Hindawi Publishing Corporation
ISRN Artificial Intelligence
Volume 2014, Article ID 864020, 13 pages
http://dx.doi.org/10.1155/2014/864020

Research Article

Weighed Nonlinear Hybrid Neural Networks in Underground Rescue Mission

Hongxing Yao,[1,2] Mary Opokua Ansong,[1,3] and Jun Steed Huang[4]

[1] *Institute of System Engineering, Faculty of Science, Jiangsu University, 301 Xuefu, Zhenjiang 212013, China*
[2] *College of Finance and Economics, Jiangsu University, 301 Xuefu, Zhenjiang 212013, China*
[3] *Department of Computer Science, School of Applied Science, Kumasi Polytechnic, P.O. Box 854, Kumasi, Ghana*
[4] *Computer Science and Technology, School of Computer Science & Telecommunication, Jiangsu University, 301 Xuefu, Zhenjiang 212013, China*

Correspondence should be addressed to Hongxing Yao; hxyao@ujs.edu.cn

Received 25 September 2013; Accepted 11 November 2013; Published 22 January 2014

Academic Editors: O. Castillo, K. W. Chau, D. Chen, and P. Kokol

In our previous work, a novel model called compact radial basis function (CRBF) in a routing topology control has been modelled. The computational burden of Zhang and Gaussian transfer functions was modified by removing the power parameters on the models. The results showed outstanding performance over the Zhang and Gaussian models. This study researched on several hybrids forms of the model where cosine ($_{\cos}$) and sine ($_{\sin}$) nonlinear weights were imposed on the two transfer functions such that $Y(\text{out}) = \text{logsig}(R) + [\exp(-\text{abs}(R))] * (\pm\cos \text{ or } \pm\sin(R))$. The purpose was to identify the best hybrid that optimized all of its parameters with a minimum error. The results of the nonlinear weighted hybrids were compared with a hybrid of Gaussian model. Simulation revealed that the negative nonlinear weights hybrids optimized all the parameters and it is substantially superior to the previous approaches presented in the literature, with minimized errors of 0.0098, 0.0121, 0.0135, and 0.0129 for the negative cosine (HSCR-BF$_{-\cos}$), positive cosine (HSCR-BF$_{+\cos}$), negative sine (HSCR-BF$_{-\sin}$), and positive sine (HSCR-BF$_{+\sin}$) hybrids, respectively, while sigmoid and Gaussian radial basis functions (HSGR-BF$_{+\cos}$) were 0.0117. The proposed hybrid could serve as an alternative approach to underground rescue operation.

1. Introduction

1.1. Background. In our earlier work we demonstrated how a routing path was generated and how the compact radial basis function could be improved by reducing the computational burden of Gaussian by removing the power parameter from the model. We had discussed the robustness and fault tolerant nature of the compact radial basis function for an emergency underground rescue operation and had discussed the performance of the sigmoid basis function and the compact radial basis function of which the latter optimised its parameters better than that of the former [1]. In this paper we look at the hybrid form of this novel algorithm, by introducing nonlinear weights of positive and negative cosine and sine.

1.2. Sigmoid Basis Function (SBF) and Radial Basis Function (RBF). Sigmoid basis function (SBF) and radial basis function (RBF) are the most commonly used algorithms in neural training. The output of the network is a linear combination of radial basis function of the inputs and neural parameters. Radial basis function networks have many uses, including function approximation, time series prediction [2, 3] classification, and system control. The structure supports the academic school of connectionist and the idea was first formulated in 1988 by Broomhead and Lowe [4]. The SBF, a mathematical function having an "S" shape (sigmoid curve), and is related to brain reasoning and the structure favors the computational believers. The sigmoid function refers to the special case of the logistic function. Another example is the Gompertz curve which is used in modeling systems that saturate at large values of input, for example, the ogee curve used in spillway of dams. A wide variety of sigmoid functions have been used as activation functions of neurons, including the logistic and hyperbolic tangent weight

functions. Sigmoid curves are also common in statistics such as integrals and logistic distribution, normal distribution, and Student's probability density functions. In our opinion, SBF offers nonlinear effects for large input value and RBF provide nonlinear effect at small input value. A nonlinear hybrid of cosine and sine will result in more nonlinear blending across the entire region.

1.3. Wireless Sensor Networks (WSN). Wireless sensor networks (WSN) gather and process data from the environment and make possible applications in the areas of environment monitoring, logistics support, health care, and emergency response systems as well as military operations. Transmitting data wirelessly impacts significant benefits to those investigating buildings, allowing them to deploy sensors which monitor from a remote location. Multihop transmission in wireless sensor networks conforms to the underground tunnel structure and provides more scalability for communication system construction in rescue situations. A significant discovery in the field of complex networks has shown that a large number of complex networks including the internet are scale-free and their connectivity distribution is described by the power-law of the form $\rho(k) \sim k^{-\phi}$, to allow few nodes of very large degree to exist making it difficult for random attack. A scale-free wireless network topology was therefore used.

This paper proposes a nonlinear Hybrid Neural Networks using radial and sigmoid transfer functions in underground communication, based on particle swarm optimisation. An alternative to this model is without hybrid, either RBF or SBF. The SBF is known to be fast, while RBF is accurate [5] and therefore blending the two will provide both speed and accuracy. In addition the two models have been examined in our previous work [1]. To this end, we model the incident location as a pure random event and calculate the probability that communication chain through particular rock layers to the ground is not broken and allows neural network to memorize the complicated relationship, such that when real accident happens, the neural network resident in the robot is used to predict the probability based on the rock layer it sees instantly. If the result is positive, the robot waits to receive the rescue signal; otherwise it moves deeper to the next layer and repeats the procedure.

Section 2 explains the preliminaries to the study and generates the routing path that has the highest survival probability for the neural training. Section 3 discusses related work to the study. Section 4 discusses the network optimization model based on the nonlinear weight on the compact radial basis function. Section 5 highlights the method used and Section 6 shows the simulation results of the various hybrids and compare with the Gaussian model whiles Section 7 gives a summary of the findings and future work.

2. Preliminaries

2.1. Sensor Deployment. Some assumptions, such as 20% of software failure, 2 safe exits assumed to be available after accidents and additional errors committed after the accidents, and the failure rate of radio frequency identification (RFID), were made in various sections. These assumptions are already included in the model and they are there to ensure that the system remains reliable.

Topological deployment of sensor nodes affects the performance of the routing protocol [6, 7]. The ratio of communication range to sensing range as well as the distance between sensor nodes can affect the network topology.

Let Ω be the sensor sequence for the deployment of total sensors $T = xyz = L, R, C$, such that

$$\Omega = \begin{cases} \text{For } t = t+1 \\ \text{node}\,(t,2) = \left(\dfrac{\left\| -(R+1) * \left(1 + (-1)^{\text{tog}J}\right) \right\|}{2} + j \right) \\ \text{node}\,(t,3) = \left(\dfrac{\left\| -(C+1) * \left(1 + (-1)^{\text{tog}K}\right) \right\|}{2} + k \right) \end{cases}$$
$$\text{for, } i = 1:L,\ \ j = 1:R,\ \ k = 1:C \text{ and node } (t,1) = i. \tag{1}$$

$\text{tog}^J = \text{ceil}(t/C/R)$ and $\text{tog}^K = \text{ceil}(t/C)$ check source and destination node, respectively.

$\Omega(i, j, k) = \{1, 1, 1\}, \{1, 2, 1\}, \ldots, \{i\text{th}, j\text{th}, k\text{th}\}$ for [level 1, row 1, column 1], [level 1, row 2, column 1], [...], and [ith level, jth row, and kth column], respectively. Therefore $T = T \times T$ matrix, in an underground mine with dimensions of $L = 3$, $R = 2$, and $C = 1$ for depth (level), row (length), and width (column), respectively, with "pm" a sensor apart, implies that a minimum of 6 sensors will have to be deployed.

2.2. Communication. The Through-The-Earth (TTE) Communication system transmits voice and data through solid earth, rock, and concrete and is suitable for challenging underground environments such as mines, tunnels, and subways. There were stationary sensor nodes monitoring carbon monoxide, temperature, and so forth, as well as mobile sensors (humans and vehicles) distributed uniformly. Both stationary and mobile sensor nodes were connected to either the Access Point (AP) and/or Access Point Heads (AP Heads) based on transmission range requirements [6]. The AP Heads serve as cluster leaders and are located in areas where the rock is relatively soft or signal penetration is better to ensure that nodes are able to transmit the information they receive from APs and sensor nodes. The APs are connected to other APs or Through-The-Earth (TTE) which is dropped through a drilled hole down 300 meters apart based on the rock type. The depth and rock type determine the required number of TTEs needed. Next the data-mule is discharged to carry items such as food, water, and equipments to the miners underground and return with underground information to rescue team.

2.3. Signal/Transmission Reach. Major challenges of sensor networks include battery constraints, energy efficiency, network lifetime, harsh underground characteristics, better transmission range, and topology design, among others. Several routing approaches for safety evacuation have been

Table 1: Common rocks found in typical mines in relation to hardness or softness.

Nonlinear mapping	Mica	Coal	Granite	Feldspar	Quartz	Mineral
Softness	0.70	0.80	0.83	0.86	0.875	0.90
Hardness	2	3	5	6	7	9
Distance	750 m	470 m	390 m	315 m	278	78 m

proposed [8–14]. These were developed depending on specific emergency situations and management requirements. Transmitting data wirelessly impacts significant benefits to those investigating buildings, allowing them to deploy sensors and monitor from a remote location [15, 16]. To effectively gain the needed results, researchers have come out with a number of techniques to address the problem of topology control (TC). These include localization of nodes and time; error and path loss; transmission range and total load each node experiences; as well as energy conservation which is very crucial in optimizing efficiency and minimizing cost in wireless sensor networks [17–19]. Minimizing transmission range of wireless sensor networks is vital to the efficient routing of the network because the amount of communication energy that each sensor consumes is highly related to its transmission range [6]. The nodes signal reach $N\delta$ was defined as the integration of the change of the minimum and maximum signal reach, taking into consideration the number of cases (τ) of the rock structure β, β is the rock hardness, $N\delta$ is the signal reach for a node, and $N\delta_{\min}, N\delta_{\max}$ are minimum and maximum signal reach, respectively. The node signal reach is calculated as

$$\begin{aligned} N\delta &= N\delta_{\min} + \left\| N\delta_{\min} - N\delta_{\max} \right\| {}^{*}\text{random}\left({}'\beta', \text{rock}, 1, 1 \right), \\ N\delta_{\min} &= \min\left(L, \min\left(\text{Row}, \text{Col}\right)\right), \\ N\delta_{\max} &= \max\left(L, \max\left(\text{Row}, \text{Col}\right)\right), \end{aligned} \tag{2}$$

and where β is a geometric figure in the range of 0.7 to 0.9. For a connection to be made the absolute difference between i, j should be less than the node signal reach-$N\delta$. The connection matrix was given as

$$\varphi k\left(i, j\right) = \begin{cases} 1 & \text{if } \left\| i - j \right\| \le N\delta \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The relationship between rock hardness and the signal reach is a complicated nonlinear function, which is related to the skin depth of the rock with alternating currents concentrated in the outer region of a conductor (skin depth) by opposing internal magnetic fields as follows:

$$\text{Skin depth} = \sqrt{\frac{2}{\left(\rho * \omega * \sigma\right)}}. \tag{4}$$

ρ is material conductivity, ω is frequency, and σ is magnetic permeability.

The signal (B-field) is attenuated by cube of distance (d) : $B = (k)d^3$.

Signal reach (distance) is equal to 3^* skin depth.

Table 1 identifies 6 common rocks found in mines in relation to hardness or softness of each rock.

A routing path was modeled using a number of $T \times T$ size matrices namely the connection matrix (φk), routing matrix (φr), explosion matrix (φx), failed matrix (φf), hope matrix (φh), optimized matrix (φo), and the exit matrix (φe). The hardware survival rate vector (φh), the survival rate vector (φv) of each miner, and the final average survival rate vector (R) were also generated. A sensor node is named by its 3D integer (x, y, z) coordinates, where $1 \le x \le R$, $1 \le y \le C$, $1 \le z \le L$ for $T = R * C * L$ being total number of nodes. If the node (a, b, c) is connected with node (d, e, f) then the element on $((a-1) * C * L + (b-1) * L + c)$th row and $((d-1) * C * L + (e-1) * L + f)$th column is 1; otherwise 0 and routing are limited to total number of multiple points connections that can be made. In arriving at the final optimized vector for transmission, each matrix was generated τ times.

The $\varphi r \subset \varphi k$; $M_\rho \le \alpha$, M_ρ is even, M_ρ representing the maximum point-to-multipoint connection, and M_ρ is even allowing bidirectional communication, with i, j checking source and destination nodes, respectively:

$$\varphi r = \begin{cases} 1 & \text{if } \left\| i - j \right\| \ge \left(\dfrac{M_\rho}{2} \right), \\ 0 & \text{otherwise;} \end{cases} \qquad \text{for } i, j = 1 : T. \tag{5}$$

2.4. Hardware, Software, and Network Fault Tolerant Considerations. Network security is a critical issue in wireless sensor networks as it significantly affects the efficiency of the communication and many key management schemes had been proposed to mitigate this constraint [18, 20]. In an event of accidents (ψ) occurring, the routing path would be affected by $(1-\psi)$, where ψ is any random value within β, which would cause explosion on $R\varphi$ matrix and result in φx such that the resulting matrix would be the failed matrix (φf):

$$\begin{aligned} \varphi x &= \left(1 - \psi\right) \varphi r, \\ \varphi f\left(i, j\right) &= \begin{cases} 1 & \text{if } \left\{ \varphi x\left(i, j\right) < \lambda L \right\} \\ 0 & \text{if } \left\{ \varphi x\left(i, j\right) \ge \lambda L \right\} \end{cases} \end{aligned} \tag{6}$$

$\varphi f(i, j) = 0$ if $\{\varphi x(i, j) \ge \lambda H\}$ else $\varphi f = \varphi x(i, j)/\lambda H$ for (λL or λH) representing the lower and higher accident impact thresholds, respectively.

A new set of routing paths (φh) and exit matrices (φe) for transmission was calculated as

$$\begin{aligned} \varphi h &= \varphi f * \varphi r, \\ \varphi e &= Ne - \psi. \end{aligned} \tag{7}$$

The mathematical objective here is to find an optimized routing matrix (φo) that has the maximum survivability. The survivability is defined as the entropy of a number of parallel connections between every node to all the sink(s). The exit matrix (φe) described the success rate from each node to the sink(s), φe assumes that the number of exits (Ne) is available with an error margin (ψ). In most practical applications, more than one sink is used, and sink node is either through the fiber or Through-The-Earth (TTE) link. It is important to note that, in real rescue situations, the software (relational database management system (RDBMS)) and hardware including radio frequency identification (RFID) may fail as a result of the effect from the explosion and attack. The matrix (φs) was used to describe software survival rate including bugs or attacks:

$$\varphi s = 1 - \left(\left(\frac{1}{T + \text{random}}\right)(\text{“Geometric”}, \text{fail}, T, T)\right). \quad (8)$$

To obtain the final survival vector (φR) it was assumed that each miner will have an RFID; a vector φI was used to describe its failure rate, including risks of running out of battery, a vector φH for the hardware failure rate, for T = total number of nodes:

$$\varphi I = 1 - \left(\left(\frac{1}{T + r}\right) * (\text{“Geometric”}, \text{fail}, 1, T)\right). \quad (9)$$

r is a random number generated from the vector T and $r : 0 \to \infty$, $T + \text{random} : T \to \infty$, $(1/(T + r)) : 1/T \to 0$; $1 - (1/(T + r)) : 1 - 1/T \to 1$ is the minimum; therefore for $T = N$ nodes, we have $1 - 1/N = (N - 1)/N \to 1$ for $(N - 1)/N \to$ node is dead and $1 \to$ node is alive:

$$\varphi H = \min\left(1, \varphi I\,[X\Psi]\right); \quad \text{for } [X\Psi] = \frac{\varphi e}{M_p}. \quad (10)$$

The survival rate of each miner was (φv). All these assumptions happen in real life and must be considered. The final survival rate vector (φR) was calculated:

$$\varphi v = \varphi s * \varphi e, \quad (11)$$

$$\varphi R = \varphi v * \varphi s, \quad (12)$$

$$R = \varphi R. \quad (13)$$

3. Related Work

3.1. Artificial Neural Networks (ANN). Having found the optimum set of routing table that has the highest survival probability of communicating with and rescuing miners, it is important to train the neurons such that the initial error will be minimized and more importantly have a reliable system [21]. The topology of a neural network can be recurrent (with feedback contained in the network from the output back to the input), the feedforward (where data flow from the input to the output units). The data processing can extend over several layers of units, but no feedback connections are present. Many researchers have come out with neural network predictive models in both sigmoid and radial basis functions [22, 23] with applications such as nonlinear transformation [23], extreme learning machine, predicting accuracy in gene classification [24], crisp distributed support vectors regression (CDSVR) model was monthly streamflow prediction was proposed by Valdez et al. [25] and other applications include fuzzy inference systems (FIS) which have been successfully applied in fields such as automatic control, data classification, decision analysis, and expert systems [26] among others. Artificial neural networks (ANN) are learning algorithm used to minimize the error between the neural network output and desired output. This is important where relationships exist between weights within the hidden and output layers, and among weights of more hidden layers. In addition other parameters including mean iteration, standard variation, standard deviation and convergent time (in sections) were evaluated. The architecture of the learning algorithm and the activation functions were included in neural networks. Neurons are trained to process store, recognize, and retrieve patterns or database entries to solve combinatorial optimization problems. Assuming that the input layer has 4 neurons, output layer 3 neurons, and the hidden layer 6 neurons, we can evolve other parameters in the feedforward network to evolve the weight. So the particles would be in a group of weights and there would be $4 * 6 + 6 * 3 = 42$ weights. This implies that the particle consists of 42 real numbers. The range of weight can be set to $[-100, 100]$ or a fitting range. After encoding the particles, the fitness function was then determined. The goodness of the fit was diagnosed using mean squared error (MSE) as against mean cubic error (MCE) and the mean absolute error (MAE). The mean cubic error will allow for fast convergence and that will gross over accuracy making the process unstable, while the mean absolute error is stable but converges slowly. A midway between the two is the MSE and is given as

$$\text{MSE} = \frac{1}{ns}\sum_{j=1}^{n}\sum_{i=1}^{s}\left(Y_{j,i} - y_{j,i}\right)^2, \quad (14)$$

where n is number of samples, s is the number of neurons at output layer, $Y_{i,j}$ is the ideal value of ith sample at jth output, and $Y_{i,j}$ is the actual value of ith sample at jth output.

3.2. The Gaussian and Zhang Models. The standard or direct approach to interpolation at locations $\{x_1, ..., x_N\} \subset R^d$ without the first term, using Gaussian kernel is given as

$$\phi(r) = e^{-(\varepsilon r)^2}. \quad (15)$$

Zhang has [27] tried to modify the Gaussian model as follows: a function $\psi : [0, \infty) \to \mathbb{R}$ such that $k(x, x') = \varphi(\|x - x'\|)$, where, $x, x' \in \chi$ and $x \cdot x'$ denotes the Euclidean norm with $k(x, x') = \exp(-\|x - x'\|^2/\delta^2)$ as an example of the RBF kernels. The global support for RBF radials or kernels has resulted in dense Gram matrices that can affect large datasets and therefore construct the following two equations: $k_{C,v}(x, x') = \phi_{C,v}(\|x - x'\|)k(x, x')$ and $\phi_{C,v}(\|x - x'\|) = [(1 - \|x - x'\|/C)(\cdot)_+]^v$, where $C > 0$, $v \geq (d+1)/2$, and $(\cdot)_+$ is the positive part. The function $\phi_C(\cdot)$ is a sparsifying operator,

which thresholds all the entries satisfying $(\|x - x'\|) \geq C$ to zeros in the Gram matrix. The new kernel resulting from this construction preserves positive definiteness. This means that given any pair of inputs x and x' where $x = x'$ the shrinkage (the smaller C) is imposed on the function value $k(x, x')$; the result is that the Gram matrices K and K_{Cv} can be either very similar or quite different, depending on the choice of C. However Zhang also ended up with an extra power parameter.

4. The Proposed Hybrid

The previous work looked at the Gaussian model and paralyzed the computational power parameter. The result was compact radial basis function (CRBF). This was used to run both SBF and RBF and the results compared. The scalability and processing efficiency were also analyzed.

The novelty of this algorithm, the weighed nonlinear hybrid, was to find several hybrids and the best for rescue operations. The cosine and sine functions were used to reduce high level nonlinear and increase small level nonlinear. Both the previous and the current algorithm used the same preliminary considerations, but the results is slightly different because data are random.

The sigmoid basis function was given as

$$\begin{gathered} \log\ \mathrm{sig}\,(R), \quad R = W \cdot P + B, \\ \log\ \mathrm{sig}\,(W \cdot P + B) = \frac{1}{1 + e^{-(W \cdot P + B)}}. \end{gathered} \tag{16}$$

Neuron function S (sigmoid) is log sig, W is weight matrix, P is input vector, and B is threshold. We therefore proposed a compact radial basis function based on the Gaussian radial basis function and Helens' [28] definition expressed as

$$\begin{gathered} \left[\exp\left(-\|R\|\right)\right], \quad R = W \cdot P + B, \\ \left(\exp\left(-\mathrm{abs}\left((W \cdot P + B)\right)\right)\right), \\ \phi\,(\mathrm{out}) = \left[\exp\left(-\|R\|\right)\right]. \end{gathered} \tag{17}$$

W is weight matrix, P is input vector, and B is threshold. The focus was to improve on the radial basis function for the mine application.

An optimized vector R was generated as the optimum set of transmission routing table that has the highest survival probability for data transmission (13):

$$\begin{gathered} PN = \sum_{i=1}^{N} R_i S_i + \sum_{i=1}^{2} S_i + \sum_{i=1}^{2} S_i S_i + 1 \\ \text{for } 1 \leq i \leq S; \quad 1 \leq k \leq R. \end{gathered} \tag{18}$$

R, S_1, S_2 are number of neurons at input, hidden, and output layers, respectively.

PN is the position of the nth particle:

$$\begin{gathered} W_i = S_i R, W_2 = S_i R, \ldots, W_m = S_m R, \\ P_i = \left[R\left(i - j\right) + K\right] = W_i\,(i, k). \end{gathered} \tag{19}$$

There are two thresholds $(S_1 \Rightarrow B_1)$:

Hidden: $B_1(i, j) = P(RS_1 + S_1 S_2) \Rightarrow \mathrm{S}_1$

Output: $B_2(i, j) = P(RS_1 + S_1 S_2) \Rightarrow S_2$.

From (16), (17), and (18) the nonlinear weight of $[\pm \cos(R)]$ or $[\pm \sin(R)]$ was imposed on the CRBF before being combined with the SBF as (HSCR-BF$_{-\cos}$, HSCR-BF$_{+\cos}$, HSCR-BF$_{-\sin}$, HSCR-BF$_{+\sin}$). The cosine weight was used to keep high level nonlinear for small input value and reduce the nonlinear for large input values, while sine weight was used to keep high level nonlinear for large input value and reduce the nonlinear for small input values:

$$\begin{gathered} \text{HSCR-BF}_{-\cos}: \\ Y\,(\mathrm{out}) = \log\ \mathrm{sig}\,(R) + \left[\exp\left(-\|R\|\right)\right] \cdot {}^{*}[-\cos\,(R)] \\ \text{HSCR-BF}_{+\cos}: \\ Y\,(\mathrm{out}) = \log\ \mathrm{sig}\,(R) + \left[\exp\left(-\|R\|\right)\right] \cdot {}^{*}[+\cos\,(R)] \\ \text{HSCR-BF}_{-\sin}: \\ Y\,(\mathrm{out}) = \log\ \mathrm{sig}\,(R) + \left[\exp\left(-\|R\|\right)\right] \cdot {}^{*}[-\sin\,(R)] \\ \text{HSCR-BF}_{+\sin}: \\ Y\,(\mathrm{out}) = \log\ \mathrm{sig}\,(R) + \left[\exp\left(-\|R\|\right)\right] \cdot {}^{*}[+\sin\,(R)]. \end{gathered} \tag{20}$$

The nonlinear weight of $[+\cos(R)]$ was imposed on the GRBF before being combined with the SBF:

$$\begin{gathered} \text{HSGR-YBF}_{+\cos}: \\ Y\,(\mathrm{out}) = \log\ \mathrm{sig}\,(R) + \left[\exp\left(-\|R\|^2\right)\right] \cdot {}^{*}[+\cos\,(R)]. \end{gathered} \tag{21}$$

5. Particle Swarm Optimization

Particle swarm optimization (PSO), an evolutionary algorithm, is a population based stochastic optimization technique. The idea was conceived by an American researcher and social psychologist James Kennedy in the 1950s. The theory is inspired by social behavior of bird flocking or fish schooling. The method falls within the category of swarm intelligence methods for solving global optimization problems. Literature has shown that the PSO is an effective alternative to established evolutionary algorithms (GA). It is also established that PSO is easily applicable to real world complex problems with discrete, continuous, and nonlinear design parameters and retains the conceptual simplicity of GA [29, 30]. Each particle within the swarm is given an initial random position and an initial speed of propagation. The position of the particle represents a solution to the problem as described in a matrix τ, where M and N represent the number of particles in the simulation and the number of dimensions of the problem, respectively [31, 32]. A random position representing a possible solution to the problem, with an initial associated velocity representing a function of the distance from the particle's current position to the previous

position of good fitness value, was given. A velocity matrix V_{el} with the same dimensions as matrix τ_x described this:

$$\tau_x = \begin{pmatrix} \tau^{11}, \tau^{12}, \ldots, \tau^{1N} \\ \tau^{21}, \tau^{22}, \ldots, \tau^{2N} \\ \vdots \\ \tau^{m1}, \tau^{m2}, \ldots, \tau^{MN} \end{pmatrix},$$
$$V_{\text{el}} = \begin{pmatrix} v^{11}, v^{12}, \ldots, v^{1N} \\ v^{21}, v^{22}, \ldots, v^{2N} \\ \vdots \\ v^{m1}, v^{m2}, \ldots, v^{MN} \end{pmatrix}. \quad (22)$$

The best feasible alternative is the genetic algorithm. However this is generally slow as compared to the PSO and therefore computationally expensive, hence the PSO.

While moving in the search space, particles commit to memorize the position of the best solution they have found. At each iteration of the algorithm, each particle moves with a speed that is a weighed sum of three components: the old speed, a speed component that drives the particles towards the location in the search space, where it previously found the best solution so far, and a speed component that drives the particle towards the location in the search space where the neighbor particles found the best solution so far [7]. The personal best position can be represented by an $N \times N$ matrix ρ_{best} and the global best position is an N-dimensional vector G_{best}:

$$\rho_{\text{best}} = \begin{pmatrix} \rho^{11}, \rho^{12}, \ldots, \rho^{1N} \\ \rho^{21}, \rho^{22}, \ldots, \rho^{2N} \\ \vdots \\ \rho^{m1}, \rho^{m2}, \ldots, \rho^{MN} \end{pmatrix},$$
$$G_{\text{best}} = \left(g_{\text{best}}{}^{1}, g_{\text{best}}{}^{12}, \ldots, g_{\text{best}}{}^{N}\right). \quad (23)$$

All particles move towards the personal and the global best, with τ, ρ_{best}, V_{el}, and g_{best} containing all the required information by the particle swarm algorithm. These matrices are updated on each successive iterations:

$$V_{mn} = V_{mn} + \gamma_{c1}\eta_{r1}\left(p_{\text{best}_{mn}} - X_{mm}\right) + \gamma_{c2}\eta_{r2}\left(g_{\text{best}_n} - X_{mn}\right)$$
$$X_{mn} = V_{mn}. \quad (24)$$

γ_{c1} and γ_{c2} are constants set to 1.3 and 2, respectively, and η_{r1} η_{r2} are random numbers.

5.1. Adaptive Mutation according to Threshold. Particle swarm optimization has been effective in training neural networks such as a Parallel Particle Swarm optimization (PPSO) method with dynamic parameter adaptation used to optimize complex mathematical functions and improved evolutionary method in fuzzy logic [30, 31, 33]. To prevent particles from not converging or converging at local minimum, an adaptive mutation according to threshold was introduced. An alternative to this is the use of the nonadaptive mutation, and this could converge to the local minimum or fail to converge at all. The advantage of the nonadaptation is simple and fast but may not provide the needed results.

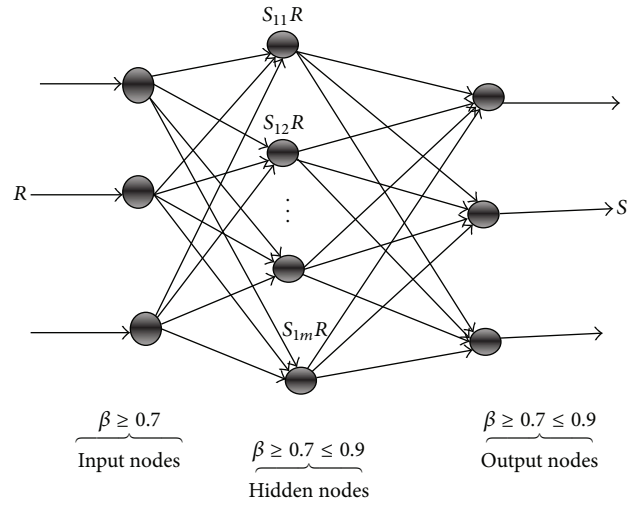


Figure 1: The structure AMPSO for CRBF, GRBF, and SBF transfer functions.

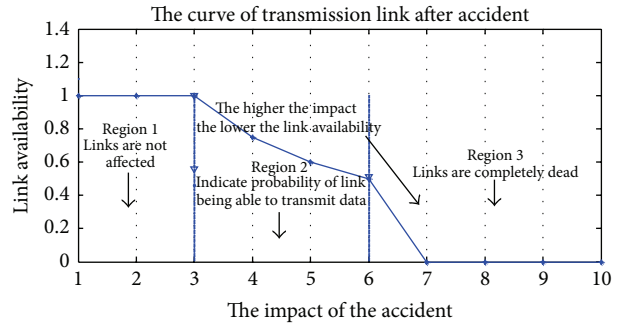


Figure 2: Impact of explosion/accident on transmission link.

Particles positions were updated with new value only when the new value is greater than the previous value; 20% of particles of those obtaining lower values were made to mutate for faster convergence and the structure of adaptive mutation PSO (AMPSO) with threshold can be found in [1]. The input layer takes the final survival vector (13), with a number of hidden layers and an output layer. The feedforward neural network was used. The structure of Adaptive Mutation PSO (AMPSO) with threshold Neural Network was used, Figure 1, [32].

6. Results and Discussion

6.1. Generating the Routing Path. Elements 0, 1, and 2 in φx imply that the link(s) were not affected and elements 3, 4, 5, and 6, represent a probability for the links being able to transmit data, while figures from 7 means the link is totally dead. Region 1 that indicates that links are not affected, region

Table 2: Optimum set of routing table redundancy with the highest survival probability (6 cases).

Vectors for each case							Rock hardness cases
$R1$	0.5409	0.4281	0.8834	0.585	0.8134	0.5163	0.7
$R2$	0.4526	0.5841	0.4547	0.8997	0.8436	0.2696	0.8
$R3$	0.2931	0.5899	0.5903	0.8309	0.6971	0.2898	0.83
$R4$	0.4851	0.6085	0.6913	0.4002	0.5026	0	0.86
$R5$	0.4917	0.5857	0.7805	0.6037	0.5182	0.6472	0.875
$R6$	0.2902	0.7253	0.458	0.5762	0.1397	0	0.9

2 gives the probability of link available, and the last region indicates the link is completely down (Figure 2).

The matrices $\varphi i, \varphi k, \varphi r, \varphi x, \varphi h$ for $\tau = n$, with dimensions of $L = 3$, $R = 2$, and $C = 1$ for depth (level), row (length), and width (column), respectively, were generated as follows:

$$S_{\text{eq}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \\ 3 & 2 & 1 \end{pmatrix}, \qquad i\varphi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$\varphi k = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

1 = signal points,

$$\varphi r = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix} \tag{25}$$

1 = transmission path,

$$\varphi x = \begin{pmatrix} 0 & 1 & 2 & 2 & 0 & 0 \\ 1 & 5 & 1 & 2 & 0 & 0 \\ 0 & 8 & 1 & 5 & 9 & 1 \\ 1 & 1 & 5 & 1 & 1 & 17 \\ 1 & 1 & 4 & 11 & 0 & 2 \\ 4 & 4 & 19 & 5 & 3 & 0 \end{pmatrix}$$

explosion sizes,

$$\varphi f = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & .6 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & .6 & 0 & 1 \\ 1 & 1 & .6 & 1 & 1 & 0 \\ 1 & 1 & .75 & 0 & 1 & 1 \\ .75 & .75 & 0 & .6 & 1 & 1 \end{pmatrix},$$

$$\varphi h = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & .6 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & .6 & 0 & 0 \\ 0 & 0 & .6 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & .6 & 1 & 1 \end{pmatrix},$$

$$\varphi o = \begin{pmatrix} .8 & .8 & .8 & 0 & 0 & 0 \\ .8 & .48 & .8 & 0 & 0 & 0 \\ 0 & 0 & .8 & .48 & 0 & 0 \\ 0 & 0 & .48 & .8 & .8 & 0 \\ 0 & 0 & 0 & 0 & .8 & .8 \\ 0 & 0 & 0 & .48 & .8 & .8 \end{pmatrix}, \tag{26}$$

$$\varphi e = \begin{pmatrix} 1.44 & 1.44 & 1.44 & 0 & 0 & 0 \\ 1.44 & .864 & 1.44 & 0 & 0 & 0 \\ 0 & 0 & 1.44 & .864 & 0 & 0 \\ 0 & 0 & .864 & 1.44 & 1.44 & 0 \\ 0 & 0 & 0 & 0 & 1.44 & 1.44 \\ 0 & 0 & 0 & .864 & 1.44 & 1.44 \end{pmatrix}$$

1 or fractional figures are transmission path.

Element "0" on φf depicts a connection lost, while "1" means that the connection will never go down and represent the connection to the fixed sink node(s) along the edge or the emergency connection to the mobile data mule(s):

$$\varphi \text{s} = \begin{pmatrix} .8333 & .8889 & .9091 & .8333 & .8571 & .8333 \\ .8889 & .8333 & .8750 & .8571 & .8750 & .8333 \\ .8571 & .9167 & .8750 & .9091 & .8889 & .8333 \\ .8750 & .8571 & .8571 & .8571 & .8571 & .8333 \\ .8333 & .8333 & .8571 & .8333 & .8889 & .9000 \\ .9167 & .8333 & .9091 & .8333 & .8889 & .8571 \end{pmatrix}. \tag{27}$$

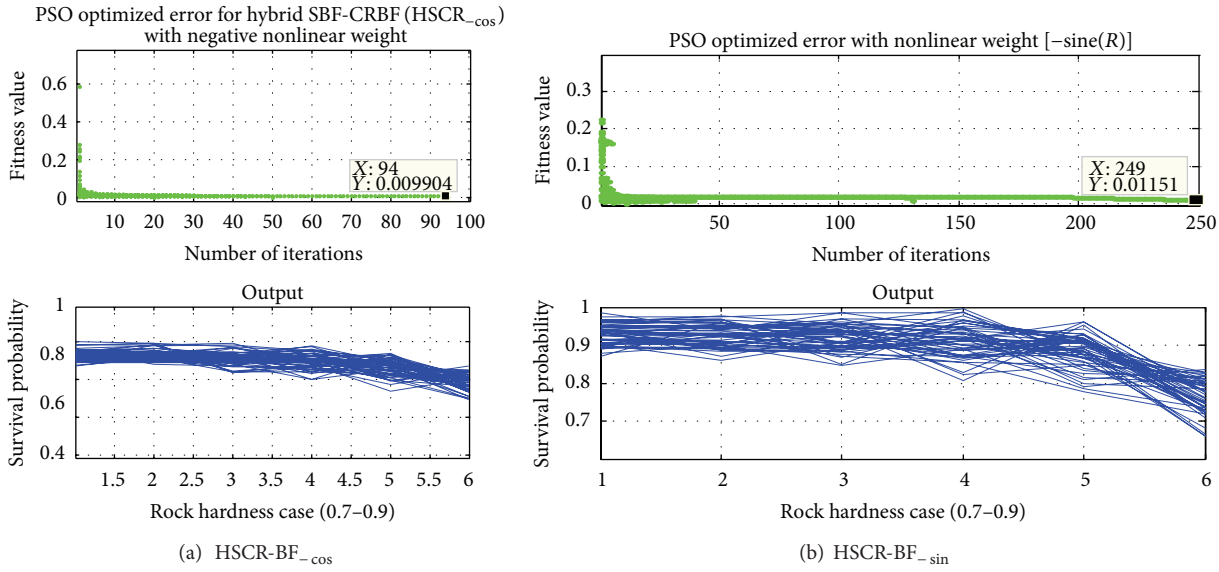
The survival rate of RFID, hardware, and each miner vectors was displayed as

$$\begin{aligned} \varphi I &= .8571 \quad .8750 \quad .8571 \quad .8889 \quad .8571 \quad .8333, \\ \varphi H &= .6236 \quad .4976 \quad 1 \quad .6851 \quad .9286 \quad .6086, \\ \varphi v &= .5409 \quad .4281 \quad .8834 \quad .5850 \quad .8134 \quad .5163. \end{aligned} \tag{28}$$

Optimization was done numerically using Matlab simulation tool to find the optimum set of routing tables through particle swarm search for rescue operation as discussed in the preliminaries. From (13) the final survival vector for dimensions of $L = 3$, $R = 2$, and $C = 1$ will be an average of the 6 vectors as in Table 2. Total cases $Rn = \tau = T \times T$ (which

Table 3: Scalability of model to survival probability range, robot location and rock type.

Location	Mica	Coal	Granite	Feldspar	Quartz	Mineral
Compact radial basis function with negative nonlinear weight (HSCR-BF$_{-\cos}$)						
(10, 6, 5)	0.8769–1.0000	0.9070–0.9927	0.8402–0.9943	0.8023–0.9778	0.7386–0.9376	0.7345–0.8740
(10, 5, 4)	0.9024–0.9696	0.9032–0.9505	0.8836–0.9430	0.8633–0.9527	0.7944–0.9465	0.7263–0.8720
(6, 5, 4)	0.8906–0.9877	0.8827–0.9845	0.8384–0.9675	0.8627–0.9820	0.7911–0.9142	0.6392–0.7840
(3, 1, 10)	0.7302–0.8481	0.6588–0.7816	0.6341–0.7473	0.6503–0.7249	0.5930–0.7156	0.4445–0.5621
Gaussian radial basis function with positive nonlinear weight (HSGR-BF$_{+\cos}$)						
(10, 6, 5)	0.8732–0.9963	0.8526–0.9987	0.8492–0.9950	0.7867–0.9568	0.7404–0.9025	0.6606–0.8315
(10, 5, 4)	0.8843–0.9593	0.8843–0.9853	0.8444–0.9753	0.7746–0.9828	0.7304–0.9595	0.6678–0.8817
(6, 5, 4)	0.8785–0.9983	0.8902–0.9996	0.8200–0.9997	0.7927–0.9446	0.7065–0.8565	0.6572–0.8063
(3, 1, 10)	0.7608–0.8552	0.6600–0.8325	0.6090–0.7371	0.5482–0.6249	0.5026–0.6056	0.4856–0.5453



(a) HSCR-BF$_{-\cos}$ (b) HSCR-BF$_{-\sin}$

Figure 3: Hybrid of SBF and CRBF (HSCR-BF) with (a) HSCR-BF$_{-\cos}$ and (b) HSCR-BF$_{-\sin}$ nonlinear weight.

represent the total nodes deployed) for this scenario. Rock softness or hardness takes the values of β and each matrix is generated τ time/cases and the final routing vector will be an average of the n vectors φR = Average ($R1$: $R6$) [Table 2] which becomes the input for the PSO training.

The final survival rate is R = (0.5409 0.4281 0.8834 0.585 0.8130 0.5163) from (13).

6.2. The Final Survival Vector. The R = (0.5409 0.4281 0.8834 0.585 0.8130 0.5163) from (13) is the maximum survival probability for a total of 6 nodes deployed. It describes the success rate from each node to the sink(s). In most practical applications, more than one sink is used, and sink node is either through the fiber or TTE connection. The size of the vector depends on the dimensions of the field. The elements R = (0.5409 0.4281 0.8834 0.585 0.8130 0.5163) represent the probability of 54%, 43%, 88%, 59%, 81%, and 52% success of each node transmitting data to and from its source or destination. It assists decision makers as to whether to send data through one or more nodes or send each message twice. The total nodes used for the simulation were 300 with underground mine dimensions of L = 10, R = 6 and C = 5 for depth (level), row (length), and width (column), respectively, with "pm" a sensor apart, pm = 100, M_ρ = 4, and Ne = 2. The PSO training used swarm size of 20, maximum position was set to 100, max velocity = 1, and maximum number of iteration = 250. The thresholds λL and λH were 3 and 6, respectively, τ = 6 cases (each case represent, a rock hardness case and also represent, the total neurons used); thus each matrix and vectors were run 6 times before neural training. The survival probability (bottom) indicates that the model survived between 90 and 100%, where rock cases were relatively soft (≥0.7). The survival probability declines as the rock becomes harder and approaches 0.8. At the hardest rock of 0.9, the survival probability fell between 72 and 84% for the entire hybrids. In view of this the AP heads had to be deployed at a location where the rock is relatively soft for maximum signal strength. Each computer simulation incorporated all the 6 different cases of rock hardness/softness to produce the matrices. Detailed location analysis or the scalability of the model in relation to survival

Table 4: Hybrid of SBF and CRBF with negative nonlinear weight on CRBF (HSCR-BF$_{-\cos}$).

Runs	Mean iteration	Std variation	Std deviation	Convergent time	Final error
Total	231.020	343.498	0.382	916.50	0.0985
AVG	23.102	34.350	0.038	91.65	0.0098
Min/Max	29.540	45.314	0.024	80.000	0.0043

Table 5: Hybrid of SBF and CRBF with nonlinear weight on CRBF (HSCR-BF$_{-\sin}$).

Runs	Mean iteration	Std variation	Std deviation	Convergent time	Final error
Total	307.320	488.381	0.376	1143.00	0.135
AVG	30.732	48.838	0.038	114.30	0.013
Min/Max	30.740	56.926	0.007	61.500	0.021

probability range, robot location, and different rock types is recorded in Table 3. Figures 4–6 represent different scenarios of SBF and CRBF hybrids with the $[\pm\cos(R)]$ and $[\pm\sin(R)]$ nonlinear weights, while Figure 7 represents the SBF and Gaussian hybrids for $[+\cos(R)]$ nonlinear weight. The top half of each figure indicates the optimized error or the final error after the neural training and the bottom half reveals model survival probability.

Figure 3 discusses the SBF and CRBF hybrids (HSCR-BF) with nonlinear weight. Training in negative nonlinear weights (HSCR-BF$_{-\cos}$, HSCR-BF$_{-\sin}$) responded well. The HSCR-BF$_{-\cos}$ had a steady and compact routing path which was consistent through all rock layers, with initial survival probability between 87.7% and 100% in soft layers declining to 73.5%–87.4% at harder rock layers. The HSCR-BF$_{-\sin}$ performed quite well but was more dispersed (87.9% to 98.5 at the soft rock and 66.5% to 81.8% at the hard rock).

In addition, the various parameters, mean iteration, standard variance, standard deviation, and the convergent time, of HSCR-BF$_{-\cos}$ were optimised. This is demonstrated by finding the relationship between the maximum and the minimum values of the parameters and comparing with the average figures and the closer the difference is to the average the more consistent the data are in the dataset (Table 4); thus HSCR-BF$_{-\cos}$ provided the best results among the proposed hybrids. The HSCR-BF$_{-\sin}$ on the RBF yielded strength in mean iteration and standard variance with error of 0.013 (Table 5).

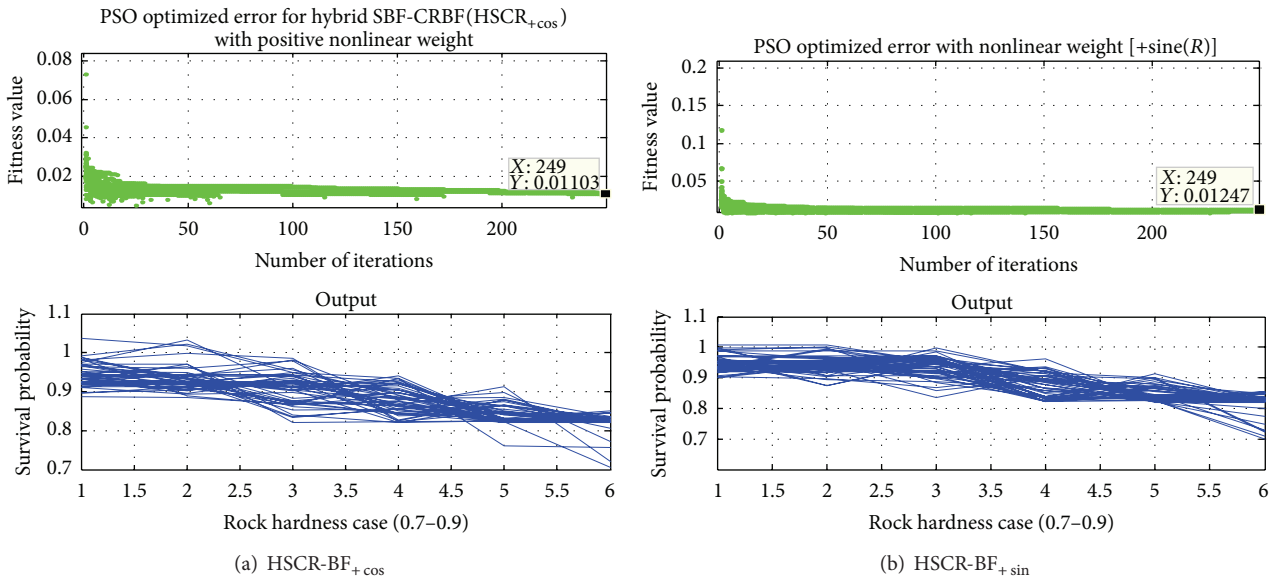
In Figure 4 training with nonlinear weight of positive cosine and sine of HSCR-BF (HSCR-BF$_{+\cos}$ and HSCR-BF$_{+\sin}$) had some similarities. The two models streamed well at the initial stages from 89.6% to 99.2% and 90.8% to 99.0% at soft layers of the rock, respectively. However at the last stages the probability of transmitting effectively became marginal, from 79.5% to 85.2% and 82.1% to 85.4%, respectively. Much as this hybrid could perform well in areas where routing conditions are much better in less dense rescue situations, this could hamper rescue mission in both cases due to battery drain or collision from traffic conjunction. The data for mean iteration and convergent time for HSCR-BF$_{+\cos}$ (Table 6) and mean Iteration, standard variance, the final error HSCR-BF$_{+\sin}$ (Table 7) were consistent with the negative sine having better results than the positive cosine.

The survival probability of the positive nonlinear weight Gaussian hybrids (HSGR-BF$_{+\cos}$) in Figure 5 is more compact both at the initial (i.e., 87.3–99.6%) and later (66.1%–83.2%) stages declining to harder rock layers with an error of 0.01173 (Table 8).

At the initial stage particles are sensitive to inputs as they moved quickly in the search space towards the target and while particles peaked closer to the target it became less sensitive to the input and began to align (Figure 6). At the later stage more RBF are used to keep the error at minimum for accuracy. It was also noticed that instead of accelerating higher, and becoming more sensitive to the inputs, while particles were far from the target, the hybrids with $^*[-\sin(R)]$ and $^*[+\sin(R)]$ were less sensitive to the inputs as $^*[-\sin(R)]$ descended from $[x = -1, y = 0.5785]$ to $[x = 1, y = 0.4215]$, while $^*[+\sin(R)]$ descended from $[x = -3, y = 0.0404]$ to $[x = -1, y = -0.04062]$ before becoming conscious of the inputs. In addition the hybrids with the $^*[+\cos(R)]$ lagged slightly between $[x = -4, y = 0.006014]$ and $[x = -2, y = 0.06288]$ before becoming sensitive to inputs. However they all lined up for accuracy in terms of the minimised error.

The nonlinearity of the hybrids is presented (Figure 7); the negative cosine/sine weight is used to reduce nonlinear for small input values and this lies between $-1 < x < +1$, and the positive cosine/sine weight is used to keep high nonlinear for large inputs for the remaining region.

6.3. CPU Time Efficiency. The relationship between various hybrids with respect to the central processing time (CPU) was profiled for different runs (Table 9) and expressed in a sixth-order polynomial given as $Y_r = \beta_7 X_1^6 + \beta_6 X_1^5 + \beta_5 X_1^4 + \beta_4 X_1^3 + \beta_3 X_1^2 + \beta_2 X_1 + \beta_0$, where X_1 is time (seconds) and β is the coefficient of the polynomial (Figure 8). The proposed hybrid has better usage of CPU time with $R^2 = 0.9160$, followed by HSCR-BF$_{-\sin}$ and HSCR-BF$_{+\cos}$ with R^2 of 0.7551 and 0.7244, respectively (Table 10). Applying the proposed algorithm into Gaussian, the CPU usage had R^2 of 0.7345



(a) HSCR-BF$_{+\cos}$ (b) HSCR-BF$_{+\sin}$

Figure 4: Hybrid of SBF and CRBF (HSCR-BF) with (a) HSCR-BF$_{+\cos}$ and (b) HSCR-BF$_{+\sin}$ nonlinear weight.

Table 6: Hybrid of SBF and CRBF with nonlinear weight on CRBF (HSCR-BF$_{+\cos}$).

Runs	Mean iteration	Std variation	Std deviation	Convergent time	Final error
Total	350.780	522.257	0.311	1156.50	0.121
AVG	35.078	52.226	0.031	115.65	0.012
Max/Min	54.840	63.413	0.009	93.500	0.006

and indeed marginally outperformed HSCR-BF$_{+\cos}$. Detailed work on SBF, CRBF, and GRBF, with regard to scalability, memory usage, and the central processing time, has been carried by the authors [1].

7. Conclusion

In summary, we made the following contributions. First we used the mix of SBF and CRBF to present several hybrids with different nonlinear weights of cosine and sine functions on compact radial basis function. Next we showed the performance of the proposed nonlinear weight hybrids; HSCR-BF$_{-\cos}$, HSCR-BF$_{+\cos}$, HSCR-BF$_{-\sin}$, and HSCR-BF$_{+\sin}$ optimised all the parameters with minimised error of 0.0098, 0.012, 0.013, and 0.013, respectively, compared to 0.0117 for Gaussian HSCR-BF$_{+\cos}$. The analyzed CPU usage with corresponding R^2 values of 0.9160, 0.75, 0.72440, 0.6731 and 0.7345 for HSCR-BF$_{-\cos}$, HSCR-BF$_{+\cos}$, HSCR-BF$_{-\sin}$, HSCR-BF$_{+\sin}$, and HSCR-BF$_{+\cos}$, respectively, demonstrated that the algorithm is scalable. There exist some evacuation models, that is, Goh and Mandic [11] which offered a choice for travelers and several schemes for the decision makers may not be applicable in emergency underground mine situations. *The proposed nonlinear hybrid algorithm with particle swarm optimisation has better capability of approximation to underlying functions with a fast learning speed and high robustness and is competitive and more computationally efficient to Gaussian with the same nonlinear weight*. The algorithm is new and that makes it difficult to identify limitations and we intend to investigate other hybrids and compare with genetic algorithm (GA) in the future.

Nomenclature

$N\delta_{\min}, N\delta_{\max}$: Minimum and maximum signal reach
HSCR-BF/HSGR-BF: Hybrid of SBF with CRBF/Hybrid of SBF with GRBF
HSCR-BF$_{-\cos}$: Hybrid of SBF and CRBF −cos nonlinear weight
HSCR-BF$_{+\cos}$: Hybrid of SBF and CRBF +cos nonlinear weight
HSCR-BF$_{-\sin}$: Hybrid of SBF and CRBF −sine nonlinear weight
HSCR-BF$_{+\sin}$: Hybrid of SBF and CRBF +sine nonlinear weight
HSGR-BF$_{+\cos}$: Hybrid of SBF and GRBF +cos nonlinear weight.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Table 7: Hybrid of SBF and CRBF with nonlinear weight on CRBF (HSCR-BF$_{+\sin}$).

Runs	Mean iteration	Std variation	Std deviation	Convergent time	Final error
Total	260.360	407.109	0.3457	998.00	0.129
AVG	26.036	40.711	0.0346	99.80	0.013
Min/Max	27.840	56.040	0.008	106.500	0.014

Table 8: Hybrid of SBF and GRBF with nonlinear weight on GRBF (HSGR-BF$_{+\cos}$).

Runs	Mean iteration	Std variation	Std deviation	Convergent time	Final error
Total	219.840	349.010	0.439	1121.00	0.1173
AVG	21.984	34.901	0.044	112.10	0.0117
Min/Max	27.000	46.580	0.008	80.000	0.0165

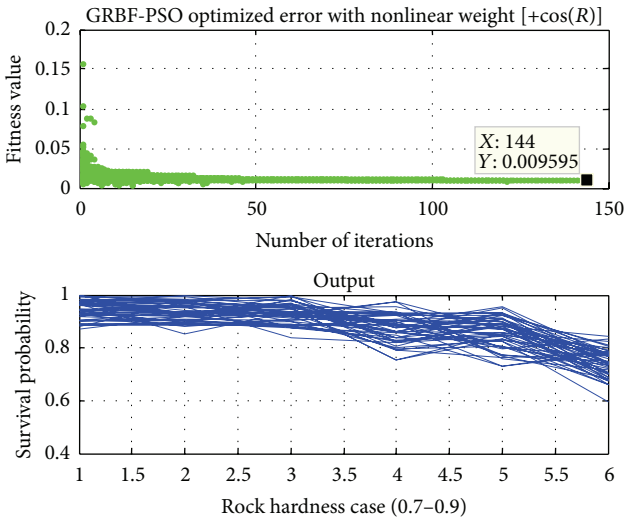


Figure 5: Hybrid of SBF and GRBF with nonlinear weight on GRBF (HSCR-BF$_{+\cos}$).

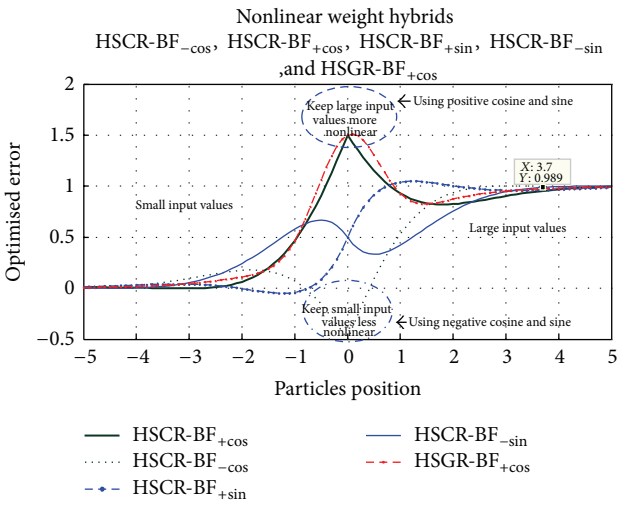


Figure 7: Nonlinear weighed curves for optimized error without Gaussian negative cosine curve.

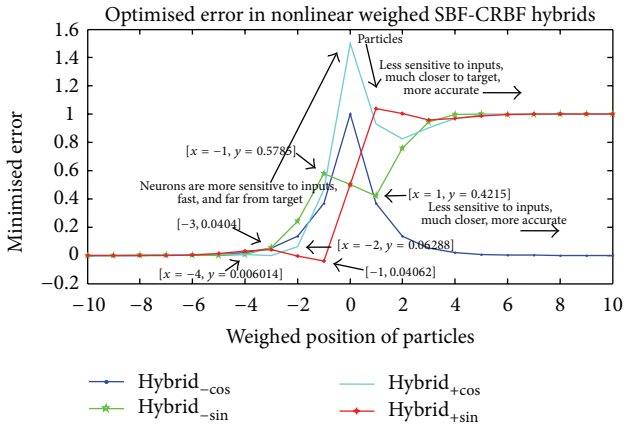


Figure 6: Particles weighed position.

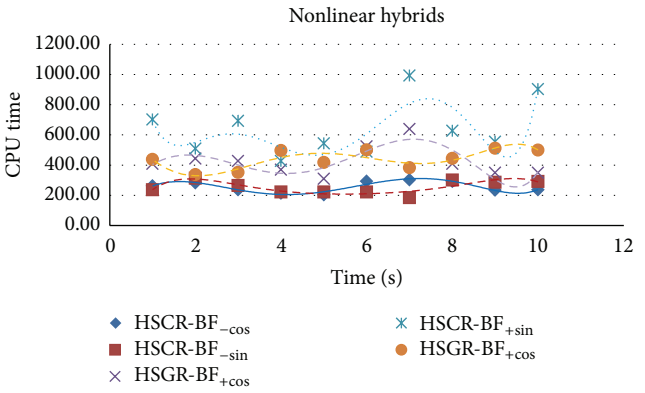


Figure 8: The trend of hybrids CPU time curves using sixth-Order Polynomial.

Acknowledgments

The authors would like to appreciate the immense contribution of the mining companies where the study was undertaken. The authors are grateful to these individuals for their immeasurable contributions to this work: Fred Attakuma, Patrick Addai, Isaac Owusu Kwankye, Thomas Kwaw Annan, Willet Agongo, Nathaniel Quansah, Francis Owusu Mensah, Clement Owusu-Asamoah, Joseph Adu-Mensah, Shadrack Aidoo, Martin Anokye, F. T. Oduro, Ernest Ekow

Table 9: CPU time usage efficiency for the hybrids.

Runs	HSCR-BF$_{-\cos}$	HSCR-BF$_{-\sin}$	HSCR-BF$_{+\cos}$	HSCR-BF$_{+\sin}$	HSGR-BF$_{+\cos}$
Total	2558.89	2543.50	4223.44	6439.62	4381.33
Average	255.89	254.35	422.34	643.96	438.13

Table 10: Analysis of CPU time on the various hybrids.

	β_6	β_5	β_4	β_3	β_2	β_1	β_0	R^2
HSCR-BF$_{-\cos}$	$0.0159x$	$-0.2575x$	$-0.759x$	$33.494x$	$-199.01x$	$403.45x$	25.322	0.9160
HSCR-BF$_{+\cos}$	$0.0903x$	$-2.3158x$	$19.932x$	$-58.027x$	$-31.337x$	$328.08x$	147.47	0.7244
HSCR-BF$_{-\sin}$	$-0.033x$	$1.1354x$	$-15.811x$	$113.68x$	$-433.7x$	$780.08x$	-207.94	0.7551
HSCR-BF$_{+\sin}$	$0.2571x$	$-14.263x$	$171.73x$	$-1006.9x$	$2993x$	$-4237.4x$	2787.9	0.6731
HSGR-BF$_{+\cos}$	$-0.0437x$	$1.0911x$	$-8.2923x$	$6.6218x$	$159.34x$	$-539.39x$	819.1	0.7345

Abano, E. Omari-Siaw, and Qiang Jia. This work was supported by the National Natural Science Foundation of China (no. 71271103) and by the Six Talents Peak Foundation of Jiangsu Province.

References

[1] M. O. Ansong, H. X. Yao, and J. S. Huang, "Radial and sigmoid basis functions neural networks in wireless sensor routing topology control in underground mine rescue operation based on particle swarm optimization," *International Journal of Distributed Sensor Networks-Hindawi*, vol. 2013, Article ID 376931, 14 pages, 2013.

[2] R. Neruda and P. Kudová, "Learning methods for radial basis function networks," *Future Generation Computer Systems*, vol. 21, no. 7, pp. 1131–1142, 2005.

[3] C. T. Cheng, J. Y. Lin, and Y. G. C. Sun, "Long-term prediction of discharges in Manwan hydropower using adaptive-network-based fuzzy inference systems models," in *Advances in Natural Computation*, vol. 3498 of *Lecture Notes in Computer Science*, pp. 1152–1161, 2005.

[4] D. Broomhead and D. Lowe, "Multivariate functional interpolation and adaptive networks," *Complex System*, vol. 2, pp. 321–355, 1988.

[5] Y. Chen, C. Chuah, and Q. Zhao, "Network configuration for optimal utilization efficiency of wireless sensor networks," *Ad Hoc Networks*, vol. 6, no. 1, pp. 92–107, 2008.

[6] P. S. Rajpal, K. S. Shishodia, and G. S. Sekhon, "An artificial neural network for modeling reliability, availability and maintainability of a repairable system," *Reliability Engineering and System Safety*, vol. 91, no. 7, pp. 809–819, 2006.

[7] W. Jang, W. M. Healy, and S. J. Mirosław, "Wireless sensor networks as part of a web-based building environmental monitoring system," *Automation in Construction*, vol. 17, no. 6, pp. 729–736, 2008.

[8] M. S. Pan, C. H. Tsai, and Y. C. Tseng, "Emergengy guiding and monitoring application in indoor 3D environment by Wireless Sensor Network," *Internation Journal Os Sensor Netwourk*, vol. 1, pp. 2–10, 2006.

[9] S. Zarifzadeh, A. Nayyeri, and N. Yazdani, "Efficient construction of network topology to conserve energy in wireless ad hoc networks," *Computer Communications*, vol. 31, no. 1, pp. 160–173, 2008.

[10] R. Riaz, A. Naureen, A. Akram, A. H. Akbar, K. Kim, and H. Farooq Ahmed, "A unified security framework with three key management schemes for wireless sensor networks," *Computer Communications*, vol. 31, no. 18, pp. 4269–4280, 2008.

[11] S. L. Goh and D. P. Mandic, "An augmented CRTRL for complex-valued recurrent neural networks," *Neural Networks*, vol. 20, no. 10, pp. 1061–1066, 2007.

[12] L. A. B. Munoz and J. J. V. Ramosy, "Similarity-based heterogeneous neural networks," *Engineering Letters*, vol. 14, no. 2, pp. 103–116, 2007.

[13] R. Taormina, K. Chau, and R. Sethi, "Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 8, pp. 1670–1676, 2012.

[14] K. Leblebicioglu and U. Halici, "Infinite dimensional radial basis function neural networks for nonlinear transformations on function spaces," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 30, no. 3, pp. 1649–1654, 1997.

[15] F. Fernández-Navarro, C. Hervás-Martínez, J. Sanchez-Monedero, and P. A. Gutiérrez, "MELM-GRBF: a modified version of the extreme learning machine for generalized radial basis function neural networks," *Neurocomputing*, vol. 74, no. 16, pp. 2502–2510, 2011.

[16] C. L. Wu, K. W. Chau, and Y. S. Li, "Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques," *Water Resources Research*, vol. 45, no. 8, Article ID W08432, 2009.

[17] C. E. A. Cheng, "Long-term prediction of discharges in Manwan Reservoir using artificial neural network models," in *Advances in Neural Networks—ISNN 2005*, vol. 3498 of *Lecture Notes in Computer Science*, pp. 1040–1045, 2005.

[18] H. H. Zhang, M. Genton, and P. Liu, "Compactly supported radial basis function kernels," 2004.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1945, December 1995.

[20] J. Eberhart, R. Eberhart, and Y. Shi, "Particle swarm optimization: developments, applications and resources," *IEEE*, vol. 1, pp. 81–86, 2001.

[21] R. Malhotra and A. Negi, "Reliability modeling using Particle Swarm optimisation," *International Journal of System Assurance Engineering and Management*, vol. 4, no. 3, pp. 275–283, 2013.

[22] D. Gies and Y. Rahmat-Samii, "Particle swarm optimization for reconfigurable phase-differentiated array design," *Microwave and Optical Technology Letters*, vol. 38, no. 3, pp. 168–175, 2003.

[23] F. Valdez, P. Melin, and C. Oscar, "An improved evolutionary method with fuzzy logic for combining Particle Swarm optimization and genetic algorithms," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2625–2632, 2011.

[24] J. Zhang and K. Chau, "Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization," *Journal of Universal Computer Science*, vol. 15, no. 4, pp. 840–858, 2009.

[25] F. Valdez, P. Melin, and O. Castillo, "Parallel Particle Swarm optimization with parameters adaptation using fuzzy logic," in *Proceedings of the 11th Mexican International Conference on Artificial Intelligence (MICAI '12)*, vol. 2, pp. 374–385, San Luis Potosi, Mexico, October 2012.

[26] J. Peng and C. P. Y. Pan, "Particle swarm optimization RBF for gas emission prediction," *Journal of Safety Science and Technology*, vol. 7, pp. 77–85, 2011.

[27] G. Ren, Z. Huang, Y. Cheng, X. Zhao, and Y. Zhang, "An integrated model for evacuation routing and traffic signal optimization with background demand uncertainty," *Journal of Advanced Transportation*, vol. 47, pp. 4–27, 2013.

[28] X. Feng, Z. Xiao, and X. Cui, "Improved RSSI algorithm for wireless sensor networks in 3D," *Journal of Computational Information Systems*, vol. 7, no. 16, pp. 5866–5873, 2011.

[29] F. Valdez and P. Melin, "Neural network optimization with a hybrid evolutionary method that combines particle swarm and genetic algorithms with fuzzy rules," in *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS '08)*, New York, NY, USA, May 2008.

[30] F. Fernandez-Navarro, C. Hervas-Martınez, P. A. Gutierrez, J. M. Pena-Barragan, and F. Lopez-Granados, "Parameter estimation of q-Gaussian. Radial Basis Functions Neural Networks with a Hybrid Algorithm for binary classification," *Neurocomputing*, vol. 75, pp. 123–134, 2012.

[31] H. Soh, S. Lim, T. Zhang et al., "Weighted complex network analysis of travel routes on the Singapore public transportation system," *Physica A*, vol. 389, no. 24, pp. 5852–5863, 2010.

[32] C. K. S. Kumar, R. Sukumar, and M. Nageswari, "Sensors lifetime enhancement techniques in wireless sensor networks—a critical review," *International Journal of Computer Science and Information Technology & Security*, vol. 3, pp. 159–164, 2013.

[33] S. Li and F. Qin, "A dynamic neural network approach for solving nonlinear inequalities defined on a graph and its application to distributed, routing-free, range-free localization of WSNs," *Neurocomputing*, vol. 117, pp. 72–80, 2013.