sensors

MDPI

Article

Hybrid Particle Swarm Optimization for Multi-Sensor Data Fusion

Hyunseok Kim [1] **and Dongjun Suh** [2,*]

[1] IoT Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon 34129, Korea; hertzkim@etri.re.kr

[2] School of Convergence & Fusion System Engineering, Kyungpook National University, Sangju 37224, Korea

* Correspondence: dongjunsuh@knu.ac.kr; Tel.: +82-54-530-1482

Received: 11 July 2018; Accepted: 22 August 2018; Published: 24 August 2018

Abstract: A hybrid particle swarm optimization (PSO), able to overcome the large-scale nonlinearity or heavily correlation in the data fusion model of multiple sensing information, is proposed in this paper. In recent smart convergence technology, multiple similar and/or dissimilar sensors are widely used to support precisely sensing information from different perspectives, and these are integrated with data fusion algorithms to get synergistic effects. However, the construction of the data fusion model is not trivial because of difficulties to meet under the restricted conditions of a multi-sensor system such as its limited options for deploying sensors and nonlinear characteristics, or correlation errors of multiple sensors. This paper presents a hybrid PSO to facilitate the construction of robust data fusion model based on neural network while ensuring the balance between exploration and exploitation. The performance of the proposed model was evaluated by benchmarks composed of representative datasets. The well-optimized data fusion model is expected to provide an enhancement in the synergistic accuracy.

Keywords: multi-sensor system; multi-sensor information fusion; particle swarm optimization; sensor data fusion algorithm; distributed intelligence system

1. Introduction

A moment of evolution is now emerging toward a new paradigm known as smart convergence, which is bringing together both heterogeneous and information communication technologies. These emerging phenomena have prompted researchers to explore new possibilities for sophisticated smart devices [1] to be embedded in various real objects and to cope with various environmental changes. Recently, multiple similar and/or dissimilar sensors, as shown in Figure 1, have been widely used to provide precise sensing information from different viewpoints [2,3] and to realize the Internet of Things in a cyber-physical system [4]. Given that the accuracy of a sensor system is dictated by the degree to which repeated measurements under unchanging conditions are able to produce the same results, a multi-sensor system has typically been thought of as a way to guarantee the accuracy of a measurement system [5]. Hence, multi-sensor systems are an emerging research topic that is becoming increasingly important in various environmental perception activities. Nevertheless, challenging problems of multisensory data fusion algorithms are still far from accomplished [6]. Evolutionary computation methods [7] recently seem to be making a comeback in order to solve real-world problems concerning typically not iid (independent and identically distributed) data or sparse labeled data, and these methods are expected to help such a fusion model enhanced.

Because a single sensor usually only recognizes a limited set of partial information about the environment, multiple similar and/or dissimilar sensors are needed to provide accurate sensing information from a variety of perspective in an integrated manner [8]. Such multiple-sensing information is combined depending on data fusion algorithms to achieve synergistic effects. However, the construction of a data fusion model is not trivial because of difficulties in meeting the restricted conditions of a multi-sensor system such as its limited options for deploying sensors and nonlinear characteristics or correlation errors of multiple sensors. Such a nonlinearity optimization problem in a data fusion model can be solved by a neural network algorithm with an effective back propagation method ensuring the best performance of the network [9]. In recent multi-sensor fusion research [10,11], neural networks (including deep neural networks) play a major role in feature classification and decision making. However, difficulties for the efficient and high accurate multisensory fusion model remain.

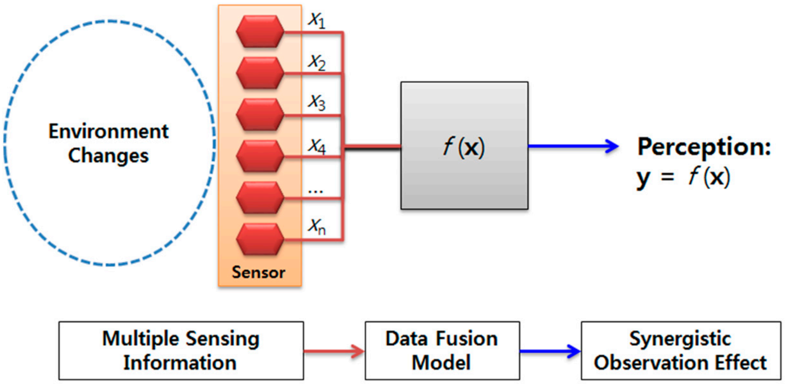


Figure 1. A multi-sensor system.

As is well known, maintaining the balance between the exploration of new possibilities and the exploitation of old certainties [12,13] has been considered a priority in designing an optimization scheme. As one of latest evolutionary approaches, particle swarm optimization (PSO) uses randomly placed particles on the search-space to explore new possibilities toward a global best position as a new solution to solve real-world problems [14–16]. PSO can be utilized in training a neural network by a population based stochastic back-propagation technique [17]. The randomly placed particles are more likely to find the global minimum than neural networks using a single particle. Venu et al. [18] showed that the parameters of feed-forward neural network converge faster using PSO than any other algorithm based on back-propagation methods, e.g., stochastic gradient descent, scaled gradient descent, or Levenberg-Marquardt (LM) [19]. They adopted PSO as a training algorithm involving adjusting the parameters (i.e., weights and biases) to optimize performance of the neural network. Kim et al. [20] suggested the PSO proportional-integral-derivative (PSOpid) which is one of the enhanced PSO algorithms through the stabilization of particle movement. Although a particle can be used as one of the solutions to regulate the parameters of neural network, it is necessary to constrain the range of search space for quicker convergence and higher fitness in PSO.

This paper proposes a hybrid PSO capable of overcoming large-scale nonlinearity or heavy correlation in the sensing data fusion and facilitates the construction of a robust data fusion model while maintaining the balance between the exploration of new possibilities and the exploitation of old certainties. The proposed algorithm was evaluated by benchmarks composed of representative datasets and the well-optimized data fusion model is expected to provide an enhancement in synergistic accuracy. In this paper, the neural network is used as a basic model and different backpropagation methods such as PSO, LM, and PSOpid are considered, thus the expression of neural network is simply omitted.

Section 2 discusses the problem of data fusion model and previous methods such as ordinary LM and PSO. We then propose a hybrid PSO, LM, and PSOpid in Section 3, and Section 4 describes how evaluation is performed for the proposed approach with the different weighing participants in diverse user-scenarios and with the different exemplary datasets from MATLAB. Section 5 concludes the research and describe possible future works.

2. Data Fusion Model Using Neural Network and PSO

Compared to a single sensor system, multiple sensor systems have the advantage of broadening the sensing range and improving the perception of environmental conditions [8]. In addition, a multi-sensor system allows information from a set of homogeneous sensors to be combined using a data fusion model. Thus, a multi-sensor system represents a proven method for enhancing the accuracy of a measurement system.

Figure 2 shows the case of using multiple Force Sensitive Resistor (FSR) sensors on a smart floor block [16]. FSRs is one of best solutions to meet multi-dimensional requirements such as maintaining visibility; holding weights of standing, moving, or jumping user; avoiding occlusion in sensing area; and reducing production cost. In this paper, an FSR is composed of two substrate layers with a conductive core, and, when pressure is applied to the FSR, the substrate moves, compressing the conductive core to detect the weight on the smart floor. The pressure changes are measured by the analog output from the FSR resistive divider. The weight is dispersed to the corner based on the approximate inverse power law which is the relation between distance and force of the FSR [21].

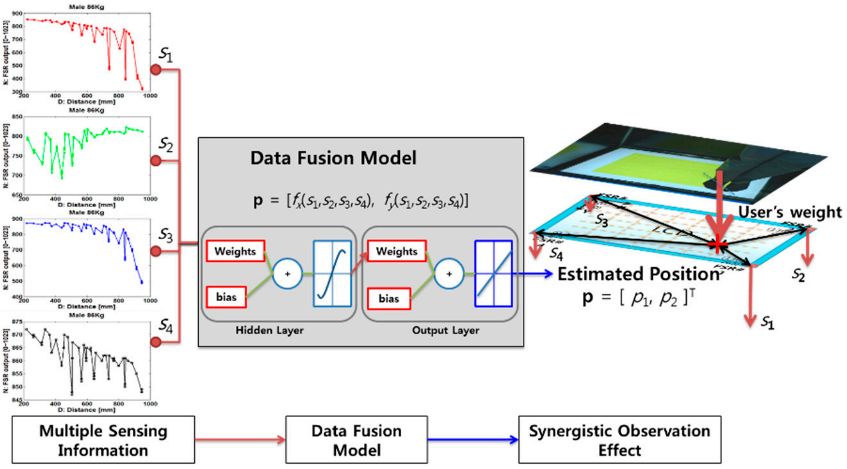


Figure 2. Example of user localization using a multi-sensor system.

In [16], a neural network adopting three layers is used, and we can clarify our notation and describe it as follows. Each four-dimensional vector is applied to the input layer as $\mathbf{s} = [\ s_1 \ \ s_2 \ \ s_3 \ \ s_4\ ]^T$, where $\{s \in \mathbb{R} : 0 \leq s \leq 2^{10}\}$. An output produced by a non-linear activation function at each hidden unit, $g(net)$, that is, a hyperbolic tangent sigmoid as,

$$g(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{1}$$

where the net is the inner product of inputs with weights at each neural network unit.

Each output of the neural network as $\mathbf{p} = [p_1\ p_2]^T$ calculates the activation function based on the hidden units described in Equation (2):

$$p_k(\mathbf{s}) \equiv \sum_{j=1}^{n_H} \theta_{kj} g\left(\sum_{i=1}^{d} \theta_{ji} s_i + \theta_{j0} \right) + \theta_{k0} \tag{2}$$

where the subscript i indexes units in the input layer and j indexes units in the hidden layer; θ_{ji} denotes the input-to-hidden layer parameters at the hidden unit j; and the dimension of input vector is d.

In addition, the subscript k indexes units in the output layer, n_H indicates the number of hidden units and θ_{kj} denotes the hidden-to-output layer weights. θ_{j0} and θ_{k0} are mathematically treated as the bias of the layer. We refer to its output, $\mathbf{p} = [p_1\ p_2]^T$, as the user's estimated 2D position (x, y) on the screen, respectively. The authors in [16] regarded the cost function as the sum of the output units of the squared difference between the target t_k and the actual output a_k.

$$J(\theta) \equiv \frac{1}{2m}\sum_{i=1}^{m}\sum_{k=1}^{c}(t_k - a_k)^2 = \frac{1}{2m}\sum_{i=1}^{m}\parallel \mathbf{t} - \mathbf{a} \parallel^2 \tag{3}$$

where **t** and **a** are the target and actual output vectors of length c (e.g., c is 2 in this experiment), the batch size is m and θ represents all weights and biases.

FSRs show nonlinear issues or problems of correlation errors. It can be caused by instability of the glass or metal frame, response time of the sensor, errors in manufacturing, assembly processes, the sensor placement, and human errors. Therefore, Kim et al. [16] suggested a PSO-based neural network model, namely PSO, which can be used as alternatives method to the gradient-descent algorithm by randomly spreading multiple particles capable of finding each optimum and being converged toward the global optima. However, if a system needs many parameters that should be adjusted for many epochs and other potential parameters, then the system needs to be improved.

3. A Hybrid PSO Model for Multi-Sensor Data Fusion

Since March [12] proposed the balancing between exploration and exploitation in learning, it has been extensively researched and widely applied to various domains [13]. In this paper, the PSO first explores possibilities as a global search, the LM then exploits certainties as a local search, and the PSOpid suggested in [20] lastly explores a new possibility within the range-optimized search space. The PSOpid is one of the enhanced PSO algorithms and each particle finds the global optimum securely while preventing a particle from becoming unstable or exploding. In addition, the algorithm can converge more quickly and get high fitness values compare to other algorithms in the range-optimized search space.

3.1. Improved Exploitation of Neural Network Using Ordinary PSO

The idea of PSO allows particles randomly placed on the search-space to explore new possibilities to the best global position. PSO is used to train a neural network with back propagation method considering a population based stochastic optimization technique [17]. On the other hand, the randomly placed particles will have a high probability of discovering the global minimum in comparison with the ordinary LM using a single particle. In this method, PSO can be alternatively used as the training algorithm at each iteration n, including parameter tuning to optimize the performance of the neural network. In addition, a particle is used to determine the value of parameters in the neural network. The magnitude of vectors θ_i and $\mathbf{v}_i$ are equivalent to the dimension of the weights and biases. The velocity and position of the i-th particle after the n-th iteration are shown in Equations (4) and (5).

$$\mathbf{v}_i[n] = K\left\{\begin{array}{c} \omega\cdot\mathbf{v}_i[n-1] \\ +c_1\cdot rand_1\cdot(\mathrm{p}_{i_{\mathrm{best}}}[n-1] - \theta_i[n-1] \\ +c_1\cdot rand_2\cdot(\mathrm{g}_{best}[n-1] - \theta_i[n-1] \end{array}\right\} \tag{4}$$

$$\theta_i[n] = \theta_i[n-1] + \mathbf{v}_i[n] \tag{5}$$

The previous best position is selected using Equation (6), and the global best position is decided by Equation (7).

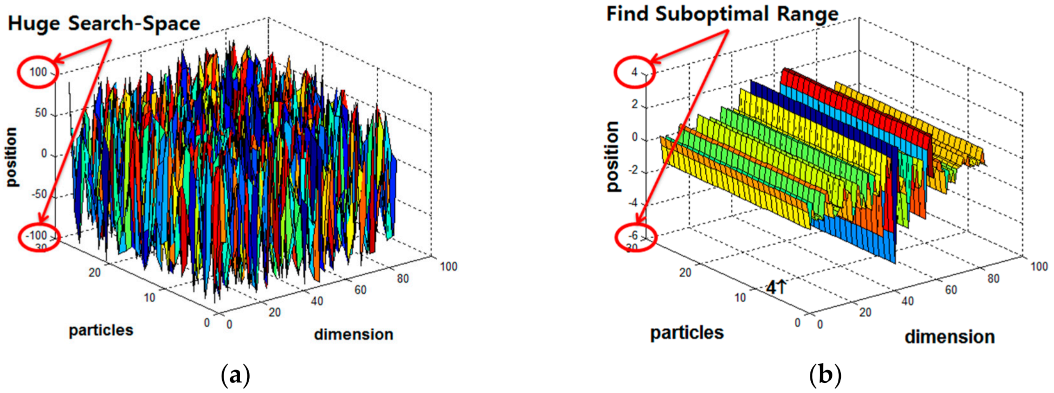
$$\mathrm{p}_{i_{\mathrm{best}}}[n] := \left\{\begin{array}{c} \theta_i \text{ if } \mathrm{J}(\theta_i[n]) \le \mathrm{J}(\theta_i[n-1]) \\ \mathrm{p}_{i_{\mathrm{best}}}[n-1] \text{ otherwise} \end{array}\right. \tag{6}$$

$$g_{\text{best}}[n] = \{\, p_{j_{\text{best}}}[n] \,\big|\, \forall i : J(p_{j_{\text{best}}}[n]) \le J(p_{i_{\text{best}}}[n]) \,\} \tag{7}$$

where the number of particles is Z_k ($\approx 10 + 2 \times$ (the dimension of the weights and biases)$^{0.5}$) and the inertia weight is ω, which dynamically adjusts the velocity. Moreover, the cognitive component c_1 and the social component c_2 change the particles' velocity toward the previous best and global best position, respectively. The PSO uses random numbers determined from a uniform distribution $rand_1$ and $rand_2$ to avoid unfortunate states in which all particles quickly settle into an unchanging direction. Consequently, the parameters are updated in accordance with the found global best position.

This paper proposes a hybrid LM and PSO scheme, namely LM-PSO, based on the concept of two-phase evolutionary programming [22]: First, PSO is used to explore a new possibility for overcoming multiple minima, and second, the initial value of the weights and biases of LM-based neural network, namely LM, is set—in other words, the neural network is trained—to the optimal configuration derived in the PSO phase. Consequently, the proposed LM-PSO can derive more accurately a globally optimal solution compared to LM.

In the first stage, normally a PSO has the initial parameters shown in Table 1 for a multiple FSR system. As illustrated in Figure 3, PSO finds suboptimal range of parameters which can reduce the configuration space for LM's parameters.



(**a**) (**b**)

Figure 3. PSO alone: (**a**) Initial position of the particles within a hypercube using a uniform random distribution; and (**b**) converged position of the particles.

Table 1. Parameter information of the PSO method alone.

Parameters	Value
Swarm size (Z_k)	28
Initial Position of Particles	Spread within a hypercube using a uniform random distribution
Minimum velocity norm	0.05
Inertial weight (ω)	1
Minimum position (min_pos)	−100
Maximum position (max_pos)	100

In the second stage, the proposed LM-PSO is trained with the optimal initial weight and bias values derived during the PSO stage. Although PSO is implausible on its own as a solution for the convergence rate and solution accuracy as illustrated in Figure 3, the proposed hybrid LM-PSO shows an enhanced performance of both accuracy and convergence rate as compared to conventional LM. However, hope remains for new possibilities for deriving an ultimate global optimum.

3.2. Exploration Toward Ultimate Goals for the Use of Enhanced PSO

Ordinary PSO alone has difficulty negotiating the tradeoff between global and local search because particles are initially deployed following a uniform random distribution in a hypercube big enough to contain a prospective global optimum. When the particle position range is not limited within the minimum and maximum positions or is too broad, the swarm can become unstable or explode, resulting in a slow convergence. Therefore, a novel approach is necessary for each particle to discover the global optimal solution securely, preventing the particles from becoming unstable or exploding.

The proportional-integral-derivative (PID) is widely applied in a feedback control loop technology [23]. The major advantages of the PID are easy to be implemented as well as only three parameters, i.e., proportional, integral and derivative terms, are required to be adjusted. The proportional gain is subject to the current error, the integral gain varies in proportion to both the magnitude and the duration error, and the derivative gain represents a prediction of future errors [24]. In [20], Kim et al. proposed a new approach using PSOpid as described in Equation (8) to change the particle's position of PSO with less oscillation.

$$\theta_i[\mathrm{n}] = \theta_i[\mathrm{n}-1] + \left\{ \begin{array}{c} K_P \cdot \mathbf{v}_i[\mathrm{n}] \\ +K_I \cdot \sum_{\mathrm{k}=0}^{\mathrm{n}} \mathbf{v}_i[\mathrm{k}] \\ +K_D \cdot (\mathbf{v}_i[\mathrm{n}] - \mathbf{v}_i[\mathrm{n}-1]) \end{array} \right\} \tag{8}$$

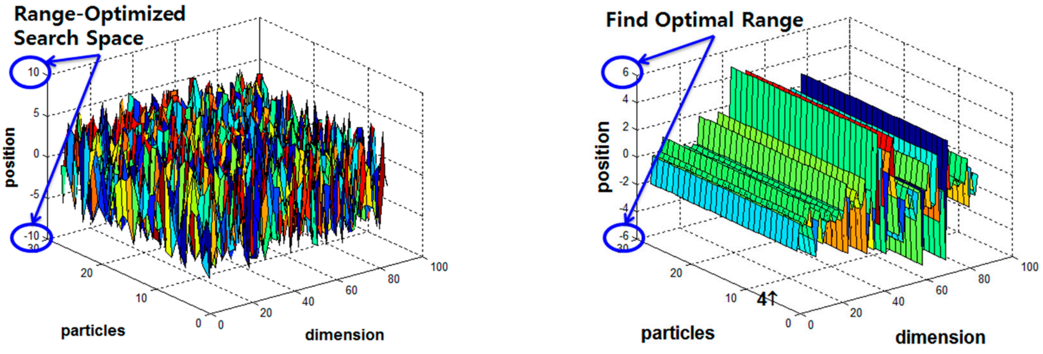
where K_P is the proportional term, K_I is integral term, and K_D is the derivative term, and they are selected through trial and error operations, as shown in Table 2.

Table 2. Parameter information of the PSOpid-based method alone.

Parameters	Value
Minimum position (min_pos)	1.2 × min (LM-PSO)
Maximum position (max_pos)	1.2 × max (LM-PSO)
Proportional term (K_P)	0.5 (fixed)
Integral term (K_I)	0.5 (fixed)
Derivative term (K_D)	0.6 (fixed)

Adjustment of the velocity is calculated from the acceleration with respect to the distance error produced by both present and its previous best position compared to the global best solution. Furthermore, two random, uniformly distributed variables are utilized in preserving the diversity. The position of each particle is updated according to the multiplying the velocity with the three PID terms, which are described in Table 2. In addition, it is necessary to restrain the dynamic range of each particle position to prevent a swarm from exploding or unstable conditions.

The position of each particle is initialized to the range between maximum and minimum derived from the output of the precedent LM-PSO technique. The initial global best position is optimally configured in accordance with the LM-PSO method. Consequently, particles randomly distributed can better guarantee an accurate global optimal solution than they can through either method alone, as shown in Figure 4.



(**a**) (**b**)

Figure 4. A hybridization of PSOpid, LM and PSO, namely PSOpid-LM-PSO: (**a**) initial position of particles within a shrunk hypercube; and (**b**) convergence position of the particles.

The stabilized PSOpid, in which each particle is evaluated by using a fitness function and update, is located in an enhanced location to discover the best optimal solution without local minimum. The convergence rate of the proposed PSOpid is faster and it has much higher performance than previous methods. This technique enables efficient implementation because of its small number of parameters, which both shortens the training time and reduces overfitting compared to ordinary PSO.

3.3. Three-Phase Hybrid PSO Method Balancing between Exploration and Exploitation

The proposed PSOpid-LM-PSO, as shown in Figure 5, assigns each particle a position calculated based on the output of the LM-PSO phase to initialize an optimal configuration of a global best position. The swarm intelligence technology, in which each individual evaluates, compares, and imitates one another, is a better way to find the best optimum with no local minima. Importantly, the dynamic range of each particle position is limited to prevent a swarm from becoming unstable or exploding. In the three-phase hybrid optimization method, PSO-based exploration of new possibilities is firstly executed for the configuration space of parameters in the multi-sensor data fusion model. Secondly, ordinary LM configured with the sub-optimized range can focus on the exploitation of old certainties. Finally, PSOpid can explore the ultimate goal with more accuracy and faster convergence rate. As a result of the hybrid method, the time needed for convergence is also shorter than that of an ordinary method. Thus, the number of iterations required for convergence counts in comparing speed of convergence.

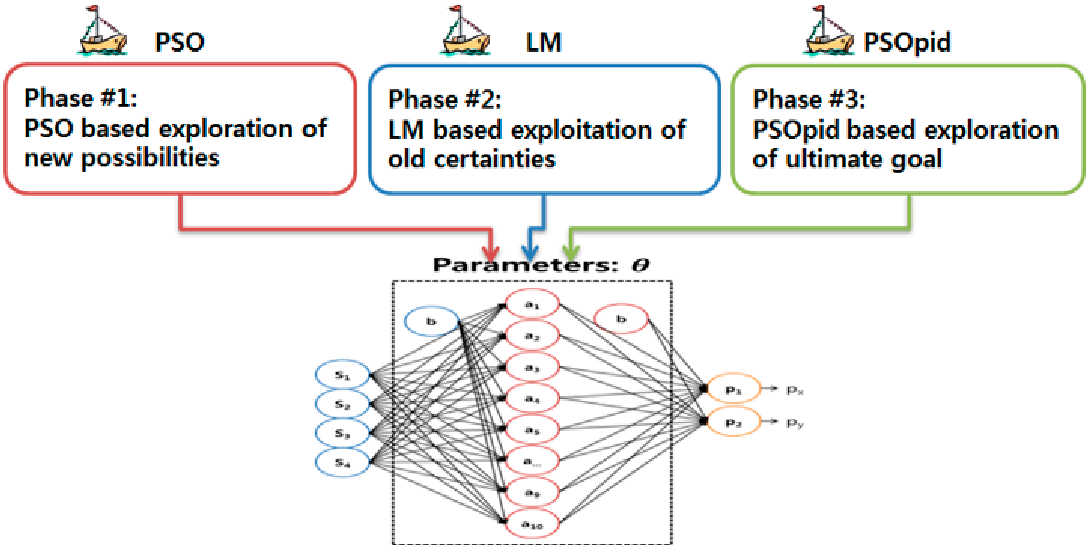

PSO
Phase #1:
PSO based exploration of new possibilities
LM
Phase #2:
LM based exploitation of old certainties
PSOpid
Phase #3:
PSOpid based exploration of ultimate goal
Parameters: θ
b
a₁
a₂
a₃
a₄
a₅
a...
a₉
a₁₀
b
s₁
s₂
s₃
s₄
p₁
p₂
pₓ
pᵧ

Figure 5. Three-phase hybrid optimization method.

In the case of PSO and PSOpid alone, the range of each particle position become unstable or explodes; therefore, the hybrid of three algorithms can be used to not only exploit old certainty optimum but also to explore new possibilities. Through all evaluative processes, this paper shows the enhanced performance by a hybrid scheme of ordinary PSO, LM, and PSOpid, namely PSOpid-LM-PSO. The results showed that the proposed PSOpid-LM-PSO provides faster convergence and better fitness than other algorithms within the range-optimized search space, as shown in Figure 6. Each coordinate in Figure 6 indicates the minimum and maximum of the range for parameters in multi-sensor data fusion model. This well-optimized data fusion model for the sensing information of a set of homogeneous FSRs is expected to provide an enhancement in synergistic accuracy.

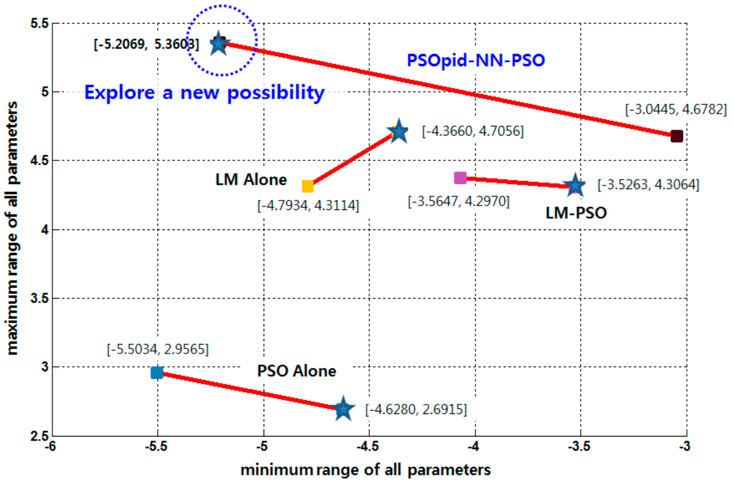


Figure 6. Exploration of a new possibility. Each (x,y) coordinate indicates the minimum and maximum of all parameters such as weights and biases, and the outcomes are from each independent trial.

4. Performance Analysis

We first evaluated the experimental dataset of the touch floor system in [16] and selected men and women from 58 kg to 90 kg as participants. While a user is standing on the floor, the user weight is distributed to four corners according to the inverse power-law in the distance vs. force relation, and the force on each FSRs is sampled through a resistive divider with noise filters. Therefore, it is reasonable to perform experiments with various weighing participants. In this experiment, we evaluated the mean distance error between prediction user location and test set location. As shown in Figure 7, five different algorithms, geometrical trilateration [25], ordinary LM-based neural network (LM), PSO-based neural network (PSO), two phases method (LM-PSO), and the proposed three-phase hybrid method (PSOpid-LM-PSO), are evaluated to investigate whether the suggested method is sufficiently robust to identify the position of differently weighing participants in realistic indoor conditions. In the evaluation, each algorithm was trained with same sized learning data consisting of 100 steps per each participant with 4 × 185 matrix. In addition, these algorithms used the same normalized input data via preprocessing with mean cancellation, principle component analysis and covariance equalization.

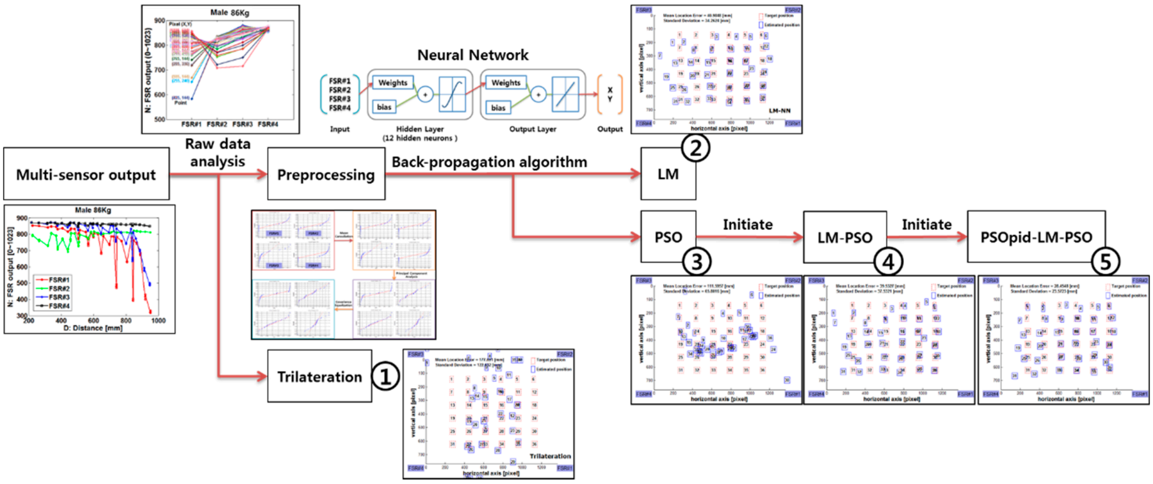


Figure 7. Performance analysis.

Table 3 presents the results from a comparison test consisting of 100 independent trials. In each trial, an inner random number generator was initialized with a nonnegative integer based on the current clock time, and, thus, the randomized functions can produce a predictable sequence of numbers. As described in Table 3, the estimation error of the proposed PSOpid-LM-PSO is reduced by approximately 88.77% in comparison with the trilateration method. (e.g., in 58 kg case, error reduction is calculated as follows: (1.0 − 23.18 mm/546.07 mm) × 100 = 95.76%).

Table 3. Comparison results of the algorithm enhancement rates (out of 100 independent runs).

Single-Subject Evaluation: Mean Distance Error [mm]					
Weight	Trilateration	LM alone	LM-PSO	PSOpid-LM-PSO	PSOpid-LM-PSO Error Reduction
58 kg	546.07	26.28	24.85	23.18	95.76%
64 kg	213.54	30.39	27.01	25.14	88.23%
72 kg	637.15	31.34	30.07	24.76	96.11%
85 kg	160.16	50.14	44.02	41.48	74.10%
90 kg	196.56	27.32	24.96	20.35	89.65%

Figure 8 shows the rate of enhancement under the PSOpid-LM-PSO in comparison with the classic LM. The PSOpid-LM-PSO method decreased the mean location error by about 18.57% compared to the LM alone case and the overall performance was enhanced [16].

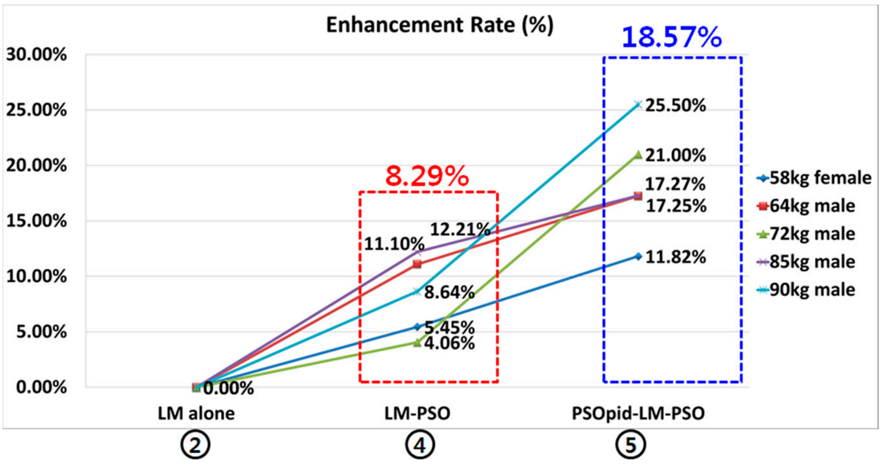
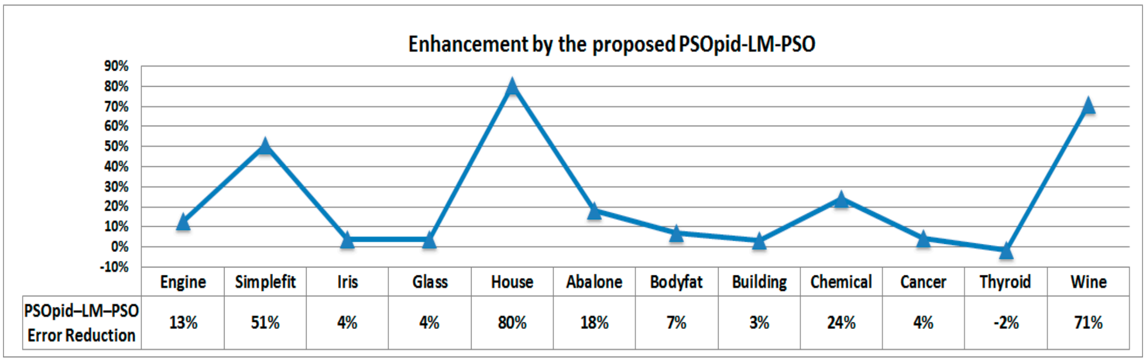


Figure 8. Enhancement graph of each algorithm as compared to the classic LM-based backpropagation algorithm.

In contrast, a performance analysis was performed using different exemplary datasets from MATLAB [26], which are widely used to evaluate the performance of machine learning, to confirm that the proposed method is robust enough to improve stability, robustness and convergence speed in terms of neural network training. In this study, we used the three values of the PID (K_P, K_I, and K_D) which are fixed to 0.5, 0.4, and 0.3, respectively to verify the reliability of performance evaluation of the proposed methodology in different exemplar datasets. Other parameters were used in the same manner with the generic PSO and the PSOpid approach. Figure 9 shows the enhancement of hybridizations among PSOpid-LM-PSO and the other algorithms. The results are summarized with the best, median, and worst results (out of 100 independent runs) reported. In the case of PSO and PSOpid, the range of each particle position becomes unstable or explodes; therefore, neither algorithm alone can be used to exploit the old certainty optimum as well as explore new possibilities. Through all evaluative processes, the proposed PSOpid-LM-PSO showed enhanced performance by using a hybrid scheme of ordinary PSO, LM, and PSOpid.



	Engine	Simplefit	Iris	Glass	House	Abalone	Bodyfat	Building	Chemical	Cancer	Thyroid	Wine
PSOpid–LM–PSO Error Reduction	13%	51%	4%	4%	80%	18%	7%	3%	24%	4%	-2%	71%

Figure 9. Performance test using a sample dataset.

5. Conclusions

This study proposed a hybridization of enhanced particle swarm optimization (PSO) and a classic neural network to build a multi-sensor data fusion model. The results show that the proposed PSOpid-LM-PSO provides a faster convergence and better fitness than other algorithms within the range-optimized search space with the system accuracy improved by approximately 18.6% compared to the classic LM algorithm. The contributions of this paper will reduce human effort in training the data fusion model using an on-line adaptation approach based on small changes from a prior trained model, or by using support vector machines for the one-to-many mapping approximation [27,28]. Another important area for further investigation is to explore the universal approximation capabilities of a standard multi-layer feed-forward neural network in most applications where numerous input samples need to be processed. Specifically, a two-hidden-layer feed-forward network using Kolmogorov's theorem can be considered for approximating the high-dimensional data fusion model [29,30]. This well-optimized model for data fusing from the sensing information of homogeneous sensors is expected to support an enhancement in the synergistic accuracy.

Author Contributions: Conceptualization, H.K.; Methodology, H.K.; Software, H.K. and D.S.; Validation, H.K. and D.S.; Formal Analysis, H.K.; Investigation, H.K.; Resources, H.K.; Data Curation, H.K.; Writing-Original Draft Preparation, H.K. and D.S.; Writing-Review & Editing, H.K. and D.S.; Visualization, H.K.; Supervision, D.S.; Project Administration, H.K.; Funding Acquisition, H.K.

Acknowledgments: This work was supported by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government [18ZH1100, Distributed Intelligence Core Technology of Hyper-Connected Space].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lippi, M.; Mamei, M.; Mariani, S.; Zambonelli, F. Distributed Speaking Objects: A Case for Massive Multiagent Systems. In Proceedings of the International Workshop on Massively Multi-Agent Systems (MMAS2018), Stockholmsmässan, Stockholm, Sweden, 14 July 2018; pp. 1–18.
2. Smaiah, S.; Sadoun, R.; Elouardi, A.; Larnaudie, B.; Bouaziz, S.; Boubezoul, A.; Vincke, B.; Espié, S. A Practical Approach for High Precision Reconstruction of a Motorcycle Trajectory Using a Low-Cost Multi-Sensor System. *Sensors* **2018**, *18*, 2282. [CrossRef] [PubMed]
3. Jagannathan, S.; Mody, M.; Jones, J.; Swami, P.; Poddar, D. Multi-sensor fusion for Automated Driving: Selecting model and optimizing on Embedded platform. *Auton. Veh. Mach.* **2018**, 256-1–256-5. [CrossRef]
4. Kim, D.; Rodriguez, S.; Matson, E.T.; Kim, G.J. Special issue on smart interactions in cyber-physical systems: Humans, agents, robots, machines, and sensors. *ETRI J.* **2018**, *40*, 417–420. [CrossRef]
5. Dong, J.; Zhuang, D.; Huang, Y.; Fu, J. Advances in multi-sensor data fusion: Algorithms and applications. *Sensors* **2009**, *9*, 7771–7784. [CrossRef] [PubMed]
6. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
7. Kendall, G. Is Evolutionary Computation Evolving Fast Enough? *IEEE Comput. Intell. Mag.* **2018**, *13*, 42–51. [CrossRef]
8. Xiong, N.; Svensson, P. Multi-sensor management for information fusion: Issues and approaches. *Inf. Fusion* **2002**, *3*, 163–186. [CrossRef]
9. LeCun, Y.; Bottou, L.; Orr, G.B.; Müller, K.-R. Efficient backprop. In *Lecture Notes in Computer Science*; Springer: Berlin/Heidelberg, Germany, 1998; pp. 9–50.
10. Gravina, R.; Alinia, P.; Ghasemzadeh, H.; Fortino, G. Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges. *Inf. Fusion* **2017**, *35*, 1339–1351. [CrossRef]
11. Novak, D.; Riener, R. A survey of sensor fusion methods in wearable robotics. *Rob. Auton. Syst.* **2015**, *73*, 155–170. [CrossRef]
12. March, J.G. Exploration and exploitation in organizational learning. *Organiz. Sci.* **1991**, *2*, 71–87. [CrossRef]
13. Li, Y.; Tian, X.; Liu, T.; Tao, D. On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 1975–1985. [CrossRef] [PubMed]
14. Yang, Q.; Chen, W.N.; Deng, J.D.; Li, Y.; Gu, T.; Zhang, J. A Level-based Learning Swarm Optimizer for Large Scale Optimization. *IEEE Trans. Evol. Comput.* **2017**, *22*, 578–594. [CrossRef]
15. Xu, J.; Yang, W.; Zhang, L.; Han, R.; Shao, X. Multi-sensor detection with particle swarm optimization for time-frequency coded cooperative WSNs based on MC-CDMA for underground coal mines. *Sensors* **2015**, *15*, 21134–21152. [CrossRef] [PubMed]
16. Kim, H.; Chang, S. High-resolution touch floor system using particle swarm optimization neural network. *IEEE Sens. J.* **2013**, *13*, 2084–2093. [CrossRef]

17. Chen, G.; Yu, J. In Particle swarm optimization neural network and its application in soft-sensing modeling. In *International Conference on Natural Computation*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 610–617.
18. Gudise, V.G.; Venayagamoorthy, G.K. Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium (SIS'03), Indianapolis, IN, USA, 24–26 April 2003; pp. 110–117.
19. Lv, C.; Xing, Y.; Zhang, J.; Na, X.; Li, Y.; Liu, T.; Cao, D.; Wang, F.Y. Levenberg-Marquardt Backpropagation Training of Multilayer Neural Networks for State Estimation of a Safety Critical Cyber-Physical System. *IEEE Trans. Ind. Inf.* **2017**, *14*, 3436–3446. [CrossRef]
20. Kim, H.; Chang, S.; Kang, T.G. Enhancement of particle swarm optimization by stabilizing particle movement. *ETRI J.* **2013**, *35*, 1168–1171. [CrossRef]
21. Interlink Electronics, Inc. Force Sensing Resistor (FSR) Guide. Available online: http://www.interlinkelectronics.com/Force-Sensing-Resistor (accessed on 24 August 2018).
22. Kim, J.-H.; Myung, H. Evolutionary programming techniques for constrained optimization problems. *IEEE Trans. Evol. Comput.* **1997**, *1*, 129–140.
23. Willis, M. *Proportional-Integral-Derivative Control*; University of Newcastle: Newcastle, UK, 1999.
24. Araki, M. PID control. In *Control Systems, Robotics and Automation: System Analysis and Control: Classical Approaches II*; Unbehauen, H., Ed.; EOLSS Publishers Co., Ltd.: Oxford, UK, 2009; pp. 58–79. ISBN 9781848265912.
25. Cotera, P.; Velazquez, M.; Cruz, D.; Medina, L.; Bandala, M. Indoor Robot Positioning Using an Enhanced Trilateration Algorithm. *Int. J. Adv. Rob. Syst.* **2016**, *13*, 110. [CrossRef]
26. Matlab. Sample Data Sets. Available online: https://kr.mathworks.com/help/nnet/gs/neural-network-toolbox-sample-data-sets.html (accessed on 24 August 2018).
27. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
28. Gunn, S.R. Support vector machines for classification and regression. *ISIS Tech. Rep.* **1998**, *14*, 5–16.
29. Kůrková, V. Kolmogorov's theorem and multilayer neural networks. *Neural Netw.* **1992**, *5*, 501–506. [CrossRef]
30. Huang, G.-B.; Chen, L.; Siew, C.K. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Trans. Neural Netw.* **2006**, *17*, 879–892. [CrossRef] [PubMed]