggregation between adaptive and the linear decrease of FSPSO can jump from the early convergence that makes FSPSO slightly better than AIWPSO in the last iteration.

Table 3 shows the minimum, mean, maximum, and the standard deviation of 10 times run of training cost and testing on neural network electrical load prediction with the data test. The table compares the performance of various algorithms in training and testing a Neural Network (NN). The algorithms compared are FSPSO-NN, AIWPSO-NN, APSO-NN, LDWPSO-NN, PSO-NN, and GA-NN. Performance metrics include the Root Mean Square Error (RMSE) for training and testing, with statistics on the minimum, mean, maximum, and standard deviation (STD) values. The FSPSO give the better performance of training step on neural network. It proofed from the minimum and mean cost of neural network training step. The AIWPSO-NN give better in testing step with the lowest of minimum RMSE. However, the FSPSO contributed robust performance in the testing with the lowest of RMSE mean. This result makes the prediction accuracy of the electrical load improve than the conventional neural network.

The detailed analysis of the table reveals nuanced insights into the comparative performance of various algorithms for training artificial neural networks (ANNs) in the specific context of electrical load forecasting. Notably, the FSPSO-NN algorithm stands out as the most promising candidate across both training and testing/deployment phases. During the training phase, FSPSO-NN demonstrates superior performance by consistently achieving the lowest minimum and mean RMSE values. This suggests its effectiveness in minimizing errors and optimizing the training process for enhanced accuracy. Furthermore, the algorithm exhibits impressive stability, as indicated by its minimal standard deviation of RMSE, highlighting its robustness across diverse training scenarios. In the testing/deployment phase, FSPSO-NN maintains its superiority with the lowest minimum and mean RMSE values, indicating its ability to generalize well to unseen data and maintain high accuracy levels in real-world applications. Its low standard deviation further underscores its reliability and consistency in performance, reinforcing its status as the most effective algorithm for electrical load forecasting tasks.

In addition,

Fig. 6 shows the result of electrical load prediction comparison between FSPSO-NN, AIWPSO-NN, and the conventional Neural Network. The Figure appears to show the performance of different algorithms in predicting electricity load over iterations. There are four curves on the graph: real data (black line), ANN Prediction (red line), AIWPSO-NN Prediction (blue line), and FSPSO-NN Prediction (green line). FSPSO-NN consistently tracks the real data more closely than the other algorithms, indicating superior prediction performance. It proved that with the Fuzzy Signature strategies to update the PSO parameters can improve the neural network prediction accuracy. This suggests that while alternative approaches may yield satisfactory results under certain conditions, they may struggle to maintain consistent performance across different scenarios or fail to generalize effectively to unseen data. The overall analysis suggests that FSPSO-NN is the most accurate and reliable algorithm for predicting electricity load in this dataset. The findings highlight the effectiveness of the FSPSO-NN approach as a powerful technique for enhancing ANN training, particularly in the field of electrical load forecasting. FSPSO-NN optimizes the training process by selecting the most relevant features and improving the network's ability to accurately predict electrical loads.

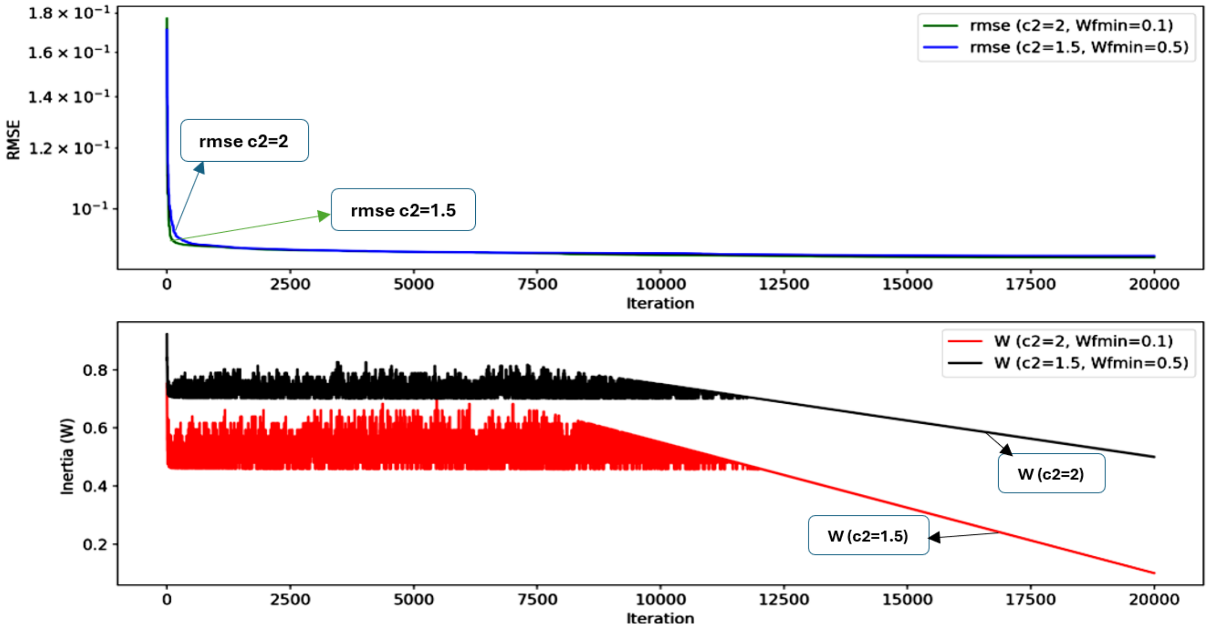

Fig. 4. Comparison result of FSPSO with difference W configuration

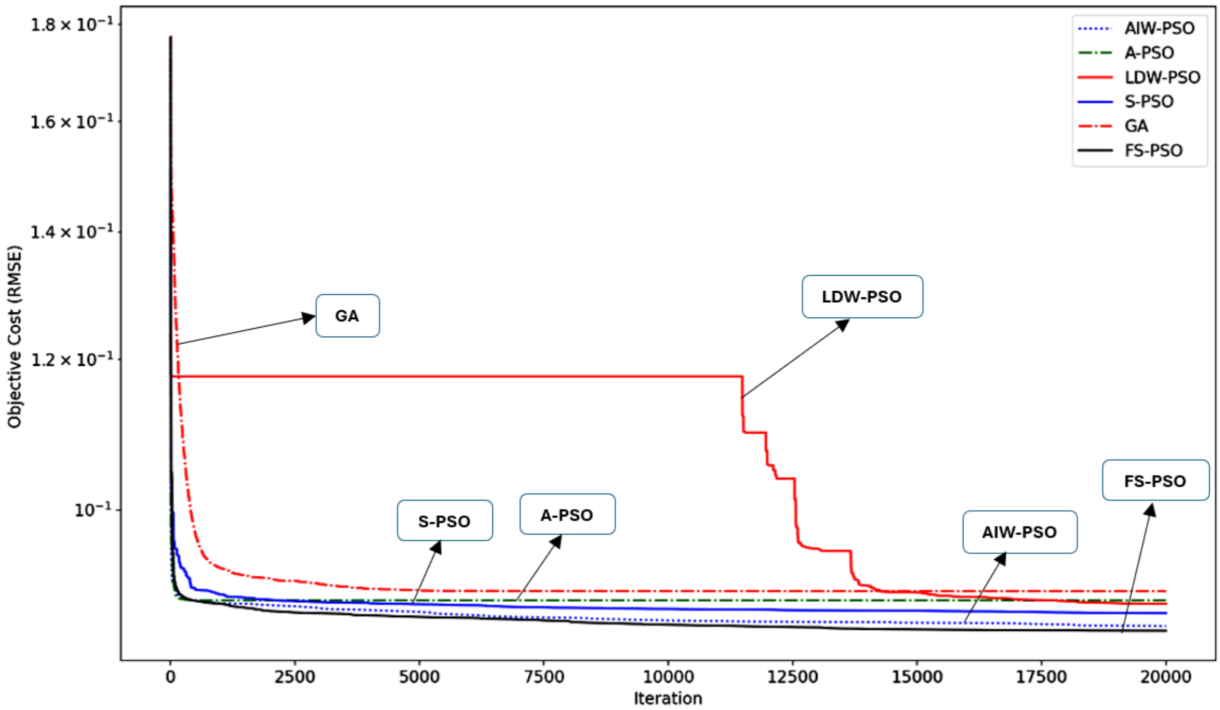

Fig. 5. Comparison of FSPSO with others optimization algorithm

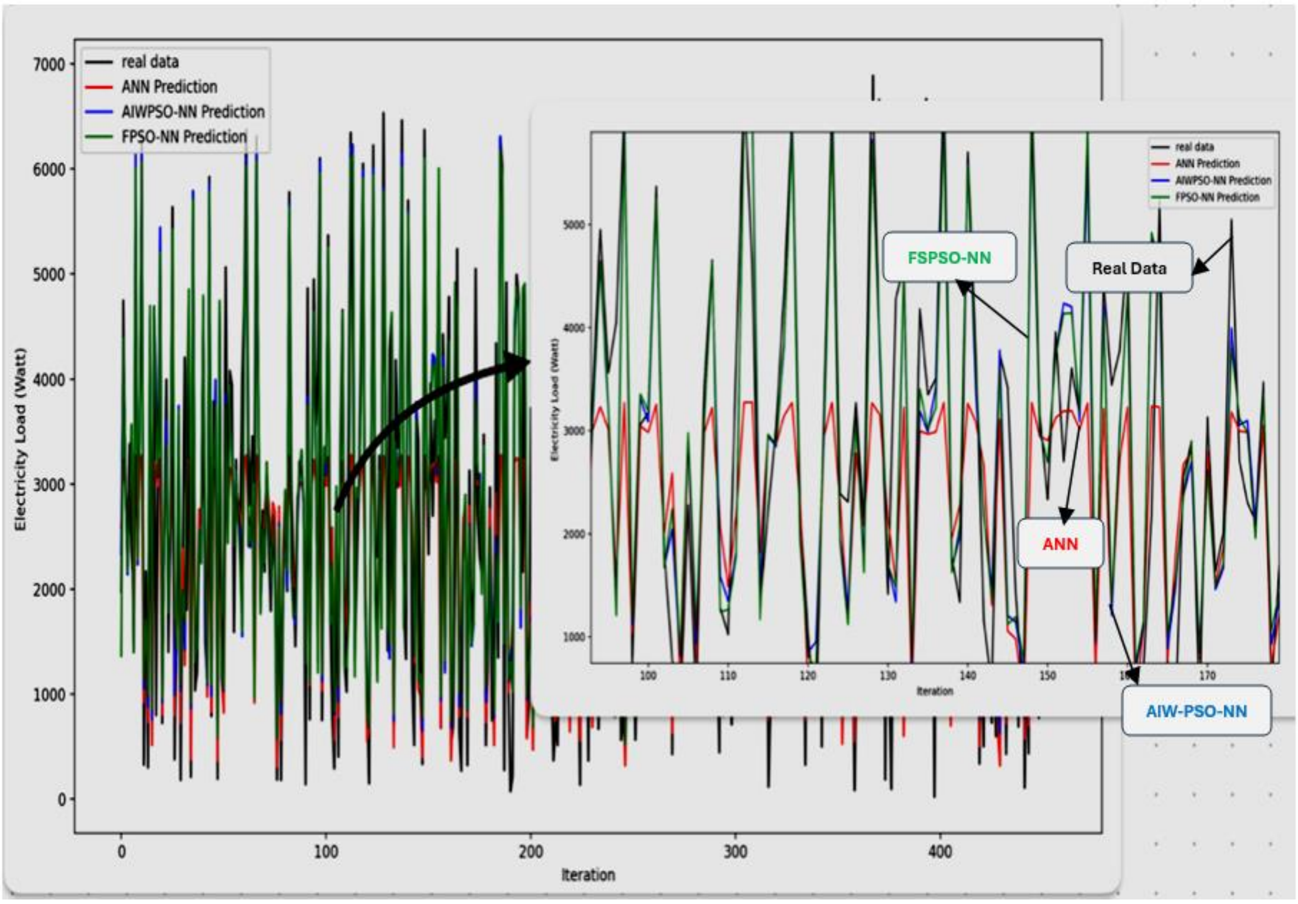

Fig. 6. Prediction Comparison Result of FSPSO-NN with Conventional ANN and AIW-PSO-NN

TABLE 2
THE FSPSO-NN RESULT WITH VARIOUS W_{fmin} AND W_{fmax} PARAMETERS.

c1	c2	W		RMSE		
		Min	Max	Min	Mean	Max
1.5	1.5	0.1	1	0.087519	0.087828	0.088195
		0.2	1	0.087525	0.087572	0.087632
		0.3	1	0.087034	0.087309	0.087930
		0.4	1	0.086970	0.087283	0.087615
		0.5	1	**0.086804**	**0.086997**	0.087209
		0.6	1	0.087698	0.087984	0.088440
1.5	2.0	0.1	1	**0.086402**	**0.086826**	0.087017
		0.2	1	0.086668	0.086940	0.087232
		0.3	1	0.087069	0.087138	0.087300
		0.4	1	0.087280	0.087555	0.087698
		0.5	1	0.087800	0.088254	0.088897

TABLE 3
COMPARISON OF FSPSO RESULTS WITH FIVE OTHERS IN TRAINING NEURAL NETWORK AND TESTING

Algorithm		FSPSO-NN	AIWPSO-NN	APSO-NN	LDWPSO-NN	PSO-NN	GA-NN
Training Cost RMSE	Min	**0.08640**	0.08689	0.08964	0.08926	0.08826	0.09063
	Mean	**0.08683**	0.08731	0.09002	0.09014	0.08875	0.09072
	Max	**0.08702**	0.08812	0.09066	0.09094	0.08916	0.09104
	STD	0.00025	0.00037	0.00030	0.00046	0.00029	0.00012
(Test/ Deploy)	Min	0.08817	**0.08732**	0.08923	0.08956	0.08879	0.09090
	Mean	**0.08807**	0.08838	0.08977	0.09052	0.08899	0.09097
	Max	**0.08934**	0.09231	0.09036	0.09166	0.08986	0.09112
	STD	0.00080	0.00179	0.00045	0.00057	0.00065	0.00008

VI. CONCLUSION

The paper presents a unique methodology for training artificial neural networks (ANNs) geared towards electrical load forecasting, employing Particle Swarm Optimization (PSO) with a distinctive inertia weight update mechanism termed Fuzzy Signature. It utilizes real-world data acquired through an Internet of Things (IoT) data collection system, demonstrating its practical applicability. In the research, the performance of the proposed algorithm is thoroughly assessed. This involves comparisons with other inertia weight adjustment strategies commonly utilized in PSO, along with traditional ANN training techniques. The experimental findings indicate that the proposed algorithm offers notable improvements in load forecasting accuracy. Specifically, it efficiently reduces the root mean square error (RMSE) during the training phase, suggesting superior prediction accuracy compared to conventional methods.

This research contributes significantly to the domain of load forecasting by introducing a novel optimization approach tailored to the specific requirements of this application. The integration of real-world IoT data enhances the relevance and applicability of the proposed algorithm, making it a valuable contribution to both academia and industry. Expanding on the promising outcomes of the study, future research directions could explore the development of hybrid models that amalgamate the proposed PSO-based approach with other optimization techniques or forecasting methods. By integrating complementary strategies, such as genetic algorithms, simulated annealing, or machine learning algorithms like support vector machines or deep learning, researchers can potentially achieve even greater accuracy in load forecasting.

According to the performance results, FSPSO-NN is the most consistently performing algorithm, especially in the training phase, with competitive performance in testing. AIWPSO-NN shows strong testing performance but with higher variability. Moreover, extending the application of the enhanced model to short-term electricity load forecasting represents a promising avenue for investigation. Short-term forecasting plays a crucial role in optimizing energy resource management and decision-making processes in diverse sectors, including energy production, distribution, and consumption. By accurately predicting near-future load demands, stakeholders can make informed decisions regarding resource allocation, energy pricing, and infrastructure planning, ultimately leading to more efficient and sustainable energy systems.

Furthermore, future research efforts could focus on addressing challenges related to data quality, model interpretability, and scalability. Enhancements in data preprocessing techniques, feature selection methods, and model validation approaches can contribute to improving the robustness and reliability of load forecasting models. Additionally, exploring the interpretability of hybrid models can facilitate better understanding and trust in the generated predictions, enhancing their adoption and practical utility.

REFERENCES

[1] B. Toual, L. Mokrani, A. Kouzou, and M. Machmoum, “Control And Management Of A Solar-Wind Hybrid System For Power Quality Improvement,” J. Eng. Sci. Technol., vol. 13, no. 6, pp. 1420–1439, 2018.

[2] M. Ahmad and M. Zeeshan, “Multi-objective optimization of concentrated solar power plants from an energy-water-environment nexus perspective under distinct climatic conditions–Part B: Environ-economic analysis,” J Clean Prod, vol. 385, p. 135689, 2023.

[3] P. Das, J. Mathur, R. Bhakar, and A. Kanudia, “Implications of short-term renewable energy resource intermittency in long-term power system planning,” Energy strategy reviews, vol. 22, pp. 1–15, 2018.

[4] S. Chapaloglou et al., “Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island’s power system,” Appl Energy, vol. 238, pp. 627–642, 2019.

[5] S. Haben, S. Arora, G. Giasemidis, M. Voss, and D. V. Greetham, “Review of low voltage load forecasting: Methods, applications, and recommendations,” Appl Energy, vol. 304, p. 117798, 2021.

[6] M. Irfan, F. Ramlie, Widianto, M. Lestandy, and A. Faruq, “Prediction of Residential Building Energy Efficiency Performance using Deep Neural Network,” IAENG Int. J. Comput. Sci., vol. 48, no. 3, pp. 731 – 737, 2021.

[7] M. Khalil, A. S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, “Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption—A systematic review,” Eng Appl Artif Intell, vol. 115, p. 105287, 2022.

[8] N. K. Rai, D. Saravanan, L. Kumar, P. Shukla, and R. N. Shaw, “RMSE and MAPE analysis for short-term solar irradiance, solar energy, and load forecasting using a Recurrent Artificial Neural Network,” in Applications of AI and IOT in Renewable Energy, Elsevier, 2022, pp. 181–192.

[9] J. Vaish, S. S. Datta, and K. Seethalekshmi, “Short term load forecasting using ANN and ensemble models considering solar irradiance,” in 2020 International Conference on Electrical and Electronics Engineering (ICE3), 2020, pp. 44–48.

[10] N. Setyawan, N. Mardiyah, K. Hidayat, and Z. Has, “Object Detection of Omnidirectional Vision Using PSO-Neural Network for Soccer Robot,” in 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2018, pp. 117–121.

[11] N. Setyawan, M. Nasar, and N. Mardiyah, “Jaya-Neural Network for Server Room Temperature Forecasting Through Sensor Network,” in 2019 International Electronics Symposium (IES), 2019, pp. 428–431.

[12] H. A. Illias, X. R. Chai, and A. H. Abu Bakar, “Hybrid modified evolutionary particle swarm optimisation-time varying acceleration coefficient-artificial neural network for power transformer fault diagnosis,” Measurement (Lond), vol. 90, pp. 94–102, 2016, doi: 10.1016/j.measurement.2016.04.052.

[13] M. S. Nobile, P. Cazzaniga, D. Besozzi, R. Colombo, G. Mauri, and G. Pasi, “Fuzzy Self-Tuning PSO: A settings-free algorithm for global optimization,” Swarm Evol Comput, vol. 39, pp. 70–85, 2018.

[14] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, “Inertia weight strategies in particle swarm optimization,” Proceedings of the 2011 3rd World Congress on Nature and Biologically Inspired Computing, NaBIC 2011, pp. 633–640, 2011.

[15] J. K. Kordestani, A. Rezvanian, and M. R. Meybodi, “An efficient oscillating inertia weight of particle swarm optimisation for tracking optima in dynamic environments,” Journal of Experimental and Theoretical Artificial Intelligence, vol. 28, no. 1–2, pp. 137–149, 2016.

[16] Y. Tuppadung and W. Kurutach, “Comparing nonlinear inertia weights and constriction factors in particle swarm optimization,” in International Journal of Knowledge-Based and Intelligent Engineering Systems, vol. 15, no. 2, pp. 65–70, 2000.

[17] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, “A novel particle swarm optimization algorithm with adaptive inertia weight,” Applied Soft Computing Journal, vol. 11, no. 4, pp. 3658–3670, 2011.

[18] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999, vol. 3, pp. 1945–1950, 1999.

[19] A. Djoewahir, K. Tanaka, and S. Nakashima, “Adaptive PSO-based self-tuning PID controller for ultrasonic motor,” International Journal of Innovative Computing, Information and Control, vol. 9, no. 10, pp. 3903–3914, 2013.

[20] A. Komarudin, N. Setyawan, L. Kamajaya, M. N. Achmadiah, and Zulfatman, “Signature PSO: A novel inertia weight adjustment using fuzzy signature for LQR tuning,” Bulletin of Electrical Engineering and Informatics, vol. 10, no. 1, pp. 308–318, 2021.

[21] R. Eberhart and J. Kennedy, “New optimizer using particle swarm theory,” Proceedings of the International Symposium on Micro Machine and Human Science, pp. 39–43, 1995.

[22] R. C. E. Y.H. Shi, “A Modified Particle Swarm Optimizer,” IEEE International Conference on Evolutionary Computation, vol. 49, no. 5, pp. 1144–1153, 1998.

[23] L. T. Kóczy, T. Vámos, and G. Biró, “Fuzzy signatures,” Proceedings of Eurofuse-SIC, vol. 99, pp. 25–28, 1999.

[24] Á. Ballagi and L. T. Kóczy, “Fuzzy signature based mobil robot motion control system,” SAMI 2008 6th International Symposium on Applied Machine Intelligence and Informatics - Proceedings, pp. 29–33, 2008.

[25] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, “Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients,” IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240–255, 2004.

[26] F. Yu and X. Xu, “A short-term load forecasting model of natural gas based on optimized genetic algorithm and improved BP neural network,” Appl Energy, vol. 134, pp. 102–113, 2014.