Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 623720, 18 pages
http://dx.doi.org/10.1155/2015/623720

Research Article

Non-Gaussian Hybrid Transfer Functions: Memorizing Mine Survivability Calculations

Mary Opokua Ansong,[1,2] Jun Steed Huang,[3] Mary Ann Yeboah,[4] Han Dun,[1] and Hongxing Yao[1,5]

[1] *Institute of Systems Engineering, Faculty of Science, Jiangsu University, 301 Xuefu, Zhenjiang 212013, China*
[2] *Department of Computer Science, Faculty of Applied Science, Kumasi Polytechnic, P.O. Box 854, Kumasi, Ghana*
[3] *Computer Science and Technology, Suqian college, Jiangsu University, 399 South Huanghe, 223800, China*
[4] *Department of Mathematics and Statistics, School of Applied Science, Kumasi Polytechnic, P.O. Box 854, Kumasi, Ghana*
[5] *College of Finance and Economics, Jiangsu University, 301 Xuefu, Zhenjiang 212013, China*

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

Received 14 July 2014; Revised 7 November 2014; Accepted 8 November 2014

Academic Editor: Valder Steffen Jr.

Hybrid algorithms and models have received significant interest in recent years and are increasingly used to solve real-world problems. Different from existing methods in radial basis transfer function construction, this study proposes a novel nonlinear-weight hybrid algorithm involving the non-Gaussian type radial basis transfer functions. The speed and simplicity of the non-Gaussian type with the accuracy and simplicity of radial basis function are used to produce fast and accurate on-the-fly model for survivability of emergency mine rescue operations, that is, the survivability under all conditions is precalculated and used to train the neural network. The proposed hybrid uses genetic algorithm as a learning method which performs parameter optimization within an integrated analytic framework, to improve network efficiency. Finally, the network parameters including mean iteration, standard variation, standard deviation, convergent time, and optimized error are evaluated using the mean squared error. The results demonstrate that the hybrid model is able to reduce the computation complexity, increase the robustness and optimize its parameters. This novel hybrid model shows outstanding performance and is competitive over other existing models.

1. Introduction

Hybrid algorithms are used in optimizing real-world implementations that is, it comes as the best optimization solution tends to have challenges in implementation cost, time, and so forth, that needs a solution by using another technique. Hybrid algorithms have received significant interest in recent years and are increasingly use to solve real-world problems. These hybrid algorithms or models include combination of two or more algorithms involving genetic algorithms (GA) [1], particle swarm optimization (PSO) [2], and other computational techniques such as, artificial intelligence or neural networks including but not limited to multilayer perceptrons (MLP) or sigmoid [3], radial basis functions (RBF) [4, 5], fuzzy systems [6] and simulation annealing [7].

An artificial neural networks (ANNs) are techniques of artificial intelligence (AI) that have the capability to learn from experiences, it is robust [8] and improves performance by adapting to the changes in the environment. The underlying advantage(s) of ANNs are the possibility of efficient operation of large amounts of data and its ability to generalize the outcome. This training algorithm, the ANNs, is largely used in applications involving classification or function approximation, and it has been proved that several classes of ANN are universal function approximations [9]. These include radial basis function (RBF) and multilayer perceptrons' (MLPs) neural networks. Taking into consideration the great potential of these techniques, this paper aims to establish a hybrid model using multilayer perceptron network (MLP) also called sigmoid basis function (SBF) and a radial basis function (RBF) network-both feed-forward learning. The RBF and MLP networks are usually employed in the same kind of applications. Examples include the nonlinear mapping approximation and pattern recognition [10]; however their internal calculation structures are different. In the multilayer fully connected feed-forward networks, the nodal transfer function activation flows from the input layer through a hidden layer to the output layer [10]. This functional description process can be expressed as $y_i = f \cdot \sum_{i=1}^{N} w_{ij} x_i + b_j$ [11] with typical processing node, where x_i is one of the N inputs for processing, node j, w_{ij} is the connection weight between node i and node j, b_j is the bias for node j, and y_i is the output from node j. Each neuron in one layer is connected in the forward direction to every nodal unit in the next layer. One disadvantage in using most feed-forward layered neural networks is the high degree of nonlinearity in the parameters. Learning must be based on nonlinear optimization techniques (i.e., back-propagation), and the parameter estimate may become trapped at a local minimum of the selected optimization criterion during the learning procedure.

Another option to such neural networks is to use the radial basis function (RBF) as a transfer function. There is a strong connection between the RBF and neural networks and it is reasonable to believe that a radial basis network (RBN) can offer approximation capabilities similar to other feed-forward, layered neural networks [12], provided that the hidden layer of the RBN is fixed appropriately. This belief is strongly supported by the theoretical results from the RBF method as a multidimensional interpolation technique [13]. A radial basis function neural network has an input, hidden, and output layers. The input layer is composed of an input vector i. The hidden layer consists of RBF activation function as networks neuron. The net input to the RBF activation function is the vector distance between its weight w and the input vector i, multiplied by the bias b. Detailed work has been done on advantages of both sigmoid and radial basis functions [14]. Radial functions are a special class of functions whose value increases or decreases in relation to the distance from a central point. There are different types of radial basis functions, but the most frequently used is the Gaussian function. It is well known that the MLP networks have been applied successfully in several difficult problems. MLP networks also work globally and the network outputs are decided by all the neurons [15]. Radial basis function (RBFs) act as local approximation networks and their outputs are determined by specified hidden units in certain local receptive fields. RBF networks are simpler than MLP networks, in spite of having more complex architectures and respond well to patterns that were not used for training from the point of generalization [15]. Comparing the properties of neural networks, fuzzy inference systems, the RBF had the advantages of easy design, stable and good generalization ability, good tolerance to input noise, and online learning ability. RBF networks are strongly recommended as an efficient and reliable way of designing dynamic systems [15].

An important issue in the RBF neural network applications is the network learning, that is, the need to optimize the adjustable parameters, the center vectors, the variances or the widths of the basis functions, and the linear output weights connecting the RBF hidden nodes to the output nodes and to determine the network structure or the number of RBF nodes [16]. Closely coupled are the determination of the network size and the adjustment of parameters on the continuous parameter space. In this wise evolutionary algorithms have been used to address this problem, nonetheless they are computationally very expensive in its implementation [16] which results in slow and premature convergence and this has attracted attention in literature. The center location and clustering techniques have been proposed [17]. An identical width can be set for all the basis functions if the input samples are uniformly distributed, otherwise a particular width has to be set for each individual basis function to reflect the input distribution [18]. Once the centers and the widths are determined, the linear output weights can be determined using Cholesky factorization, orthogonal least squares, or singular value decomposition [19]. In contrast to the conventional two-stage learning procedure, supervised learning methods aim to optimize all the network parameters [20]. Various techniques have been introduced to improve the network convergence and these include hybrid algorithms to improve the convergence; various techniques combine the gradient-based search for the nonlinear parameters (the widths and centers) of the RBF nodes and the least squares estimation of the linear output weights [18] and combing the merits of fuzzy and crisp clustering [21].

Supervised learning is thought to be superior to conventional two-stage approaches but it can be more demanding computationally. The Akaike information criterion was used when dealing with different network size; however, it is equally computational demanding [21]. With respect to the determination of the RBF neural network structure, a popular approach is to formulate it as a linear-in-the parameters' problem, where all the training patterns/samples are usually used as the candidate RBF centers. To improve the network generalization, the regularized forward selection algorithm has been proposed [22], which combines subset selection with zero-order regularization. Backward selection methods have also been used in RBF center selection [23]. However, forward selection algorithms are thought to be superior to backward methods in terms of computational efficiency, however these methods have several major disadvantages such as being computationally too expensive or sometimes impossible to implement. The search for the optimal values of the nonlinear parameters (RBF centers and widths) is a continuous optimization problem. In order to optimize the RBF center and width parameters along with the network structure determination process, a sparse incremental regression (SIR) modeling method was proposed very recently to determine the network structure and the associated nonlinear parameters simultaneously [24]. This can deal with large dataset and improve the network significantly. Others include the moving k-means clustering to position the RBF centres with givens least squares to estimate the weights [25] and forward algorithm in RBF construction [9], to mention a few.

Different from existing methods in RBF neural network construction and multilayer perceptron, this paper proposes a novel hybrid (HSRF) feed-forward algorithm involving the multilayer perceptron (sigmoid) and non-Gaussian type radial basis transfer functions which is robust and performs parameter optimization within an integrated analytic framework, leading to two main technical advantages.

(1) The network can be significantly improved through the optimization of the nonlinear RBF parameters on the continuous parameter space.

(2) Using the speed of the multilayer perceptron and the simplicity and accuracy of RBF to produce fast and accurate model for rescue operations. In addition the paper uses coded genetic algorithm to train the proposed hybrid algorithm. Finally, network outcomes including mean iteration, standard variation, standard deviation, convergent time, and optimized error are evaluated using 5th order polynomial.

1.1. Problem Statement and Objective. There are generally heavy casualties and tremendous loss of property in the event of accident such as fire, rock fall, flooding, or poisonous gasses as well as lose of human life in the mining sector [26]. This calls for a model that is fast and robust for monitoring and locating survivors to safety in times of accident. The justification of this work is that, the focus of current research is moving from system analysis of small-world networks to that of millions of nodes. This will demand large computers to process, and even if those computers are available, it will demand considerable time to run. This implies that there is the need for fast prediction algorithm using NN to memorize precalculated results to deal with large number of sensors (i.e., as sensors grow so rapidly to thousands and millions that battery drain will not permit calculations on the spot of a problem). In addition the base station can be destroyed in times of accident.

Further justification for a research like this is that the simple imitations of the human brain (called neural network models) demonstrate fast and accurate learning and classification properties in problems that otherwise require human experts. Although such tools cannot obviously replace human experts, they are used as on-the-fly diagnostic tools and supporting evidences in quick decision making. With these in mind the main objective of this study is to investigate and improve upon the Gaussian radial basis function and develop a non-Gaussian hybrid of MLP (sigmoid or SBF) and the compact radial basis functions (CRBF) with enhanced optimization features. From this an optimized hybrid model is assessed that has the highest predicted survival probability for an emergency rescue operation in an underground mining with genetic algorithm. The two main objectives examined in this paper are as follows.

(i) To investigate the radial basis function of Gaussian model and remove the additional computation burden on the model by paralyzing the power operation on Gaussian model to generate a compact radial basis functions (non-Gaussian) literally but novel to reduce computational cost and increase processing efficiency. The study focuses on the use of absolute operation instead of square operation. In Figure 1 the green outline (for online viewing) represents additional requirement of resources in terms of time, cost, and so forth, assuming the resource is proportional to the calculated value, that is, zeros are not stored.

(ii) To develop an optimized hybrid neural network called HSRF model with nonlinear weights of negative cosine, imposed on new compact radial basis function. This nonlinear weight was introduce to further reduce the RBF magnitude or element in the model for accuracy and maintained speed.

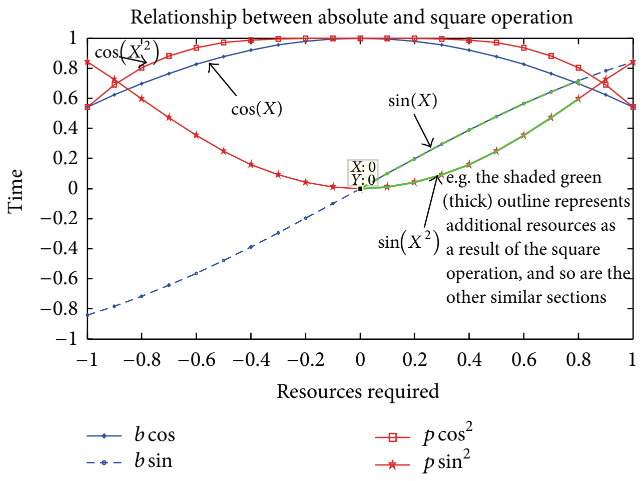


FIGURE 1: The effect of using power operation instead of absolute operation.

2. Preliminary Consideration

2.1. Deployment of Sensors and Connection. Topological deployment of sensor nodes affects the performance of the routing protocol [27]. The ratio of communication range to sensing range as well as the distance between sensor nodes can affect the network topology. To get the right deployment, the following are essential.

(i) Initialise the mean iteration (MI), standard deviation (Std var), standard deviation (SD), average convergent time (CT), epoch, and gbest-fitness.

(ii) State the relative rock hardness parameters, that is, hardness or softness of rock is chosen to be in the range 0.7 and 0.9 and 6 values are chosen to reflect 6 types of rocks obtained from the industry in Table 1.

(iii) Initialise matrix index as 0.

(iv) State the 3 dimensions of the mine or field: depth (D), length (L), and width (W).

(v) T is total number of nodes deployed in 3D view ($T = D * L * W$).

The sensor sequence location explains the shortest search path criteria of the robot. The robot's job is to make sure

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 m	78 m

Data was collected from Wang Xing village, located in Xinzhen City, Henan Province, China.
The table identifies 6 common rocks found in mines in relation to hardness or softness of each rock.

```
Initialize the following
For $i = 1$: Depth
  For $j = 1$: Length
    For $k = 1$: Width
      $t = t + 1$; % the index of the $T$ by $T$ matrix later on
      $(t, 1) = i$; % used to find the Depth/level.
      $\text{tog}J = \text{ceil}(t/\text{Width}/\text{Length})$;
node$(t, 2) = \text{abs}\left(-(\text{Depth} + 1) * (1 + (-1)^{\text{tog}J})\right)/(2 + j)$. % used to find the row.
  $\text{tog}K = \text{ceil}(t/\text{Width})$,
node$(t, 3) = \text{abs}\left(-(\text{Width} + 1) * (1 + (-1)^{\text{tog}K})\right)/(2 + k)$. % used to find the column.
    End
  End
End.
```

Algorithm 1

every sensor is working. It moves to the first level, searches the row(s) and the respective column(s) and proceeds to the next level to repeat the procedure.

The sensor sequence position is expressed in Matlab code as shown in Algorithm 1.

2.2. Communication and Transmission Range. The Through-the-Earth (TTE) communication system transmits voice and data through solid earth, rocks, and concrete and is suitable for challenging underground environments such as mines, tunnels, and subways [28, 29]. There are stationary sensor nodes as well as mobile sensors (humans and vehicles) distributed uniformly. The survivability of the mobile sensor is the main concern of this paper; we need to predict the survivability based on the rock type and location that the sensor sees, using the NN which is trained ahead of time. Both stationary and mobile sensor nodes are connected to either the access point (AP) and/or access point heads (AP Heads) based on transmission range requirements. The AP heads serve as cluster leaders and are located in areas where the rock is relatively soft (large size resting and eating room) or has relatively better signal penetration to ensure optimum transmission. The TTE is dropped through a drilled hole on top of the large size room or bay, approximately 300 metres beneath the ground 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 equipment to the miners underground and return with underground information to rescue team. Minimizing the transmission range of wireless sensor networks is vital to the efficient routing of the network. This is because the amount of communication energy that each sensor consumes is highly related to its transmission range or signal reach [30]. The node signal reach N is defined as the absolute difference of the minimum signal reach and maximum signal reach of nodes plus the minimum signal reach, taking into consideration the 6 cases of the rock structure β, where β lies between the soft-rock 0.7 and the hardest rock 0.9. Routing is also limited to the load of the nodes and the distance each node travels [31]. The minimum and maximum signal reach of nodes ($N_{\min}$ and $N_{\max}$) and N are calculated in Matlab code below.

Initialize the connection matrix

Mc = zeros (T, T); rock = rock cases

$$N_{\min} = \min(\text{Depth}, \min(\text{Length}, \text{Wight})),$$
$$N_{\max} = \max(\text{Depth}, \max(\text{Length}, \text{Width})), \tag{1}$$

for $i = 1 : T$, and for $j = 1 : T$

$$N = N_{\min} + |N_{\min} - N_{\max}| * \text{a random number from hardness of rock.} \tag{2}$$

The relationship between rock hardness/softness (β) and the signal reach is a complicated nonlinear function. It 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 [32] as follows: Skin depth $= \sqrt{2/(\rho * \omega * \sigma)}$, where ρ is material conductivity, ω is frequency, σ is magnetic permeability, the signal (B-field) is attenuated by cube of distance expressed as: signal reach (distance) $= 3 *$ skin depth [33]. The common rocks identified for this work are presented in Table 1.

To depict how the deployed sensor nodes were connected, the connection matrix is introduced before transmission of data can be made. Similarly, to depict the path in which data is transferred from source to destination and vice versa, the routing matrix is introduced. Let K denote the connection matrix, a sensor node is named by its 3D integer (x, y, z) coordinates, where $1 \leq x \leq$ length, $1 \leq y \leq$ width, $1 \leq z \leq$ depth and $T =$ Legth $\cdot$ Width $\cdot$ Depth is the total number of nodes. Node (a, b, c) is connected with node (d, e, f), if the element on $((a-1)\cdot C\cdot L+(b-1)\cdot L+c)$th row and $((d-1)\cdot C\cdot L+(e-1)\cdot L+f)$th column is 1, otherwise 0; therefore

$$K_{i,j} = \begin{cases} 1, & \|i-j\| \leq N \\ 0, & \|i-j\| \geq N \text{ for } i, j = 1, 2, \ldots, T, \end{cases} \tag{3}$$

where N is the node signal reach. The routing matrix $\mathfrak{R}$ is limited to total multiple points' connections to be made. In arriving at the final optimized vector for transmission, each matrix was generated τ times, where τ is the number of cases it iterates before producing the final survival rate vector (R), M_ρ is point-to-multi-point connection (hub or switch) are less or equal 4 and is an even number allowing bidirectional communication, thus, between the source and destination of nodes. Consider

$$\mathfrak{R}_{i,j} = \begin{cases} 1, & \|i-j\| \geq \dfrac{M_\rho}{2}, \\ 0, & \|i-j\| \leq \dfrac{M_\rho}{2}, \text{ for } i, j = 1, 2, \ldots, T, \end{cases} \tag{4}$$

where i, j are the source and destination nodes respectively.

2.3. Fault Tolerant Considerations. Fault tolerant considerations for hardware, software, and network security are critical areas in emergency situations as they significantly affect the efficiency of the communication, and many key management schemes have been proposed to mitigate the constraints [34, 35]. In an event of accident, let ψ be the probability of accident occurring, then routing path or matrix $\mathfrak{R}$ would be affected by $(1-\psi)$, where ψ is any random value within β (i.e., relative rock hardness or softness), that would cause explosion. Let X be explosion matrix whose elements depict the level of damage to the sensor nodes or routing path caused by the explosion and X is defined in the following:

$$X = (1-\psi)\,\mathfrak{R}. \tag{5}$$

The damage of explosion will consequently result in the failure of the routing path and matrix F is introduced to depict the signal status of the sensor nodes which is used as new generated connection matrix. The matrix F is defined as

$$F_{i,j} = \begin{cases} 1, & X_{i,j} \leq \lambda_L \\ 0, & X_{i,j} \geq \lambda_H \\ \dfrac{\lambda_L}{X_{i,j}}, & \lambda_L < X_{i,j} < \lambda_H, \end{cases} \tag{6}$$

where λ_L and λ_H represent the lower and higher accident impact thresholds, respectively.

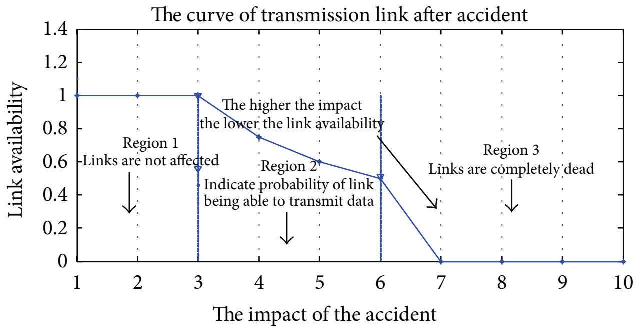


Figure 2: Impact of explosion/accident on transmission link.

For example, $X_{i,j} \leq \lambda_L$ implies that the number of nodes will be 0, 1, 2, and 3; when $X_{i,j} \geq \lambda_H$ the number of nodes is expected to be 7, 8, and above. Whiles $\lambda_L < X_{i,j} < \lambda_H$ is expected to be 4, 5, and 6. Figure 2 explains the effect of the explosion on the transmission link in three regions. Region 1 of the figure implied the link(s) were not affected. Nodes with numbers 4, 5, and 6 in region 2 represent a probability that the links will be able to transmit/receive data while nodes above number 7 in region 3 signal dead links (Figure 2).

To find the matrix with the maximum survival probability, a new set of routing path is defined after the damage. To optimize transmission after damage caused by explosion, the hope matrix was introduced whose elements indicate status of hope of signals to reorganize the routing path. To describe the success rate of signal from each node to the sink(s), the exit matrix is introduced. In most practical applications, more than one sink is used, and sink's node is either through the fiber or through the earth (TTE).

Let H be the hope matrix, and let E be the exit matrix. Then H and E are, respectively, defined as

$$\begin{aligned} H &= F \cdot \mathfrak{R}, \\ E &= (n_e - \psi_2)\,H, \end{aligned} \tag{7}$$

where matrix $\mathfrak{R}$ and F are respectively given in (4) and (6), n_e is the total number of safe exits available for use, and ψ_2 is the probability of additional error that miners may commit in trying to escape danger.

2.4. Hardware and Software Considerations. In real rescue situations, software and hardware, including radio frequency identification (RFID) [36], can fail as a result of accidents or explosion which can significantly affect the routing path and thwart the efforts of the rescue team. Equally, miners as well as personnel can make other mistakes in the face of accident that can compound the existing problem(s), especially where miners find themselves more than 4,000 feet underground, as it is in the case of one of the mine-fields used for this study. In addition personnel may also fail to cope with the stress that comes with accident. It is therefore imperative to consider such failures in developing rescue models. Let S represent the software survival rate matrix. A matrix S is used to describe software or relational database management systems (RBDMS) survival rate including effects from bugs or attacks. Let $S_{i,j}$ represent elements of matrix S in row i and column j, then S is defined as

$$S_{i,j} = 1 - \left(\frac{1}{T + M_{i,j}}\right), \tag{8}$$

where T is the total number of nodes deployed and $M_{i,j}$ is the element of matrix of the random hardness of rock that is generated as a T by T matrix according to the geometric distribution. The maximum probability of survivability to rescue miners is defined by vector R. To obtain R, it is assumed that each miner will have an RFID. To describe probability of the survival rate, including risks of running out of battery, we introduce the row vector α. Let α_i represent element of α (i.e., survival rate of each miner to be rescued) in column i, then α_i is defined in the following:

$$\alpha_i = 1 - \left(\frac{1}{T + \gamma_i}\right), \quad i = 1, 2, \ldots, T, \tag{9}$$

where the γ_i is ith element of a vector of the random hardness of rock generated according to the geometric distribution. To measure the hardware survival rate of miners to be rescued, the vector δ is introduced, and δ is closely related to the exit matrix and is defined in the following:

$$\delta_i = \min\left\{1, \left(\alpha \cdot \frac{E}{\rho_m}\right)_i\right\}, \quad i = 1, 2, \ldots, T, \tag{10}$$

where δ_i and $(\alpha \cdot (E/\rho_m))_i$ were, respectively, ith element of row vectors δ and $\alpha \cdot (E/\rho_m)$, E is the exit matrix given by expression (7), and ρ_m is the total number of hubs used. To ensure that a reliable system is in place for emergencies, the final survival rate vector is (R) introduced by the following equation:

$$R = \mu\left(\frac{1}{T}\sum_{i=1}^{T}\sum_{j=1}^{T} S_{i,j}\right)\delta, \tag{11}$$

where μ is the mean and the vector R represents the highest survival probability to ensure miners safety.

3. Related Work

3.1. MLP (Sigmoid Basis Function). A sigmoid basis function (SBF) is a mathematical function having an "S" shape (sigmoid curve) and is related to brain reasoning and the structure favours the computational believers. Often, sigmoid function refers to the special case of the logistic function. It is used in modeling systems that saturate at large input values, for example, the ogee curve as used in the spillway of some dams [37]. Wide varieties 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 [37]. The generalized function and the sigmoid output function [10] can be expressed, respectively, as

$$\begin{gathered} u = \sum_{i=1}^{n} w_i x_i, \\ y = f(u) = \frac{1}{(1 + e^{-cu})}. \end{gathered} \tag{12}$$

In the above formula, the initial inputs $x_1, x_2, \ldots, x_n$ are summed together where $w_1, w_2, \ldots, w_n$ are weights of the neurons for the ith input, u is the activation level scaled according to the output function $f(u)$, giving the actual output y of the neuron. A positive constant c controls the slope or steepness of the sigmoid; the sigmoid function amplifies the small activation levels and limits the high activation levels. In practice; the output y lies between 0 and 1 [38] however, outputs requiring negative values use the hyperbolic tangent as expressed below:

$$\begin{gathered} y = f(u) = \tanh(cu), \\ \text{or} \quad f(u) = \frac{e^{cu} - e^{-cu}}{e^{cu} + e^{-cu}}. \end{gathered} \tag{13}$$

The initial stage of the sigmoid basis function (SBF) grows relatively exponential as x touches the zero (0) line, and then the growth begins to slow down towards saturation and stops at maturity as x goes to 1.

3.2. Gaussian RBF. The use of RBF kernels, mainly Gaussian and its global acceptance into various applications, cannot be overemphasized. However the model carries additional power computational burden that is translated into cost. The objective is to explore and remove this power computation burden and apply it in an emergency rescue system. The section discusses many related works to the proposed approach; this includes the Gaussian radial basis function (GRBF) neural network; radial basis function neural network consists of the input layer, the hidden layer, and the output layer. The inputs of hidden layer are the linear combinations of scalar weights and input vector, where the scalar weights are usually assigned as unit values, that is, the whole input vector appears to each neuron in the hidden layer. The incoming vectors are being mapped by the radial basis functions in each hidden node. The output layer yields a vector by linear combination of the outputs of the hidden nodes to produce the final output [39]. The structure of an n inputs and m outputs RBFNN is depicted as

$$y = f_j(u) = \sum_{k=1}^{g} w_j^k \varphi_k(u), \quad \text{for } j = 1, \ldots, m, \tag{14}$$

where $u = \{u_1, u_2, \ldots, u_n\}$ denotes the input vector for n inputs and $y = \{y_1, y_2, \ldots, y_m\}$ denotes the output vector for m outputs, w_j^k denotes the weight of the kth hidden nodes and the jth output node, and g is the total number of hidden nodes, $\varphi_k(\cdot)$ denotes the radial basis function of the kth hidden node. The final output of the jth output node

$f_j(u)$, is the linear combination of all hidden nodes. Using the summation as the denominator, expression (14) can be normalized as

$$y = f_j(u) = \frac{\sum_{k=1}^{n} w_j^k \varphi_k(u)}{\sum_{k=1}^{g} \varphi_k(u)}, \quad \text{for } j = 1, \ldots, m. \tag{15}$$

A multidimensional function RBF describing the distance between a given input vector and a predefined center vector is given as

$$\varphi_k(u) = \exp\left(-\frac{\|u-\mu\|^2}{2\sigma_k^2}\right) \quad \text{for } k = 1, \ldots, g. \tag{16}$$

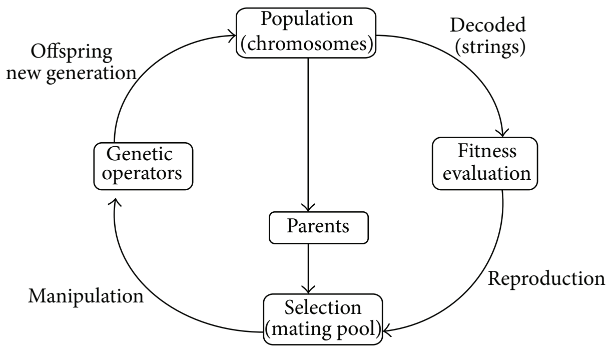


Figure 3: The mutation, crossover, and reproductive cycle of GA.

4. Training Methods

In a genetic algorithm (GA), a populace of candidate solutions (individuals or creatures), to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties (its chromosomes or genotype) which can be mutated and transformed. Traditionally solutions are represented in binary as strings of "0s" and "1s" [40]. The development usually starts from a population of arbitrarily generated individuals and is an iterative process, with the population in each iteration called a generation [41]. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation [42, 43]. The new generation of candidate solutions is then used in the next iteration of the algorithm. Generally, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

A typical genetic algorithm requires a genetic representation of the solution domain and a fitness function to evaluate the solution domain. A standard representation of each candidate solution is as an array of bits. Arrays of other types and structures can be used in essentially the same way as used in Hasan [44]. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming, that is, a mix of both linear chromosomes and trees is explored in gene expression programming [45]. Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion, and selection operators. Parent chromosomes are selected with a probability related to their fitness. The chromosomes with high fitness have higher probability to be selected for mating than chromosomes with less fitness. The mutation, crossover, and reproductive chart is displayed by Wright [46] in Figure 3. The GA operates in a very simple cycle of stages outlined below, creation of a population of strings, evaluation of string, selection of best string, and genetic manipulation to create new population of strings. Each cycle of GA produces a new generation of possible solution for a given problem. This describes the representatives of potential solution to initiate the search process. The population elements are encoded into bits-string called chromosomes. The performance or fitness of each string is evaluated. Based on the fitness of chromosomes, selection is made for the next genetic manipulation. Selection is mainly responsible for survival of the best-fit individuals and ends the population string. Here genetic manipulation processes consisting of two steps are carried out. This includes the crossover operation that combines the bits (genes) of each two selected strings (chromosomes) to be executed.

However, there are several limitations of the use of a genetic algorithm compared to alternative optimization algorithms. Repeated fitness function evaluation for complex problems is often the most prohibitive and limiting segment of artificial evolutionary algorithms. Finding the optimal solution to complex high dimensional, multimodal problems often requires very expensive fitness function evaluations. In real world problems such as structural optimization problems, one single function evaluation may require several hours to several days of complete simulation. Typical optimization methods cannot deal with such types of problem. In this case, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. It is apparent that amalgamation of approximate models may be one of the most promising approaches to realistically use GA to explain complex real life problems. Genetic algorithms do not scale well with complexity. That is, when the number of elements which are exposed to mutation is large, then there is often an exponential increase in search space size. This makes it extremely difficult to use the technique on problems such as designing an engine, a house, or plane. In order to make such problems tractable to evolutionary search, they must be broken down into the simplest representation possible. The second problem of complexity is the issue of how to protect parts that have evolved to represent good solutions from further destructive mutation, particularly when their fitness assessment requires them to combine well with other parts. The best solution is only in comparison to other solutions.

As a result, the stop criterion is not clear in every problem. In many problems, GAs may have a tendency to converge towards local optima or even arbitrary points rather than the global optimum of the problem [42]. Various types of crossover operators are used such as the single-point and two-point crossover operators among others [47].

In this paper the coded genetic algorithm was used to further optimize the parameters.

Research has shown that a small population size with relatively large mutation rate is far superior to large population sizes and low mutation rates [48]. The paper argued that the best mutation rate for GAs falls between 5% and 20% while the population size should be less than 16. However in choosing the population size that optimizes the GA, a number of options were considered. The total input elements chosen is 300 which corresponds to the matrix size or the dimensions of the field space giving by depth (D), length (L), and width (W) that is 10, 6, and 5 respectively. To balance the speed and accuracy we took the square root of the total element of the matrix size to obtain the optimum size of 17.32. Guided by this figure a population size of 20 is used. In addition 10% of the matrix size (i.e., 30) is also used as a basis for comparison.

5. Proposed Hybrid Non-Gaussian Model Based on Compact Radial Basis Function (CRBF)

As already stated, this proposed novel hybrid involving a special mix of the multilayer perceptron and radial basis transfer functions is an empirical study to initiate and determine such guidelines (e.g., how much of the MLP or RBF should be included and vice versa) of mixing two major transfer functions. It will also examine several parameter combinations to find the one that will give the right mix to keep the error at minimum, maintain speed and accuracy, and optimize various parameters well, in one engineering design problem. The Gaussian-like kernel was adapted because it has gained global acceptance. The paper paralyzed the power parameter arbitrary but novelly as expressed below:

$$k_P\left(x, x'\right) = \exp\left(\frac{-\left\|x - x'\right\|}{\delta^2}\right), \quad (17)$$

which is in the form $k_P(x, x') = \exp(-\text{abs}(x - x'))$, $x, x' \in \mathbb{R}^n$ and $k_P(x, x') = \log\text{sig}(x, x')$, $x, x' \in \mathbb{R}^n$, respectively. The function MLP and RBF are given as

$$\begin{aligned} y_1 &= \log\text{sig}(R), \\ y_2 &= \exp(-\text{abs}(R)), \end{aligned} \quad (18)$$

where $\log\text{sig}(R)$ is the function for multilayer perceptron or sigmoid (SBF) and $\exp(-\text{abs}(R))$ is the function for RBF, respectively (in Matlab code). The R is the optimum set of the routing table with the maximum survival probability which is used as an input for the neural network; y_1 and y_2 are the transfer functions for the new sigmoid and compact radial basis functions, respectively. The hybrids of the transfer function are summarized; with a weight of $-\cos(R)$ on CRBF, the $\text{HSRF}_{-\cos}$ is given as

$$\text{HSRF}_{-\cos} = \log\text{sig}(R) + \left[\exp(-\|R\|)\right] * \left[-\cos(R)\right]. \quad (19)$$

The study also investigated a parameter between the Gaussian models (GRBF) and proposed compact radial function (CRBF) model (with a power of 2 and 1 resp.), for simplicity we referred to it as a reduced parameter (ZRBF)

$$\text{ZRBF} = \exp(-\|R\|)^{1.5}. \quad (20)$$

Finally a hybrid of Gaussian RBF using our model is also investigated as follows:

$$\text{HSGF}_{-\cos} = \log\text{sig}(R) + \left[\exp(-\|R\|)^2\right] * \left[-\cos(R)\right]. \quad (21)$$

The GA-neural network (GA-NN) is used to train the neurons such that the initial error will be minimized and make the model more reliable [49]. Now it must be noted that the MLP is also referred to as SBF is a complete function and so is the RBF, and therefore adding these will be an abuse of usage of the transfer functions. Other options include considering 50% of each transfer function and considering some percentages of each as well. The conceptual view of the model can be seen in at the top section of the semantic structure in Figure 4. To examine the position of each neuron, let PN denote the position of the nth genome, then PN is expressed as follows:

$$\text{PN} = \sum_{i=1}^{T} R_i S_i + \sum_{i=1}^{2} S_i + \sum_{i=1}^{2} S_i S_i + 1, \quad (22)$$

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

The initial population is made up of randomly determined parameters within specified boundaries. These parameters are called the genes of the chromosome or the genes of an individual [50, 51]. Assuming a population P of three random individuals with genes that represent the values of three design variables, the selection of an individual is done by evaluating the performance of each one and then ranking them from best to worst. Finally an individual is chosen for production [52]. The performance of the individuals, in this case, is the difference between the measured (or desired) value of the spectrum and the calculated value. This is measured by using the mean square error. Generally there are three methods to diagnose the fitness and these include the mean cubic error (MCE), the mean absolute error (MAE), and the mean squared error (MSE). The mean cubic error (MCE) will allow for fast convergence at the expense of accuracy, making the process unstable. Whiles the mean absolute error (MAE) is stable but converges slowly [53]. A midway between these two is the MSE. Therefore the goodness of the fit in this study is diagnosed using mean squared error (MSE) as against the other two. In addition other parameters including mean iteration, standard

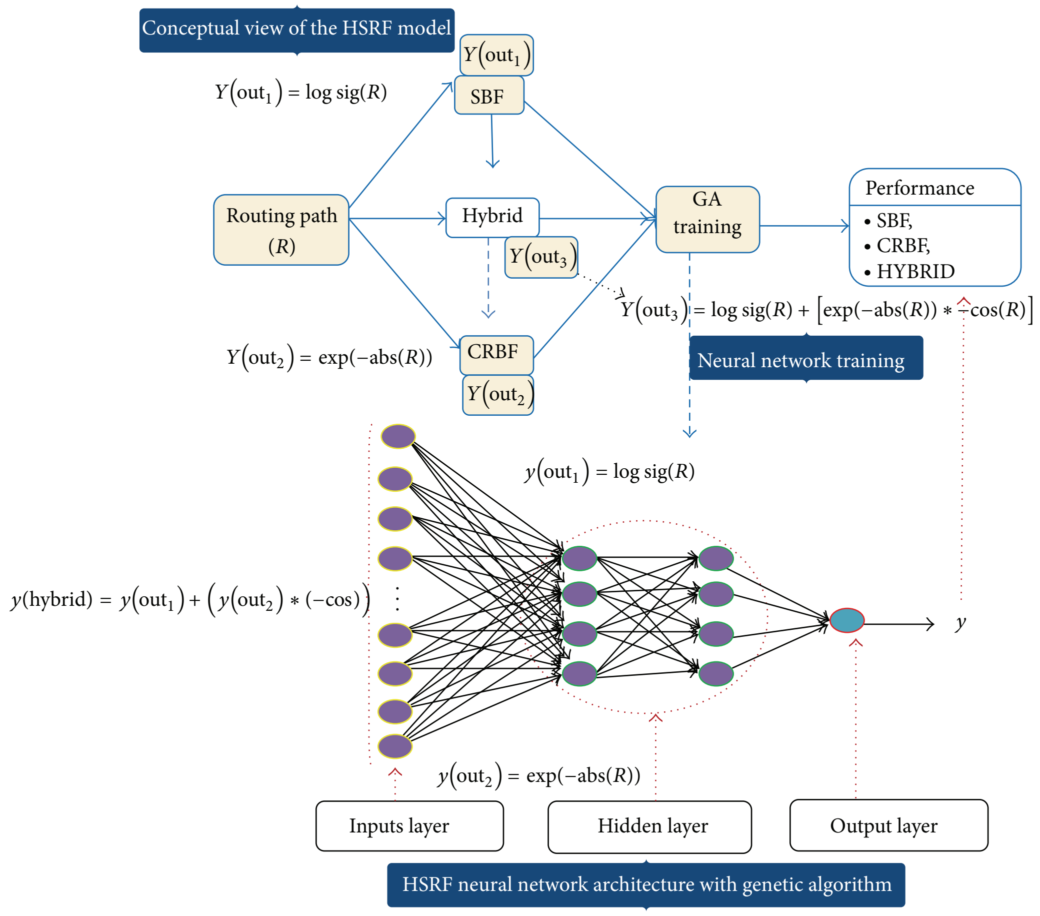


FIGURE 4: The conceptual view with GA semantic structure of the proposed hybrid neural network.

variation, standard deviation, and convergent time are also evaluated. The MSE is given as

$$\mathrm{MSE} = \frac{1}{ns}\sum_{j=1}^{n}\sum_{i=1}^{s}\left(Y_{j,i} - y_{j,i}\right)^2, \tag{23}$$

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

The individual with the best performance receives a relative weight (RW) of (τ^{a}), where τ is the number of individuals in the population and a is the weight that is usually between 1.0 and 1.5 [51]. The next best individual receives a RW of $(\tau - 1)^{a}$, and so forth, until the worst individual receives a RW of 1. The probability of reproduction rate of each individual is given as $\mathrm{PR}_j = \mathrm{RW}_j/(1/\tau)\sum \mathrm{RW}$, where PR_j is the probability of reproduction for individual j, and $j = 0, 1, \ldots, \tau$. Individuals selected are made to reproduce. Reproduction involves randomly selecting two parents to form the reproduction pool to cross and create offspring. The performances of each offspring are evaluated and the best offspring selected. Mutation permits the introduction of extra variability into the population. Here the study permits 20% of those with worse performance to mutate for faster convergence; greater percentage will equally present slow convergence. The semantic structure of the proposed hybrid neural network with GA is presented in Figure 4.

The performance of the model is measured using polynomial or curvilinear trendline to examine the contributions of the whole models as well as the model parameters. Trendlines also called regression analysis estimates the relationship between variables so that a given variable can be predicted from one or more other variables. By using regression analysis, you can extend a trend line in a chart beyond the actual data to predict future values. A polynomial trendline is used for this work and the closer the R-square value is to 1, the trend is better, that is, if R-squared value is 0.979, then the corresponding trend is a good fit of the line to the data. A polynomial or curvilinear trendline is given by $p(x) = c_0 + c_1x + c_2x^2 + c_3x^3 + \cdots + c_6x^6$ to calculate the least squares fit through points, where $c_0, c_1, c_2, \ldots, c_6$ are constants. The 5th order polynomial is used to examine the performance among the individual parameters in each model because it has a higher predictive power over the 3rd and 2nd orders and it is *the best fit for the proposed model, especially* where the environments are highly controlled and observations are made to a specified level of tolerance. In addition the 5th order gives a more reliable result than the 2nd and 3rd degrees. The expression of the 5th order is displayed as

$$Y_r = \beta_5 X_1^5 + \beta_4 X_1^4 + \beta_3 X_1^3 + \beta_2 X_1^2 + \beta_1 X_1 + \beta_0, \qquad (24)$$

where X_1 is time (seconds) and β_i is the coefficient of the polynomial, $i = 0, 1, \ldots, 5$.

6. Simulation Results and Discussion

The simulation results are based on the following: the total nodes used for the simulation is 300 with underground mine dimensions of $L = 10$, $R = 6$ and $C = 5$ for depth (level), length (row), and width (column) respectively. The intervals between sensors in all directions are 100 meters. Multipoint connection (hubs/switches) used is 4. It is assumed that 2 exits are safe for miners to escape after accident. The GA training used population size of 20. The thresholds λL and λH are 3 and 6, respectively. Six cases of routing paths R are estimated base on relative rock hardness/softness between 0.7 and 0.9. The number of neurons is 6.

It must be noted that in presenting the matrices only 6 nodes are used $L = 3$, $R = 2$, and $C = 1$ for the dimensions.

After simulation the sensor location sequence S_{eq} is generated for the three rows represent the depth (level), length (row), and width (column), respectively, in underground mine space, as indicated earlier this explains the shortest search path criteria of the robot. The following matrices and vectors are also generated and represent the connection K and routing $\mathfrak{R}$ matrices. Element "1" in the connection matrix and routing matrix represent signal reach and routing (transmission links), whiles "0" means there is no signal reach or transmission link is down. Consider

$$S_{\text{eq}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \\ 3 & 2 & 1 \end{pmatrix},$$

$$K = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}, \qquad (25)$$

$$\mathfrak{R} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}.$$

Elements in the explosion matrix X show the effect as a results of accident or explosion. If the number of a node is 7 and above, it implies that the link is totally down, 3 and below means the link is good and has 100% assurance that transmission can go on, and 4, 5, and 6 represent probabilities of the link(s) being able to transmit. Equally in the failed matrix F, the nodes "0," "1" and fractions represent the link is down ("0"), good ("1") and the fraction, for example, 0.75, represent the probability for transmission. Consider

$$X = \begin{pmatrix} 1 & 5 & 1 & 2 & 2 & 9 \\ 0 & 1 & 6 & 1 & 0 & 0 \\ 4 & 6 & 2 & 2 & 2 & 0 \\ 2 & 1 & 0 & 6 & 0 & 3 \\ 1 & 0 & 1 & 5 & 1 & 1 \\ 4 & 0 & 3 & 1 & 2 & 3 \end{pmatrix},$$

$$F = \begin{pmatrix} 1 & .6 & 1 & 1 & 1 & 0 \\ 1 & 1 & .5 & 1 & 1 & 1 \\ .75 & .5 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & .5 & 1 & 1 \\ 1 & 1 & 1 & .6 & 1 & 1 \\ .75 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}. \qquad (26)$$

As discussed already in Figure 2, the nodes represent the effect of the explosion in transmission of data and the number of nodes 0, 1, 2, and 3 in Region 1 of the figure implied the link(s) are not affected. Nodes with numbers 4, 5, and 6 in region 2 represent a probability that the links will be able to transmit/receive data while nodes with number 7 and above in region 3 mean the links are totally dead as indicated on the impact of explosion/accident on transmission link curve. The remnant matrix called the hope matrix (H) is optimized for transmission. The nodes on the hope matrix indicate hope to reorganize the routing path to get the optimized matrix (O) or a new routing path with the elements 0.4, 0.48, and "0.8" strongly connected as the probability to be able to transmit and receive data whiles "0" is not able to transmit or receive data. Consider

$$H = \begin{pmatrix} 1 & .6 & 1 & 0 & 0 & 0 \\ 1 & 1 & .5 & 1 & 0 & 0 \\ 0 & .5 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & .5 & 1 & 0 \\ 0 & 0 & 0 & .6 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix},$$

$$O = \begin{pmatrix} .8 & .4 & .8 & 0 & 0 & 0 \\ .8 & .8 & .4 & .8 & 0 & 0 \\ 0 & .4 & .8 & .8 & 0 & 0 \\ 0 & .8 & .8 & .4 & .8 & 0 \\ 0 & 0 & 0 & .48 & .8 & .8 \\ 0 & 0 & 0 & 0 & .8 & .8 \end{pmatrix}. \qquad (27)$$

In the exit matrix E the resulting safe routes are 0.72, 0.864, and 1.44 values whiles the values "0s" represent unsafe routes. The values in the rest of the vectors infrared α, hardware δ, and software matrices S represent the probability of successes of survival of the infrared, hardware, and software, respectively; that is, the elements in the hardware survival

Table 2: Parameters for SBF/CRBF Hybrid.

Parameter	MI	SD VAR	SD	CONV. TIME	ERROR	CPU
1	113.1	76.53	0.008482	75.00	0.0116	228
2	112.2	75.71	0.008769	60.00	0.0104	280
3	112.5	76.10	0.009036	60.00	0.0121	244
4	113.5	74.38	0.008666	80.00	0.0121	250
5	115.50	72.72	0.009078	88.00	0.0133	180
6	115.50	77.17	0.008584	80.00	0.0104	150
7	121.40	76.63	0.008244	85.00	0.0103	238
8	116.60	77.31	0.008806	80.00	0.0103	260
9	113.70	76.09	0.008485	75.00	0.0102	252
10	112.30	75.60	0.008300	100.00	0.0102	100
Total	**579.5**	**758.24**	**0.08645**	**783.00**	**0.110903**	**1,954**
Avg	114.63	75.824	0.008645	78.30	0.01109	217.1111

rate δ mean the probability that the hardware will not fail. Consider

$$E = \begin{pmatrix} 1.44 & 0.864 & 1.44 & 0 & 0 & 0 \\ 1.44 & 1.44 & 0.72 & 1.44 & 0 & 0 \\ 0 & 0.72 & 1.44 & 1.44 & 0 & 0 \\ 0 & 1.44 & 1.44 & 0.72 & 1.44 & 0 \\ 0 & 0 & 0 & 0.864 & 1.44 & 1.44 \\ 0 & 0 & 0 & 0 & 1.44 & 1.44 \end{pmatrix},$$

$$S = \begin{pmatrix} 0.8571 & 0.9167 & 0.875 & 0.8889 & 0.8333 & 8333 \\ 0.8889 & 0.9545 & 0.8571 & 0.8333 & 0.8333 & 0.8333 \\ 0.8333 & 0.8333 & 0.875 & 0.875 & 0.8333 & 0.8571 \\ 0.8333 & 0.8333 & 0.871 & 0.875 & 0.8333 & 0.8571 \\ 0.875 & 0.8333 & 0.9231 & 0.8571 & 0.8889 & 0.9 \\ 0.8333 & 0.8571 & 0.8571 & 0.8571 & 0.8889 & 0.9231 \end{pmatrix},$$

$$\alpha = \begin{pmatrix} 0.8333 & 0.9167 & 0.9333 & 0.8571 & 0.9412 & 0.8571 \end{pmatrix},$$

$$\delta = \begin{pmatrix} 0.63 & 0.9866 & 1 & 1 & 0.956 & 0.6474 \end{pmatrix},$$

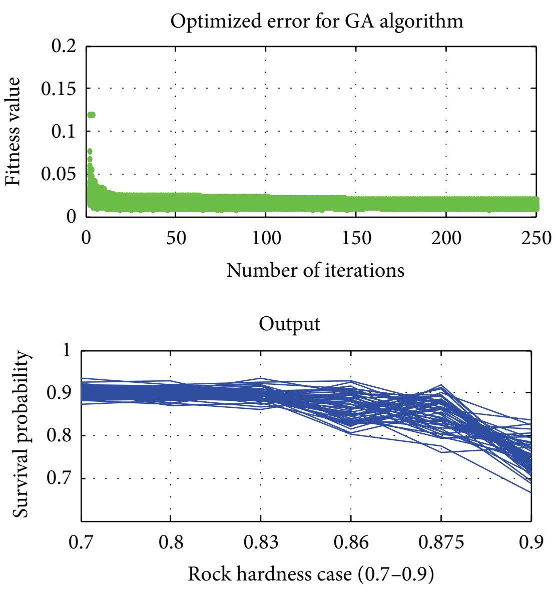
$$R_i = \begin{pmatrix} 0.5377 & 0.8597 & 0.8741 & 0.8644 & 0.8143 & 0.56157 \end{pmatrix},$$

$$\phi R_{n.\text{cases}} = \begin{pmatrix} 0.5377 & 0.8597 & 0.8741 & 0.8644 & 0.8143 & 0.5615 \\ 0.1706 & 0.4122 & 0.4628 & 0.6959 & 0.5594 & 0.1673 \\ 0.5450 & 1.8966 & 0.7650 & 0.7483 & 0.5712 & 0.2839 \\ 0.5579 & 0.8173 & 0.1457 & 0.4733 & 0.6451 & 0.2869 \\ 0.3076 & 0.5930 & 0.7958 & 0.5116 & 0 & 0.2988 \\ 0.5407 & 0.5026 & 0.8445 & 0.1875 & 0.7910 & 0 \end{pmatrix}. \tag{28}$$

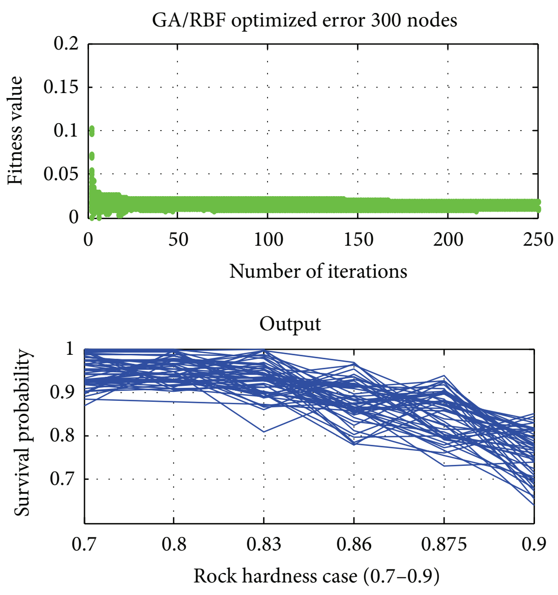
As noted already each iteration in generating the routing path is 6, representing the six (6) types of rock identified for the study. The R is the mean or average of all the 6 cases $\phi R_{n.\text{cases}}$ which is used as inputs to the neural network. The nodes $R_i = (0.5377\ \ 0.8597\ \ 0.8741\ \ 0.8644\ \ 0.8143\ \ 0.5615)$ represent the maximum survival probability for a total of 6 nodes deployed, each row of the $\phi R_{n.\text{cases}}$ matrix represents a vector at each iteration (i.e., The elements $R_i = (0.5377\ \ 0.8597\ \ 0.8741\ \ 0.8644\ \ 0.8143\ \ 0.5615)$ represent the probability of 54%, 86%, 87%, 86%, 81%, and 56% success of each node transmitting data to and from its source and destination resp.). 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, in this study TTE is used. The size of the vector depends on the dimensions of the field. It assists decision makers as to whether to send data through one or more nodes or send each message twice.

Figures 5 and 6 represent the training results of both basic and hybrid models. GA-SBF (Figure 5(a)), GA-CRBF (Figure 5(b)), GA-Gaussian (Figure 5(c)), and GA-ZRBF (Figure 5(d)) for the basic models and Figures 6(a) and 6(b) for the two hybrids of CRBF and Gaussian. Figures 6 and 7 indeed represent the MSE convergence curves from the simulation results for the proposed hybrid and the Gaussian hybrid. This is so because the fitness value or the error is obtained at convergence. These figures reveal three major areas: (1) the MSE convergence curves which is plotted on fitness value against iteration (top of each figure), (2) the fitness values or the optimized error from the same curve, and (3) the survival probabilities and scalability plotted on survival probabilities over rock hardness values. The figures recorded in (Figures 5 and 6) represent just one run in each case. For example, in Table 2, the details for each parameter for the proposed model are recorded from the 10 runs in order to obtain a fair value since some converge earlier than the others. It must be noted that Figures 6(a) and 6(b) are indeed the MSE curves for the hybrid models HSRF$_{-\cos}$ and HSGF$_{-\cos}$ respectively. It worth noting that 3D dimension with total matrix size of 6 has the MSE convergent curve displayed as in Figure 7.

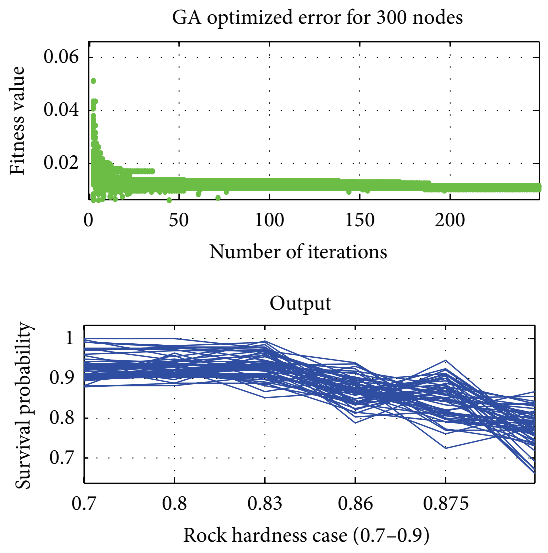
Details of the survival probability and the scalability of the GA algorithms are in Table 3, for 30, 120, 200, and 300 nodes, respectively. The initial survival probability for the GA-SBF is between 87.3% and 94.2% in soft rock layers declining to 68.7%–84.5% at harder rock layers (Figure 6(a)), the GA-CRBF is 88.9%–100% at the soft rock layers and 66.0%–86.6% at the harder rock layers.



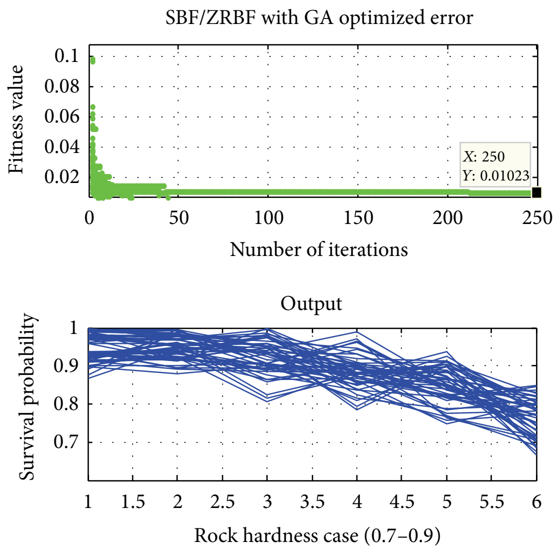
(a) GA/SBF model



(b) GA/CRBF model



(c) GA/GRBF model



(d) SBF-ZRBF model

Figure 5: GA optimized error for basic models—SBF (a), CRBF (b), and GRBF (c).

The average performance trends of each parameter of each model after a number of simulations are displayed in Table 3. The performance of the model parameters using the 5th order polynomial reveals that SBF and CRBF are almost at par with R^2 value for SBF 0.8228, 0.8513, 0.8337, 0.6544, 0.6373, and 0.9293 and CRBF R^2 values of 0.8463, 0.8642, 0.8025, 0.6025, 0.7230, and 0.8457 (Table 2) represent the mean iteration (MI), standard variance (SD VAR), standard deviation (SD), convergent time (conv. time), optimized error (ERROR), and central processing unit or time (CPU), respectively. It must be noted that R^2 values indicate the strength or contribution as well as weakness of the model or model parameters, that is, the higher the value is close to 1 the better the trend. The GA-ZRBF and GA-GRBF followed in decreasing order of performance. Computationally, the performance of the models is SBF with R^2 value of 0.9292 and CRBF R^2 value of 0.9022 and the rest follow in descending order in Table 3.

From Table 3 the nonlinear hybrid has R^2 value of 0.9022, 0.8102, 0.5716, 0.5558, 0.9087, and 0.7975, for HSRF-CRBF, performed better than HSRF-GRBF with R^2 value of 0.7515, 0.8584, 0.246, 0.5743 0.5094, and 0.3708 for mean iteration, standard variance, standard deviation, convergent time, optimized error, and central processing time, respectively.

Again the optimized errors recorded from MSE convergence curves were in order of best performance HSRF-GRBF (0.01076), HSRF-CRBF (0.01109), ZRBF (0.012183), GRBF (0.012291), SBF (0.0126), and CRBF (0.0129). The trends of the

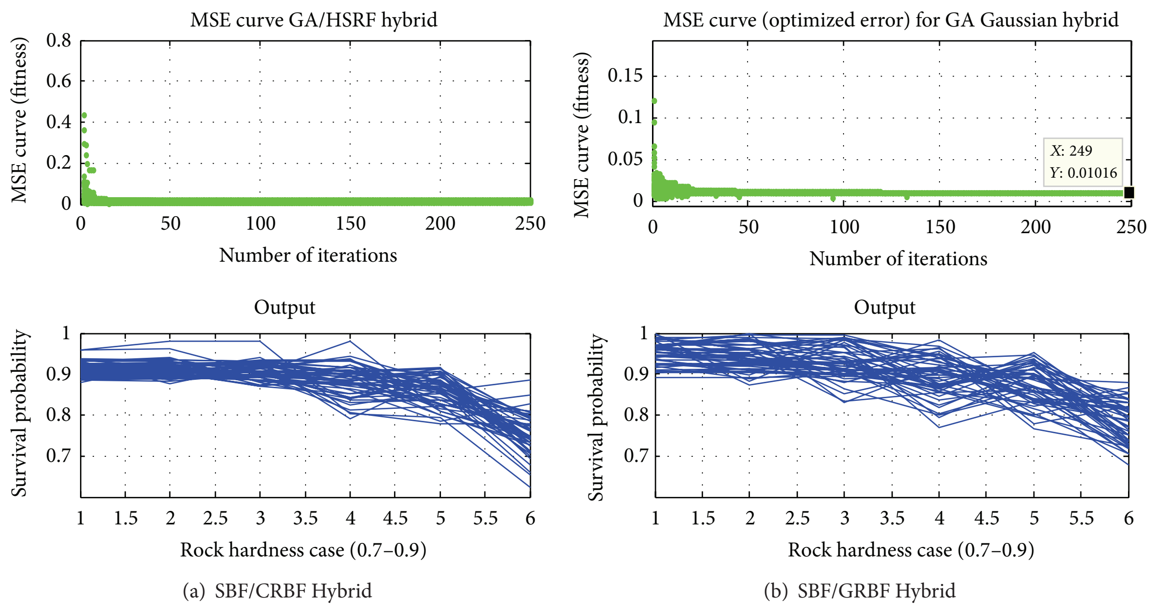


Figure 6: GA Convergence and optimized error for hybrids.

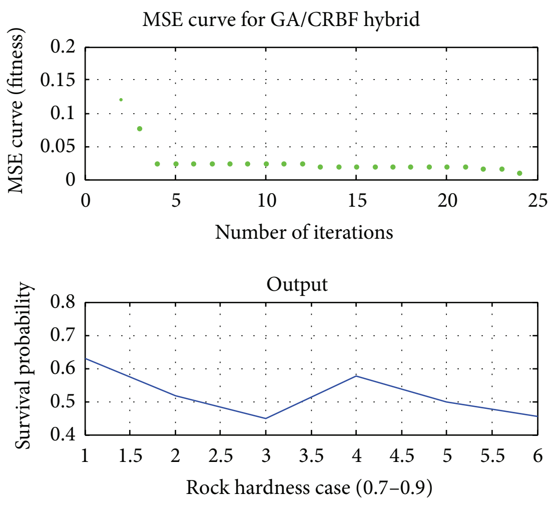


Figure 7: MSE Convergence curve (optimized error) for GA/CRBF Hybrid.

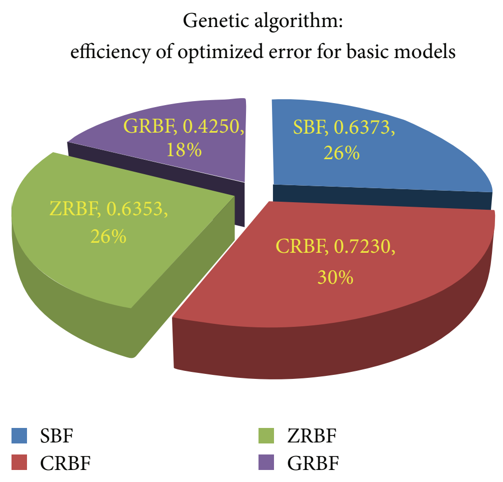


Figure 8: The optimized error for GA trained model.

optimized errors of the basic models are displayed in Figure 8 using the fifth order polynomial. The scalability of the basic model is displayed in Table 4, for 30, 120, 200, and 300 nodes and the results indicate strong scalability with respect to rock types and rock hardness.

The relationship between the various models with respect to the central processing unit (CPU) was profiled for different runs to assess its scalability (Table 5). The proposed GA-SBF and GA-CRBF have better usage of CPU time, and optimizes its parameters with $R^2 = 1$. It was very comparative to the PSO-CRBF in addition the PSO-CRBF was outstanding among the swarm search whiles the SBF is outstanding in the GA.

Comparing the GA trained in this work and PSO trained [54], the GA-SBF has better utilization of resources (e.g., processing time) among the basic models in the genetic algorithm, whilst the PSO-CRBF was outstanding in keeping the error at minimum. The GA-SBF shows outstanding performance in terms of processing time which was slightly higher than that of the CRBF as in Figures 9 and 10. Again the results reveal that both GA-SBF and GA-CRBF with the absolute operation have efficient resource utilization better than that of Gaussian RBF. It has been ascertained that RBF produces more accurate predictions than other models such as MLP, RBFNN-MQ when applied to other applications. This is also true in the basic PSO models displayed in [54]. Similar works in evacuation procedures including that of [55]

Table 3: Efficiency of CPU time and other parameters for the Genetic Algorithm.

	B_5	B_4	B_3	B_2	B_1	B_0	R^2
			Basic Models				
CRBF							
MI	$0.2387x$	$-6.7183x$	$69.811x$	$-327.23x$	$670.01x$	-401.21	**0.8228**
SD VAR	$0.4104x$	$-11.516x$	$119.15x$	$-555.26x$	$1127.8x$	-676.67	**0.8642**
SD	$1E-06x$	$-4E-05x$	$0.0004x$	$-0.0022x$	$0.0048x$	0.0044	**0.8025**
CONV TIME	$-0.0036x$	$0.0871x$	$-0.7031x$	$2.0152x$	$-1.0244x$	6.9667	**0.6025**
ERROR	$9E-06x$	$-0.0002x$	$0.0015x$	$-0.0031x$	$-0.0038x$	0.0243	**0.6705**
CPU TIME	$0.0234x$	$-0.7047x$	$8.0295x$	$-41.794x$	$93.664x$	11.913	**0.8457**
GRBF							
MI	$-0.0291x$	$0.4358x$	$0.3208x$	$-29.708x$	$134.9x$	-89.869	**0.5841**
SD VAR	$-0.0719x$	$2.1191x$	$-22.04x$	$94.541x$	$-141.9x$	125.91	**0.7413**
SD	$-8E-07x5$	$2E-05x$	$-0.0003x$	$0.0012x$	$-0.0025x$	0.009	**0.3639**
CONV TIME	$-0.0012x$	$0.025x$	$-0.1077x$	$-0.6216x$	$4.0274x$	4.7333	**0.6921**
ERROR	$2E-05x$	$-0.0005x$	$0.0047x$	$-0.021x$	$0.0402x$	-0.0136	**0.4250**
CPU TIME	$-0.0048x$	$0.0185x$	$1.7387x$	$-21.473x$	$80.745x$	46.753	**0.5226**
MLP/SBF							
MI	$0.2387x$	$-6.7183x$	$69.811x$	$-327.23x$	$670.01x$	-401.2	**0.8228**
SD VAR	$0.4142x$	$-11.635x$	$120.49x$	$-561.86x$	$1140.9x$	-684.67	**0.8513**
SD	$1E-06x5$	$-3E-05x$	$0.0004x$	$-0.0024x$	$0.0053x$	0.004	**0.8337**
CONV TIME	-0.1348	$3.3687x$	$-29.916x$	$114.35x2$	$-187.89x$	227.07	**0.6544**
ERROR	$3E-05x$	$-0.0008x$	$0.008x$	-0.0365	$0.0722x$	-0.0336	**0.6373**
CPU TIME	$0.262x$	-7.2534	$73.789x$	-336.13	$661.68x$	-299.77	**0.9293**
ZRBF							
MI	$0.0027x$	$-0.118x$	$1.2589x$	$-5.4278x$	$21.815x$	-5.806	**0.6852**
SD VAR	$-0.0604x$	$1.8218x$	$-20.244x$	$98.534x$	$-182.06x$	113.65	**0.5061**
SD	$5E-07x$	$-1E-05x$	$9E-05x$	$-0.0003x$	$0.0005x$	0.0069	**0.9609**
CONV TIME	$0.006x$	$-0.108x$	$0.3516x$	$2.5529x$	$-13.913x$	26.633	**0.5668**
ERROR	$9E-06x$	$-0.0002x$	$0.0015x$	$-0.0037x$	$0.0029x$	0.0087	**0.6353**
CPU TIME	$0.1515x5$	$-3.9365x$	$36.494x$	$-147.02x$	$262.11x$	-54.847	**0.6764**
			HYBRIDS				
HSBF-CRBF							
MI	$0.0084x$	$-0.2218x$	$2.049x$	$-7.8279x$	$12.126x$	106.78	**0.8102**
SD VAR	$0.0054x$	$0.1557x$	$1.6269x$	$-7.278x$	$12.885x$	69.237	**0.5716**
SD	$-6E-07x$	$2E-05x$	$-0.0001x$	$0.0004x$	$-0.0002x$	0.0084	**0.5558**
CONV TIME	$-0.0042x$	$0.3085x$	$-5.3823x$	$35.963x$	$-91.22x$	135.47	**0.9087**
ERROR	$-6E-06x$	$0.0002x$	$-0.0017x$	$0.0078x$	$0.0148x$	0.0201	**0.7975**
CPU TIME	$-0.0862x$	$1.3547x$	$-2.4501x$	$-42.277x$	$174.77x$	96.533	**0.9022**
HSBF-GRBF							
MI	$-0.0008x$	$-0.1558x$	$3.962x$	$-33.969x$	$119.01x$	-25.5	**0.7515**
SD VAR	$0.0092x$	$-0.3751x$	$5.3891x$	$-34.64x$	$98.858x$	-23.064	**0.8584**
SD	$6E-07x$	$-2E-05x$	$0.0002x$	$-0.001x$	$0.0019x$	0.0074	**0.246**
CONV TIME	$0.0367x$	$-1.1456x$	$13.231x$	$-70.314x$	$171.94x$	-39.933	**0.5743**
ERROR	$2E-06x$	$-6E-05x$	$0.0006x$	$-0.0029x$	$0.0063x$	0.0058	**0.5094**
CPU TIME	$-0.2077x$	$4.9802x$	$-41.885x$	$149.4x$	$218.83x$	355.33	**0.3708**

have been done; however underground characteristics and limitation of choice of safe exits makes it impractical.

Furthermore, the results indicate that both the PSO and the GA have strong survival probabilities and are scalable. The future trend of the models for the PSO and GA are good. In addition the model as a whole in both cases had the trend of the R^2 value of 1 which implies that both can stand the test of time. However the PSO trained models are more accurate in terms of the optimized errors, that is, CRBF (0.0108), SBF (0.016), GRBF 0.013, ZRBF (0.012109), and CRBF with nonlinear weight hybrid of negative cosine (0.009) [56]. On the other hand, the optimized errors of the

Table 4: Scalability of model to survival probability range, robot location, and rock type.

Location	Mica	Coal	Granite	Feldspar	Quartz	Mineral
			GA-CRBF			
(10, 6, 5)	0.8828–1	0.9022–1	0.873–1	0.7822–0.9675	0.7323–0.9369	0.6566–0.8449
(10, 5, 4)	0.9194–0.9795	0.9032–0.9505	0.8946–0.9522	0.8763–0.9665	0.7977–0.9695	0.7973–0.8689
(6, 5, 4)	0.9705–0.9936	0.8928–0.9815	0.8977–0.9576	0.8626–0.9710	0.7911–0.9431	0.6589–0.7931
(3, 1, 10)	0.7232–0.8348	0.6655–0.7914	0.6542–0.7874	0.6613–0.7448	0.5841–0.7291	0.4883–0.5702
			GA-SBF			
(10, 6, 5)	0.8911–0.9259	0.8808–0.9235	0.8592–0931	0.82–0.9287	0.8075–0.935	0.7222–0.8286
(10, 5, 4)	0.8943–0.9292	0.8683–0.9342	0.8711–0.9256	0.8454–0.9262	0.7926–0.922	0.7286–0.8443
(6, 5, 4)	0.8875–0.9384	0.8811–0.9773	0.8310–0.8997	0.8227–0.9035	0.6995–0.9467	0.7562–0.8163
(3, 1, 10)	0.7944–0.8455	0.7533–0.7924	0.7372–0.7461	0.5982–0.6947	0.5626–0.7153	0.5546–0.6857
			GA-GRBF			
(10, 6, 5)	0.8819–1	0.8819–0.9789	0.8521–0.994	0.8021–0.936	0.7237–0.9466	0.6609–0.8658
(10, 5, 4)	0.8819–0.9295	0.8733–0.915	0.8761–0.9133	0.8332–0.9068	0.8119–0.9169	0.7208–0.8717
(6, 5, 4)	0.7685–0.9384	0.7711–0.9773	0.7310–0.8997	0.6927–0.8935	0.6995–0.8467	0.5562–0.8453
(3, 1, 10)	0.6844–0.8455	0.6633–0.7924	0.6322–0.7461	0.5582–0.6947	0.5126–0.59153	0.5346–0.5857

Table 5: Average performance of the models CPU time (GA).

Models	B_5	B_4	B_3	B_2	B_1	B_0	R^2
			Average performance of cpu time (after 10 runs)				
SBF	$0.262x$	$-7.2534x$	$73.789x$	$-336.13x$	$661.68x$	−299.77	0.9293
CRBF	$0.0234x$	$-0.7047x$	$8.0295x$	$-41.794x$	$93.664x$	11.913	0.8457
ZRBF	$0.1515x$	$-3.9365x$	$36.494x$	$-147.02x$	$262.11x$	−54.847	0.6764
GRBF	$-0.0048x$	$0.0185x$	$1.7387x$	$-21.473x$	$80.745x$	46.753	0.5226
CRBF HYB	$-0.0862x$	$1.3547x$	$-2.4501x$	$-42.277x$	$174.77x$	96.533	0.9022
GRBF HYB	$-0.2077x$	$4.9802x$	$-41.885x$	$149.4x$	$-218.83x$	355.33	0.3708
			Average performance of the models				
SBF	$12.601x$	−217.38	$1403x$	$-4170.3x$	$5585.3x$	−2558	1
CRBF	$4.1824x$	$-75.171x^4$	$512.36x$	$-1624.6x$	$2310.9x$	−1072.5	1
ZRBF	$4.5959x$	$-79.917x$	$526.65x$	$-1615.4x$	$2231.7x$	−1019.9	1
GRBF	$5.3574x$	$-95.466x$	$646.21x$	$-2038.5x$	$2890.4x$	−1341.4	1
CRBF HYB	$10.527x$	$-178.75x$	$1135x$	$-3306.1x$	$4288.6x$	−1833.3	1
GRBF HYB	$13.263x$	$-224.22x$	$1415.3x$	$-4096.5x$	$5297.6x$	−2298.8	1

Table 6: Comparing the two population sizes.

Run/PopnSize	CPU TIME 30	CPU TIME 20	Error 30	Error 20	SD 30	SD 20
1	408.77	293.27	0.01122	0.009856	0.008483	0.008628
2	349.41	208.515	0.01178	0.01001	0.008237	0.00886
3	216.81	226.515	0.009872	0.01103	0.008768	0.008956
4	464.5	254.49	0.007898	0.01179	0.008337	0.008761
5	192.82	181.17	0.0014	0.02014	0.008318	0.008786
6	319.36	330.87	0.01756	0.01017	0.008072	0.00851
7	304.38	264.15	0.01006	0.01962	0.008702	0.009362
8	322.18	259.2	0.1639	0.0107	0.00851	0.009586
9	469.86	113.14	0.01318	0.01001	0.008743	0.008618
10	331.21	106.75	0.01075	0.0100013	0.008719	0.007096
Total	3379.3	2238.07	0.25762	0.1233273	0.084889	0.087163
Average	**337.93**	**223.807**	**0.025762**	**0.01233273**	**0.0084889**	**0.0087163**
5th order R^2 value	**0.3667**	**0.7475**	**0.3968**	**0.4194**	**0.4212**	**0.9206**

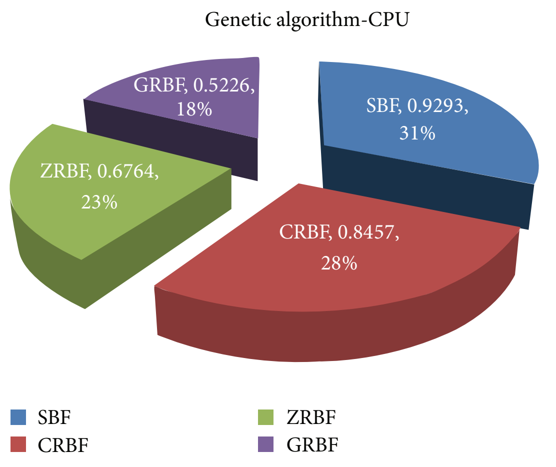


Figure 9: Computational efficiency of GA trained model.

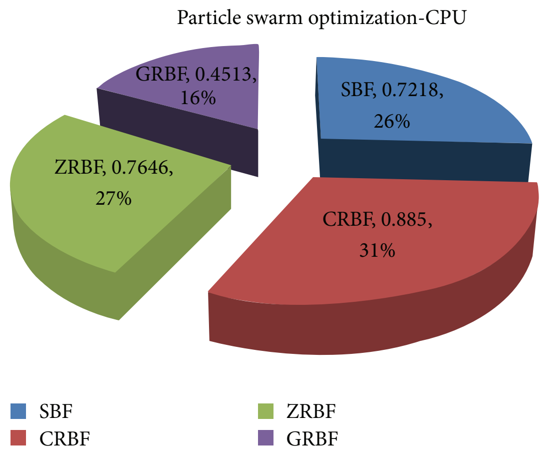


Figure 10: Computational efficiency of PSO trained model.

GA trained models are CRBF (0.012923), SBF (0.0126), GRBF (0.01229), ZRBF (0.012183), and nonlinear hybrid of negative cosine weight CRBF (0.012291). In terms of training, the GA trains better than the PSO but the PSO's signal transmission is better in terms of harder rock areas. Generally the errors were higher in GA than in PSO. For more information on the particle swarm models, and the hybrids, readers may refer to the following references [54, 56]. It must be noted that the GA-SBF performed well with genetic algorithm. This was demonstrated in the computational efficiency in Figures 8 and 9. The GA's values were 31%, 28%, 23%, and 18% for SBF, CRBF, ZRBF, and GRBF, respectively, while the PSO values were 31%, 26%, 27%, and 16% for CRBF, SBF, ZRBF, and GRBF, respectively. The PSO favoured the CRBF and the GA favoured the SBF.

To verify the optimal parameters the study used both population sizes 20 and 30 to simulate the results. The differences of the CPU time, SD, and the error for the population sizes of 20 and 30 as seen in Table 6. The population size of 20 is outstanding in all the three outcomes: CPU time, SD, and error as 223.807, 0.0087163, and 0.01233273, respectively. The simulated results (CPU time, SD, and the error) for the two population sizes are displayed in Table 6. These results reveal that there is much demand on CPU time with error more than twice compared with target set for population size of 30.

7. Conclusion

This study has been able to construct a hybrid neural network for underground rescue system using absolute operation with non-Gaussian and compact radial basis transfer functions. This novel hybrid transfer function model shows outstanding performance over the non-Gaussian models. The study has discussed the new model using GA training and has compared it with the PSO training in previous work. The proposed model is very competitive and efficient to existing models like the Gaussian. The results of the GA show an alternative training method for rescue system for emergency operation for mining grounds and tunnels, where rock hardness is relatively constant. It can also be used for hospitals, building surveillance, and evacuation situations. The processing efficiency in the models is better as compared with Gaussian model.

The challenge is that the optimized errors obtained from the models with genetic algorithm are high (note that this is only in comparison with the PSO) and future work should address this limitation.

For future work there is the need to use the trained NN to predict the survivability with instant inputs of complete/correct and/or incomplete/incorrect location and rock type from the randomly picked mobile sensor, to further verify the accuracy and robustness of the prediction. In addition, more hybrids will need to be investigated with the GA of both linear and nonlinear weights to further optimize the error as well as investigate the signal sensitivity of the model.

It is envisioned that this model is a good alternative in the mining industry and can go a long way to help save lives. It is robust, fault tolerant, and competitive to the Gaussian and non-Gaussian models.

Conflict of Interests

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

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (no. 71271103) and by the Six Talents Peak Foundation of Jiangsu Province. 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, Thomas Kwaw Annan, Isaac Owusu Kwankye, Clement Owusu-Asamoah, Willet Agongo, Nathaniel Quansah, Francis Owusu Mensah, Joseph Adu-Mensah, F. T. Oduro, Henry Kwabena Asante Antwi, E. Omari-Siaw, and Qing Zou.

References

[1] M. M. Bhaskar and S. Maheswarapu, "A hybrid genetic algorithm approach for optimal power flow," *Telkomnika*, vol. 9, no. 2, pp. 211–216, 2011.

[2] M.-W. Li, W.-C. Hong, and H.-G. Kang, "Urban traffic flow forecasting using Gauss-SVR with cat mapping, cloud model and PSO hybrid algorithm," *Neurocomputing*, vol. 99, pp. 230–240, 2013.

[3] N. T. Liu, J. B. Holcomb, C. E. Wade et al., "Development and validation of a machine learning algorithm and hybrid system to predict the need for life-saving interventions in trauma patients," *Medical & Biological Engineering & Computing*, vol. 52, no. 2, pp. 193–203, 2014.

[4] C.-N. Ko and C.-M. Lee, "Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter," *Energy*, vol. 49, no. 1, pp. 413–422, 2013.

[5] J. Zheng, X. Shao, L. Gao, P. Jiang, and Z. Li, "A hybrid variable-fidelity global approximation modelling method combining tuned radial basis function base and kriging correction," *Journal of Engineering Design*, vol. 24, no. 8, pp. 604–622, 2013.

[6] S. Azim and S. Aggarwal, "Hybrid model for data imputation: using fuzzy c means and multi layer perceptron," in *Proceedings of the 4th IEEE International Advance Computing Conference (IACC '14)*, pp. 1281–1285, Gurgaon, India, February 2014.

[7] K. Hasani, S. A. Kravchenko, and F. Werner, *A Hybrid Harmony Search/Simulated Annealing Algorithm for Minimizing Mean Flow Time on Two Identical Parallel Machines with a Single Server*, 2013.

[8] E. Merlo, I. McAdam, and R. de Mori, "Feed-forward and recurrent neural networks for source code informal information analysis," *Journal of Software Maintenance and Evolution: Research and Practice*, vol. 15, no. 4, pp. 205–244, 2003.

[9] J.-X. Peng, K. Li, and D.-S. Huang, "A hybrid forward algorithm for RBF neural network construction," *IEEE Transactions on Neural Networks*, vol. 17, no. 6, pp. 1439–1451, 2006.

[10] 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.

[11] F. Fernández-Navarro, C. Hervás-Martínez, R. Ruiz, and J. C. Riquelme, "Evolutionary generalized radial basis function neural networks for improving prediction accuracy in gene classification using feature selection," *Applied Soft Computing Journal*, vol. 12, no. 6, pp. 1787–1800, 2012.

[12] F. Fernández-Navarro, C. Hervás-Martínez, P. A. Gutiérrez, and M. Carbonero-Ruz, "Evolutionary q-Gaussian radial basis function neural networks for multiclassification," *Neural Networks*, vol. 24, no. 7, pp. 779–784, 2011.

[13] K. W. Wong, T. D. Gedeon, and D. Tikk, "An improved multidimensional α-cut based fuzzy interpolation technique," in *Proceedings of the International Conference Artificial Intelligence in Science and Technology (AISAT '00)*, pp. 29–32, Hobart, Australia, 2000.

[14] X. Wu and B. M. Wilamowski, "Advantage analysis of sigmoid based RBF networks," in *Proceedings of the 17th IEEE International Conference on Intelligent Engineering Systems (INES '13)*, pp. 243–248, June 2013.

[15] H. Yu, T. Xie, S. Paszczyñski, and B. M. Wilamowski, "Advantages of radial basis function networks for dynamic system design," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 12, pp. 5438–5450, 2011.

[16] J.-F. Qiao and H.-G. Han, "A repair algorithm for radial basis function neural network and its application to chemical oxygen demand modeling," *International Journal of Neural Systems*, vol. 20, no. 1, pp. 63–74, 2010.

[17] S.-F. Cheng, C.-I. Hung, and C.-K. Chen, "A clustering analysis method for exploring microorganism conversion of biomass energy," *Environmental Progress & Sustainable Energy*, vol. 33, no. 2, pp. 636–648, 2014.

[18] H. Peng, T. Ozaki, V. Haggan-Ozaki, and Y. Toyoda, "A parameter optimization method for radial basis function type models," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 432–438, 2003.

[19] X. Liu, K. Li, M. McAfee, and G. W. Irwin, "Improved nonlinear PCA for process monitoring using support vector data description," *Journal of Process Control*, vol. 21, no. 9, pp. 1306–1317, 2011.

[20] C. W. Chen, P. C. Chen, and W. L. Chiang, "Stabilization of adaptive neural network controllers for nonlinear structural systems using a singular perturbation approach," *Journal of Vibration and Control*, vol. 17, no. 8, pp. 1241–1252, 2011.

[21] A. D. Niros and G. E. Tsekouras, "A novel training algorithm for RBF neural network using a hybrid fuzzy clustering approach," *Fuzzy Sets and Systems*, vol. 193, pp. 62–84, 2012.

[22] 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.

[23] M. G. Hinchey, J. L. Rash, C. A. Rouff, and D. Gračanin, "Achieving dependability in sensor networks through automated requirements-based programming," *Computer Communications*, vol. 29, no. 2, pp. 246–256, 2006.

[24] A. Le, J. Loo, A. Lasebae, M. Aiash, and Y. Luo, "6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1189–1212, 2012.

[25] M. Y. Mashor, "Hybrid training algorithm for RBF network," *International Journal of the Computer, the Internet and Management*, vol. 8, pp. 50–65, 2000.

[26] M. J. Brnich Jr. and K. M. Kowalski-Trakofker, *Underground Coal Mine Disasters 1900–2010: Events, Responses, and a Look to the Future*, NIOSH Office of Mine Safety and Health, Pittsburgh, Pa, USA, 2014, http://www.cdc.gov/niosh/mining/UserFiles/works/pdfs/ucmdn.pdf.

[27] 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.

[28] A. Patri, A. Nayak, and S. Jayanthu, "Wireless communication systems for underground mines—a critical appraisal," *International Journal of Engineering Trends and Technology*, vol. 4, no. 7, 2013.

[29] L. E. Linderman, J. A. Rice, S. Barot, B. F. Spencer, and J. T. Bernhard, "Characterization of wireless smart sensor performance," *Journal of Engineering Mechanics*, vol. 136, no. 12, pp. 1435–1443, 2010.

[30] Q. Ling, Y. Fu, and Z. Tian, "Localized sensor management for multi-target tracking in wireless sensor networks," *Information Fusion*, vol. 12, no. 3, pp. 194–201, 2011.

[31] 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.

[32] V. Bulbenkiene, A. Pecko, E. Zulkas, A. Kuprinavicius, A. Sokolov, and G. Mumgaudis, "Energy sub-metering data acquisition system," *Elektronika ir Elektrotechnika*, pp. 99–102, 2011.

[33] R. Haner and P. Keifer, *Encyclopedia of Magnetic Resonance*, John Wiley & Sons, 2009.

[34] A. Patri, A. Nayak, and S. Jayanthu, "Nav view search," 2013.

[35] 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, no. 2, 2013.

[36] X. Feng, Z. Xiao, and X. Cui, "Improved RSSI for wireless Sensor Network in 3D," 2011.

[37] D. S. Broomhead and D. Lowe, *Radial Basis Functions, Multi-Variable Functional Interpolation and Adaptive Networks*, DTIC Document, Royals Signals & Radar Establishment, Worcestershire, UK, 1988.

[38] D. Baldwin, Ü. Göktaş, and W. Hereman, "Symbolic computation of hyperbolic tangent solutions for nonlinear differential-difference equations," *Computer Physics Communications*, vol. 162, no. 3, pp. 203–217, 2004.

[39] Y.-K. Yang, T.-Y. Sun, C.-L. Huo, Y.-H. Yu, C.-C. Liu, and C.-H. Tsai, "A novel self-constructing Radial Basis Function Neural-Fuzzy System," *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2390–2404, 2013.

[40] S. Marsili and P. Alba, "Adaptive mutation in genetic algorithm," in *Soft Computing*, vol. 4, pp. 76–80, Springer, 2000.

[41] X. Tang, L. Zhuang, J. Cai, and C. Li, "Multi-fault classification based on support vector machine trained by chaos particle swarm optimization," *Knowledge-Based Systems*, vol. 23, no. 5, pp. 486–490, 2010.

[42] R. Hassan and W. Crossley, "Spacecraft reliability-based design optimization under uncertainty including discrete variables," *Journal of Spacecraft and Rockets*, vol. 45, no. 2, pp. 394–405, 2008.

[43] E. A. Williams and W. A. Crossley, "Empirically-derived population size and mutation rate guidelines for a genetic algorithm with uniform crossover," in *Soft Computing in Engineering Design and Manufacturing*, pp. 163–172, Springer, 1998.

[44] R. Hassan, "Genetic algorithm approaches for conceptual design of spacecraft systems including multi-objective optimization and design under uncertainty," in *Optimization and Design under Uncertainty, Doctoral Thesis*, Purdue University, 2004.

[45] P. Jain, P. Sharma, V. Sethi, and M. Pandey, "Optimization of the wind power generation unit using genetic algorithm," *International Journal of Engineering Science & Technology*, vol. 4, no. 11, pp. 4592–4597, 2012.

[46] A. H. Wright, "The exact schema theorem," Tech. Rep., Computer Science Department, University of Montana, Missoula, Mont, USA, 2011.

[47] M. Kaya, "The effects of two new crossover operators on genetic algorithm performance," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 881–890, 2011.

[48] J. T. Alander, "On optimal population size of genetic algorithms," in *Proceedings of the IEEE Computer Systems and Software Engineering (CompEuro '92)*, pp. 65–70, 1992.

[49] K.-L. Du and M. Swamy, *Recurrent Neural Networks*, Springer, New York, NY, USA, 2014.

[50] L. Chen, "Real coded genetic algorithm optimization of long term reservoir operation," *Journal of the American Water Resources Association*, vol. 39, no. 5, pp. 1157–1165, 2003.

[51] G. Cormier, R. Boudreau, and S. Thériault, "Real-coded genetic algorithm for Bragg grating parameter synthesis," *Journal of the Optical Society of America B: Optical Physics*, vol. 18, no. 12, pp. 1771–1776, 2001.

[52] L. Davis, *Handbook of Genetic Algorithms*, vol. 115, Van Nostrand Reinhold, New York, NY, USA, 1991.

[53] J. Wu, "On the performance of principal component Liu-type estimator under the mean square error criterion," *Journal of Applied Mathematics*, vol. 2013, Article ID 858794, 7 pages, 2013.

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

[55] Q. Tan, G. H. Huang, C. Wu, and Y. Cai, "IF-EM: an interval-parameter fuzzy linear programming model for environment-oriented evacuation planning under uncertainty," *Journal of Advanced Transportation*, vol. 45, no. 4, pp. 286–303, 2011.

[56] H. Yao, M. O. Ansong, and J. S. Huang, "Weighed nonlinear hybrid neural networks in underground rescue mission," *ISRN Artificial Intelligence*, vol. 2014, Article ID 864020, 13 pages, 2014.