

Optimized Neural Network Using Differential Evolutionary and Swarm Intelligence Optimization Algorithms for RF Power Prediction in Cognitive Radio Network: A Comparative study

Sunday Iliya, Eric Goodyer, Jethro Shell, and Mario Gongora

Centre for Computational Intelligence,
School of Computer Science and Informatics,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom

Email:sundayiliyagoteng@yahoo.com, eg@dmu.ac.uk,
jethros@dmu.ac.uk, mgongora@dmu.ac.uk

John Gow

School of Engineering,
Media and Sustainable Development,
De Montfort University, The Gateway,
Leicester LE1 9BH, England,
United Kingdom

Email:jgow@dmu.ac.uk

Abstract—Cognitive radio (CR) technology has emerged as a promising solution to many wireless communication problems including spectrum scarcity and underutilization. The a priori knowledge of Radio Frequency (RF) power (primary signals and/or interfering signals plus noise) in the channels to be exploited by CR is of paramount importance. This will enable the selection of channel with less noise among idle (free) channels. Computational Intelligence (CI) techniques can be applied to these scenarios to predict the required RF power in the available channels to achieve optimum Quality of Service (QoS). In this paper, we developed a time domain based optimized Artificial Neural Network (ANN) model for the prediction of real world RF power within the GSM 900, Very High Frequency (VHF) and Ultra High Frequency (UHF) TV bands. The application of the models produced was found to increase the robustness of CR applications, specifically where the CR had no prior knowledge of the RF power related parameters such as signal to noise ratio, bandwidth and bit error rate. The models used, implemented a novel and innovative initial weight optimization of the ANN's through the use of differential evolutionary and swarm intelligence algorithms. This was found to enhance the accuracy and generalization of the ANN model. For this problem, DE/best/1/bin was found to yield a better performance as compared with the other algorithms implemented.

I. INTRODUCTION

Most of the licensed radio spectrums are grossly underutilized while the unlicensed spectrums are overcrowded. The present spectrum scarcity is the direct consequence of spectrum allocation policy and not the fundamental lack of spectrum. The first bands to be approved for CR communication by the US Federal Communication Commission (FCC) because of their gross underutilization in time, frequency and spatial domain are the very high frequency and ultra-high frequency (VHF/UHF) TV bands [1][2]. This paper is focused on the study of real world RF power distribution in specific selected channels (54MHz to 110MHz, 470MHz to 670MHz, 890MHz to 908.3MHz GSM up-link, 935MHz to 953.3MHz GSM down-link) within the VHF/UHF bands

and the GSM 900 band. In order to circumvent the problem of spectrum scarcity and underutilization, a new paradigm of wireless communication needs to be adopted. Advanced Cognitive Radio (CR) or Adaptive Spectrum Sharing (ASS) is one of the ways to optimize our wireless communications technologies for high data rates whilst maintaining user desired quality of service (QoS) requirements. CR is a radio equipped with the capability of awareness, perception, adaptation and learning of its radio frequency (RF) environment [3]; through which many of the digital signal processing that were traditionally done in hardware are implemented via software. Irrespective of the definition of CR, it has the followings basic characteristics: observation, adaptability and intelligence. CR is the key enabling technology for dynamic spectrum access and a promising solution for the present problem of spectrum scarcity and underutilization. Cognitive radio network consists of two users i.e. the primary users (PU) who are the incumbent licensed owners of the spectrum and the cognitive radio commonly called the secondary users (SU) who intelligently and opportunistically access the unused licensed spectrum on a no interference base i.e. when the licensed owners are not using the channel (the overlay) or SU can coexist with the PU as long as the interference to the PU is below a given threshold which will not be harmful to the PU nor degrade the QoS requirements of the PU (the underlay CR network scheme) [4][1]. There are four things involved in cognitive radio network, these are: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility [5][6]. In spectrum sensing, the CR senses the PU spectrum using either energy detector, cyclostationary features detector, match filter detector, eigen values detector, etc to sense the occupancy status of the PU. Based on the sensing results, the SU (CR) will take a decision using a binary classifier to classify the PU channels (spectrums) as either busy or idle there by identifying the white spaces. Spectrum sharing deals with how to efficiently share the available white spaces (spectrum holes) among other CR (SU) within a given geographical location at a given period of time while spectrum mobility is the ability of the CR to

vacate the channel when the PU reclaimed ownership of the channel and search for another spectrum hole to communicate. During the withdrawal period, the CR should maintain seamless communication. Many wireless broadband devices ranging from simple communication to complex systems automation, are deployed daily with increasing demand for more, this calls for optimum utilization of the limited spectrum resources via CR paradigm. Future wireless communication device should be equipped with cognitive capability for efficient spectrum utilization. In this paper, we combine the power of differential evolutionary (DE) and swarm intelligence algorithms, and ANN to develop an optimized ANN model for the prediction of real world radio frequency (RF) power. The proposed algorithm used a priori data to enable the system to avoid noisy channels. The previous knowledge of the RF power allowed the cognitive radio to predictively select channels with the least noise among those that were unused or free. This would allow for a reduced utilization of radio resources including transmitted power, bandwidth, in turn maximizing the usage of the limited spectrum resources. The data used in this study was obtained by capturing real world RF data for two months using Universal Software Radio Peripheral 1 (USRP 1). The digital signal processing and capturing of the data were done using gnuradio which is a combination of Python and C++ while the ANN models design and prediction were done in Matlab. The experiment was conducted at Centre for Computational Intelligence, De Montfort University, UK, located very close to Leicester city centre.

The rest of this paper is consist of the following sections. Section II previously presented research in this field. This will be followed by Section III and section IV, that gives brief description of neural network and the optimization algorithms implemented. Experimental details are discussed in section V. The paper is concluded with Section VI, which discusses the results of the experiments, section VII gives the summary of the findings.

II. RELATED WORK

There are different types of Computational Intelligence (CI) and machine learning that can be used in CR such as genetic algorithms, fuzzy logic system (FLS), neural network, hidden Markov model, game theory, linear regression and linear predictors, Bayesian inference based predictors, swarm intelligence, etc. Some for learning and prediction, some for optimization of certain transmission parameters while others for decision making [7][8][9]. Spectrum hole prediction using Elman recurrent artificial neural network (ERANN) was proposed in [10]. It uses the cyclostationary features of modulated signals to determine the presence or absence of primary signals while the input of the ERANN consists of time instances. The inputs and the target output used in the training of the ERANN and prediction were modeled using ideal multivariate time series equations and assumptions which are often different in real life where PU signals may be embedded in noise and/ or interfering signals. The fact that spectrum holes were detected does not depict any information about the expected noise and/ or level of interference based on the channels history which is vital in selecting the channels to be used among the unused channels. Traffic pattern prediction using seasonal autoregressive integrated moving-average (SARIMA) was proposed for reduction of SU hopping rate and interference effects

on PU while maintaining a fare blocking rate [11]. The model (SARIMA) does not depict any information about the expected noise power.

Transmission rate prediction using adaptive neural fuzzy inference system (ANFIS) was proposed in [12]. This model is used to predict the data rate (6, 12, 24, 36, 48 and 54 Mbps) that can be achieved in WLAN using a 802.11a/g configuration as a function of time. The training data was obtained by generating a random data rate with an assigned probability of occurrence at a given time instance, thereby forming a time series. In this paper, no real world data was used. Additionally, the research did not take into account the noise or interference level which can affect the predicted rates. Semi Markov model (SMM) and continuous-time Markov chain (CTMC) models have also been used for the prediction of packet transmission rates [13]. This avoids packet collisions by periodic spectrum sensing and prediction of temporal wireless local area network (WLAN) activities combined with hoping to a temporary inactive channel. However, no reference is made to the expected noise level among the inactive channels that are selected. Fuzzy logic (FL) is a CI method that can capture and represent uncertainty. As a result in has been used in CR research for decision making processes. An FL based decision system was modeled for spectrum hand-off decision-making in a context characterized by uncertain and heterogeneous information [14] and fuzzy logic transmit power control for cognitive radio. The proposed system was used for the minimization of interference to PU's while ensuring the transmission rate and quality of service requirements of secondary users [15]. The researcher did not, however, include any learning from past experience or historical data. An exponential moving average (EMA) spectrum sensing using energy prediction was implemented in [16]. The EMA achieved a prediction average error of 0.2436 with the assumption that the channel utilization follow exponential distribution with rate parameter =0.2 and signal to noise (SNR) of 10dB; RF real world data was not used in their study. Within this paper we demonstrate the use of an ANN for prediction of real world RF power of selected channels within the GSM band, VHF and UHF bands. An optimized ANN model was produced by combining the global search ability of deferential evolution (DE) algorithms and the local search advantages of back-propagation algorithms. The initial weights of the ANN were evolved using DE after which the ANN was trained more accurately using back-propagation algorithms. This methodology demonstrates the application of previously acquired real world data to enhance the prediction of RF power to assist the implementation of CR applications.

III. ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANN) are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems [17]. Due to the nonlinearity often associated with RF traffic pattern, coupled with random interfering signals or noise resulting from both artificial and natural sources, a fully connected multilayer perceptron (MLP) ANN with two hidden layers was used in this study. The input layer was cast into a high dimensional first hidden layer for proper features selection. The activation functions used in the two hidden layers is nonlinear hyperbolic tangent function, and a linear symmetric straight line is used for the output activation function. Other activation functions were also used, but this

selection gave a better promising results. These functions introduced a nonlinear transformation into the network. The hidden layers serve as a feature detector i.e. during the training; they learn the salient attributes (features) that characterizes the training data. The ANN is trained using differential evolutionary and swarm intelligence algorithms after which the weights are fine tuned using backpropagation algorithm. The synaptic weights and biases were updated every epoch to minimize the Mean Square Error (MSE). A supervised batch training method was used with 60% of the data used for training the ANN, 20% for validation and 20% for testing the trained ANN. Since the back propagation algorithm is used as a local searcher, the learning rate was kept low at 0.01. The inputs of the ANN consist of time domain data of varying rates of change i.e. second, minute, hour, week day (1 to 7), date day (1 to at most 31), week in a month (1 to 5), and month whilst the output gives the power in Decibels (dB). Each input enables the ANN to keep track with the trend of RF power variation as a function of that particular input. In this paper we designed three ANN models. The first model is shown in Fig 1; it consists of only one output neuron and is dedicated for power prediction of only one channel which implies that each channel will have its own dedicated ANN. The second model is depicted in Fig 2, it has only one output neuron but with additional inputs representing the channels. The output neurons of the third model is equal to the number of channels to be considered Fig 3. The data used in this study were obtained by capturing real world RF signals within the GSM 900, VHF and UHF TV bands and FM bands for a period of two months. In all the models, no RF power related parameters such as signal to noise ratio (SNR), bandwidth, and modulation type, are used as the input of the ANN. Thus making the models robust for cognitive radio application where the CR has no prior knowledge of these RF power related parameters. Four ANN topologies were considered: feed forward (FF), feed forward with output feedback (FFB), cascaded feed forward (CFF) and layered recurrent (LC) ANN.

The accuracy and level of generalization of ANN depend largely on the initial weights and biases, learning rate, momentum constant, training data and also the network topology. In this paper, the learning rate and the momentum were kept constant at 0.01 and 0.008 respectively while the initial weights and biases were evolved using differential evolutionary and swarm intelligence algorithms. The initial weights and biases were randomly generated and constrained within the decision space of -2 to 2 using a toroidal bound. A population of 30 individuals or particles was maintained over 1000 generations. At the end of the generation, the ANN weights and biases were initialized using the candidate with the best fitness (candidate with the least MSE, obtain using test data) and then train further using backpropagation algorithm (BPA) to fine tune the weights as detailed in selection process IV-E. Thus producing the final optimized ANN model.

IV. OPTIMIZATION ALGORITHMS

A brief description of the optimization algorithms implemented are presented in this section. We combine the global search capability of population based differential evolutionary and swarm intelligence algorithms with the single solution local search advantages of BPA to evolve the weights and

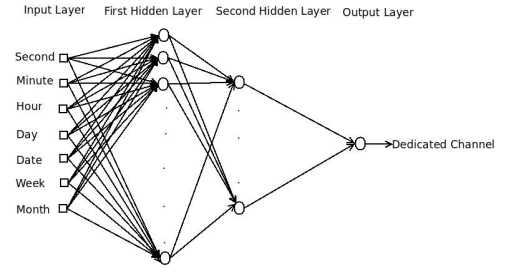


Fig. 1. Dedicated ANN model for one channel

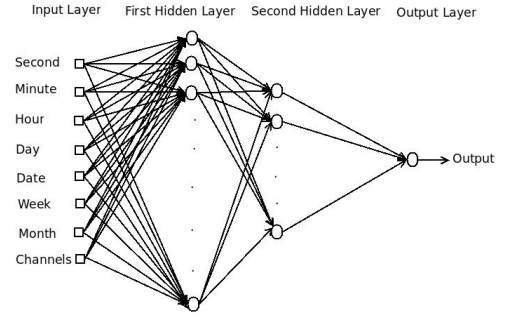


Fig. 2. Multiple channels, single output ANN model

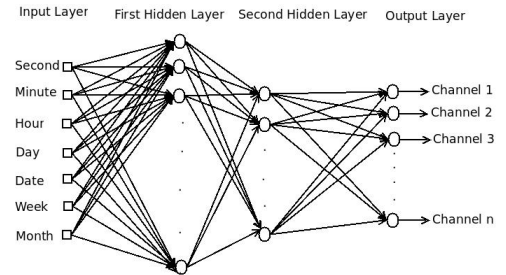


Fig. 3. Multiple channels, parallel outputs ANN model

biases of the optimized ANN model as described in selection process IV-E.

A. Differential Evolution

DE are population based direct search algorithms used to solve continuous optimization problems [18][19]. DE aims at evolving NP population of D dimensional vectors which encodes the G generation candidate solutions $X_{i,G} = \{X_{i,G}^1, \dots, X_{i,G}^D\}$ towards the global optimum, where $i = 1, \dots, NP$. The initial candidate solutions at $G = 0$ are evolves in such a way as to cover the decision space as much as possible by uniformly randomizing the candidates within the search domain using (1).

$$X_{i,G} = X_{min} + rand(1, 0) \cdot (X_{max} - X_{min}) \quad (1)$$

Where $i = 1, \dots, NP$. $X_{min} = \{X_{min}^1 \dots X_{min}^D\}$, $X_{max} = \{X_{max}^1 \dots X_{max}^D\}$ and $rand(1, 0)$ is a uniformly distributed random number between 0 and 1.

B. Mutation

For every individuals (target vectors) $X_{i,G}$ at generation G , a mutant vector $V_{i,G}$ called the provisional or trial offspring is generated via certain mutation schemes. The mutation strategies implemented in this study are:

- DE/rand/1:

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \quad (2)$$

- DE/best/1:

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) \quad (3)$$

- DE/rand-to-best/1:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G}) \quad (4)$$

- DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \quad (5)$$

Where the indexes r_1, r_2, r_3 and r_4 are mutually exclusive positive integers and distinct from i . These indexes are generated at random within the range $[1 PN]$. $X_{best,G}$ is the individual with the best fitness at generation G while F is the mutation constant.

C. Cross over

After the mutants are generated, the offspring $U_{i,G}$ are produced by performing a crossover operation between the target vector $X_{i,G}$ and its corresponding provisional offspring $V_{i,G}$. The two crossover schemes i.e. exponential and binomial crossover are used in this study for all the DE algorithms implemented [20]. The binomial crossover copied the j th gene of the mutant vector $V_{i,G}$ to the corresponding gene (element) in the offspring $U_{i,G}$ if $rand(0, 1) \leq CR$ or $j = j_{rand}$. Otherwise it is copied from the target vector $X_{i,G}$ (parent). The crossover rate CR is the probability of selecting the offspring genes from the mutant while j_{rand} is a random number in the range $[1 D]$, this ensure that at least one of the offspring gene is copied from the mutant. The binomial crossover is represented as:

$$U_{i,G}^j = \begin{cases} V_{i,G}^j & \text{if } (rand(0, 1) \leq CR \text{ or } j = j_{rand}) \\ X_{i,G}^j & \text{otherwise} \end{cases} \quad (6)$$

For exponential crossover, the genes of the offspring are inherited from the mutant vector $V_{i,G}$ starting from a randomly selected index j in the range $[1 D]$ until the first time $rand(0, 1) > CR$ after which all the other genes are inherited from the parent $X_{i,G}$. The exponential crossover is as shown in Section IV-D

D. Exponential crossover

$$U_{i,G} = X_{i,G}$$

generate $j = randi(1, D)$

$$U_{i,G}^j = V_{i,G}^j$$

$$k = 1$$

while $rand(0, 1) \leq Cr$ AND $k < D$ **do**

$$U_{i,G}^j = V_{i,G}^j$$

$$j = j + 1$$

if $j == n$ **then**

$$j = 1$$

end if

$$k = k + 1$$

end while

E. Selection process

The objective function in this study is the MSE of the optimized ANN computed using the test data. After every generation, each offspring $U_{i,G}$ and the corresponding parent $X_{i,G}$ are used to set the weights and biases of the ANN and the MSE of the ANN models are obtain using the test data. The use of the test data (data not known by the ANN nor used to train it) for computation of the fitness function (MSE) does not only result in a more accurate network but also a more robust and generalized ANN model. A greedy selection schemes is used in which if the MSE of the offspring is less than or equal to that of it parent, the offspring will replace the corresponding parent in the next generation otherwise the parent will be maintained among the next generation individuals. At the end of the generation, the most fittest individual (global best) i.e. the candidate with the least MSE among the final evolved solutions is used to initialize the weights and biases of the ANN which is further trained using back propagation algorithms (BPA) to fine tune the weights to produce the final optimized ANN model. The DE is run with 30 individuals for 1000 generations. The fine tuning of the ANN weights using BPA was constrained within a maximum of 200 epoch and 6 validation fails, i.e. the training stop if any of these constrain thresholds is satisfy. The combination of the global search capabilities of DE and the local search advantages of BPA to evolve the weight and biases of ANN have proving to be superior to using only the famous BPA for this problem. The DE algorithm pseudocode is depicted in Section IV-F

F. Differential Evolution Pseudocode

Generate an initial population $X_{G=0}$ of Np individuals.

Initialize ANN weights and biases with each solution in the population Np and Evaluate fitness (MSE) of each using the test data.

while termination condition is not met (Generation) **do**

for $i = 1$ to Np **do**

Initialize ANN weights and biases with $X_{i,G}$ (parent) and evaluate its fitness $FitParent_i$ using the test data.

Generate trial offspring $V_{i,G}$ by mutation using (2).

Generate offspring $U_{i,G}$ by either binomial crossover for DE/rand/1/bin or exponential crossover for DE/rand/1/exp.

Initialize ANN weights and biases with $U_{i,G}$ (offspring) and evaluate its fitness $FitOffspring_i$ using the test data.

end for

for $i = 1$ to Np **do**

Selection Process: Select the best between parent and offspring.

if $FitOffspring_i \leq FitParent_i$ **then**

$$X_{i,G+1} = U_{i,G}$$

else

$$X_{i,G+1} = X_{i,G}$$

end if

end for

end while

Select the global best X_{gb} (i.e. individual with the best fitness at the end of the budget (generation)), initialize the ANN weights and biases with X_{gb} and train it (fine tune the weights) using BPA constrain within a maximum of just 200 epoch and 6 validation fails. This yield the final optimized ANN model.

G. Comprehensive learning particle swarm optimization (CLPSO)

Since this is a comparative study, we also consider two algorithms which are not based on DE schemes rather on swarm intelligence framework. These two particle swarm optimization (PSO) variants are: the standard PSO with inertia weight [21] and the CLPSO [22]. PSO emulates the swarm behavior of which each member of the swarm adapts its search path by learning from its own experience and other members' experiences. The velocity update of PSO and CLPSO are giving by (7) and (8) respectively. In the inertia weighted PSO, each of the particles learn from its local best $pbest$ and the global best $gbest$. The parameters C_1 and C_2 are the acceleration constants that reflect the weighting of the stochastic acceleration term that pull each particle toward $pbest$ and $gbest$ respectively. The inertia weight w is used to facilitate both global and local search. Large w facilitate global search while smaller values favored local search. In this study w was made to decrease exponentially as the generation progresses. This approach facilitate global search within the early stage (generations) and then start to favor local search as the budget (generation) comes to an end. In standard PSO, all particles learn from its own $pbest$ and $gbest$ for all dimension. Constraining the the social learning aspect to only the $gbest$ lead to premature convergence of the original PSO. Since all particles in the swarm learn from the current $gbest$ even if the $gbest$ is very far from the global optimum, all the particles stand the risk of been attracted to $gbest$ and get trapped in a local optimum especially when solving complex problems with numerous local optimums. To circumvent the problem of premature convergence associated with the standard PSO, a CLPSO was proposed [22]. In CLPSO, instead of particles learning from its $pbest$ and $gbest$ for all dimensions, and for all generations, each element (dimension) of a particle can learn from any other particle's $pbest$ including its own $pbest$. The decision on whether a particle's dimension should learn from its own $pbest$ or other particles' $pbest$ depends on probability P_c called learning probability. Each particle has its own P_c . For every dimension of particle i a random number in the range $[0, 1]$ is generated, if this random number is greater than P_{ci} , the particular dimension will learn from its own $pbest$ otherwise it will learn from another particle's $pbest$. To ensure that at least one of the dimension of each particle learn from another particle's $pbest$, if all dimensions happen to learn from its own $pbest$, one dimension is pick at random and two particles are pick at random from the population, the selected dimension will learn from the corresponding dimension of the particle with the best fitness ($pbest$). In this study, P_{ci} is given by (9) base on an empirical rule stated in [22].

$$V_{i+1}^d = w_i \cdot V_i^d + C_1 \cdot rand1_i^d \cdot (pbest_i^d - X_i^d)$$

$$+ C_2 \cdot rand2_i^d \cdot (gbest_i^d - X_i^d) \quad (7)$$

$$v_{i+1}^d = \phi_1 v_k^t + \phi_2 \mathbf{U} (x_{ib-f}^t - x_i^t) \quad (8)$$

$$P_{ci} = 0.05 + 0.45 \frac{e^{\frac{10(i-1)}{SS-1}} - 1}{(e^{10} - 1)} \quad (9)$$

where SS is the swarm size (number of particles).

V. EXPERIMENT AND SIMULATION DATA

The datasets used in this study were obtained by capturing real world RF signals using universal software radio peripheral 1 (USRP 1) for a period of two months. The USRP are computer hosted software radios with one motherboard and interchangeable daughter board modules for various ranges of frequencies, these serve as the RF front end. Two daughter boards, SBX and Tuner 4937 DI5 3X7901, having a frequency ranges of 4MHz to 4.4GHz and 50 MHz to 860 MHz respectively, were used in this research. The daughterboard perform the function of down-conversion, filtering, analog to digital conversion (ADC) and other signal conditioning while the motherboard decimate the signal to a lower sampling rate that can easily be transmitted to the host computer through a high-speed USB cable where the signal is processed by software. For TV channels with channels bandwidth of 5MHz, we divided the channels into subchannels each consisting of 500 KHz bandwidth. To ensure that no spectral information was lost, we used a sample frequency of 1MHz and obtained 1024 samples for each sample time. For GSM 900 and FM band with a bandwidth of 200 KHz, we used 1MHz sample frequency and 512 samples for each sample time. The power was obtained using both the time and frequency domain data. For the frequency domain, after passing the signal through the channel filter, the signal was windowed using a hamming window in order to reduce spectral leakage. The stream of the data was converted to a vector and decimated. This is then converted to the frequency domain and the magnitudes of the bins were passed to a probe sink. The choice of probe sink is essential because it can only hold the current data and does not increase thereby preventing stack overflow or a segmentation fault. This allows Python to grab the data at run time for further analysis. The interval of time between consecutive sample data was selected at a random value between 5 seconds and 30 seconds. The choice of this range is based on the assumption that for any TV programme, FM broadcast or GSM call will last for not less than 5 to 30 seconds. In order to capture all possible trends, the time between consecutive sample data is selected at random within the given range instead of using regular intervals. For the VHF and FM band we captured RF signals from 54MHz to 110MHz and 470 to 670MHz for the VHF TV bands. For the GSM band, 62 down-link channels (935MHz to 953.3MHz) and 62 uplink channels (890MHz to 908.3MHz) were captured. The real world RF data was divided into three subsets, randomly selected with 60% used for training the ANN, 20% for validation and 20% for testing the trained ANN model. The training or estimation data were the only known data sources used in training the ANN. The test data set was unknown to the network i.e. they are not used in training the network but are used in testing the trained ANN as a measure of the generalization performance of the ANN model. The ANN design, optimization and the simulation

were done in Matlab while the capturing of the data and the signal processing were implemented using gnu-radio which is a combination of Python and C++.

VI. RESULTS

To minimize the MSE of the ANN when tested with the test data, the above listed algorithms were run for 30 independent runs. each run has been continued with 30000 fitness evaluations. A population of 30 individuals was maintained for 1000 generations. After a manual tuning of the parameters, the following parameters are used in this study:

- DE/rand/1/bin, DE/rand/1/exp, DE/best/1/bin, DE/best/1/exp, DE/best/2/bin, DE/best/2/exp, DE/rand-to-best/1/bin and DE/rand-to-best/1/exp has been run with $F = 0.1$ and $CR = 0.2$
- CLPSO and PSO have been run with $C_1 = 1$, $C_2 = 0.8$, $V_{max} = 0.25 * (MaxW - MinW)$, $V_{min} = -V_{max}$, $W_{max} = 1$, $W_{min} = -1$, $MaxW = Maximumweight = 2$, $MinW = Minimumweight = -2$
- BPA has been run with $Epoch = 1200$, $learningrate = 0.01$ $mumentum = 0.008$ the other specifications are shown in Table I.

TABLE I. ANN MODELS SPECIFICATION

	ANN Models		
	Dedicated one channel	Multiple channels, single output	Multiple channels, parallel output
First Hidden Neurons	10	30	30
Second Hidden Neurons	5	25	25
Output Neurons	1	1	20
Number of Channels	1	20	20

Table II shows the numerical results in terms of the MSE obtained using the test data (data not known by the ANN nor used in training the ANN). The final results of each algorithm was obtained by taken the average of the MSE (AMSE) for the 30 independent runs and their corresponding standard deviation (STD). From the results, DE/best/1/bin outperform all the other algorithms, hence it is used as the reference for the Wilcoxon test [23]. A '+' indicate that DE/best/1/bin outperform the other algorithm while '-' mean that the other algorithm outperform the reference. When the Wilcoxon test was perform by changing the reference algorithm, the second best algorithm for this problem is DE/RAND/1/bin follow by inertia PSO. Statistically inertia PSO is the third but the best model with the least MSE of 0.0272 was obtained from the combination of inertia PSO and BPA. To validate the fact that the combination of these population based variants of optimization algorithms with the famous BPA to evolves the weights and biases of ANN will produce a more accurate, robust and generalized model than using only BPA; we use the same topology but train with only BPA at constant learning rate of 0.01 and another one with varying learning rate starting from 0.8 and keep on changing with change in MSE using inbuilt Matlab training function traingda. For both models trained with only BPA, each was run 30 times, each run was constrain within a maximum of 1200 epoch and 6 validation fails, the average

results is as shown in table II. For the hybridized training i.e combining DE or PSO with BPA, the DE or PSO were run for 1000 generations and the final best solution was used to reinitialized the ANN weights and further train using BPA constrained within just 200 epoch and 6 validation fails. In all cases, the hybridize training outperform the training with only BPA. Fig 4, 5 and 6 shows the prediction graphs of some selected channels using test data. This results depict a good generalization of the three models. For this problem, the combination of global search capabilities of DE and swarm intelligence algorithms, and the local search advantages of BPA to evolve the weights of the ANN was found to yield an improved performance as compared to using only the famous BPA.

TABLE II. TEST RESULTS WITH DE/BEST/1/BIN AS REFERENCE

Algorithms	Algorithms		Algorithms + BPA	
	AMSE	STD	AMSE	STD
PSO	0.2811	0.0382	0.0338	(0.0051+)
CLPSO	0.1563	0.0354	0.0346	(0.0035+)
DE/rand/1/bin	0.2685	0.0213	0.0338	(0.0024+)
DE/rand/1/exp	0.3296	0.0453	0.0345	(0.0032+)
DE/best/2/bin	0.1496	0.0354	0.0375	(0.0120+)
DE/best/2/exp	0.3243	0.0400	0.0418	(0.0366+)
DE/best/1/bin	0.1436	0.0335	0.0332	(0.0019)
DE/best/1/exp	0.3376	0.0419	0.0383	(0.0181+)
DE/rand-to-best/1/bin	0.1575	0.0212	0.0340	(0.0031+)
DE/rand-to-best/1/exp	0.5233	0.1233	0.0339	(0.0033+)
BPA (constant learning rate)			0.0441	(0.0055+)
BPA (varying learning rate)			0.2508	(0.0332+)

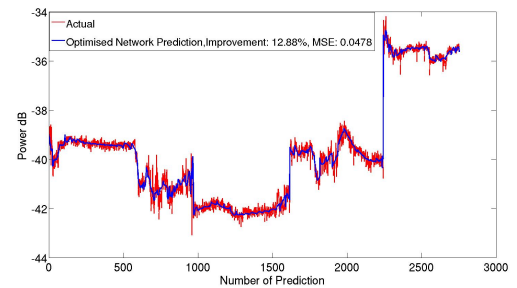


Fig. 4. Feed forward dedicated model prediction

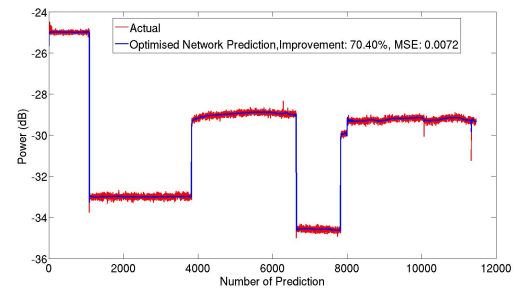


Fig. 5. Cascaded feed forward, parallel output model prediction

VII. CONCLUSION

This paper demonstrate that the accuracy and generalization of ANN can be improved by combining the global search advantages of population based optimization algorithms

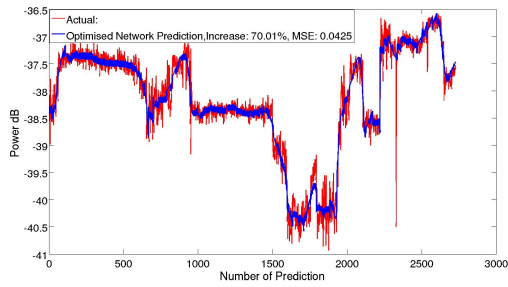


Fig. 6. Multiple channels, single output model prediction

and the local search capability of single solution BPA. This study has unveiled the fact that we can combine population based solutions optimization algorithms with the famous single solution BPA to obtain a more robust ANN model than using only PBA. The prior knowledge of the RF power will enhance noisy channels avoidance not just in cognitive radio network but in any wireless communication thus delivering optimum QoS at less radio resources.

REFERENCES

- [1] FCC, "Federal communication commission notice of inquiry and notice of proposed rule making, in the matter of establishment of an interference temperature metric to quantify and manage interference and to expand available unlicensed operation in certain fixed, mobile and satellite frequency bands," no. 03-237, November 13, 2003.
- [2] S. Haykin, D. J. Thomson, and J. H. Reed, "Spectrum sensing for cognitive radio," in *IEEE Transactions on Cognitive Radio*, May 2009.
- [3] J. Oh and W. Choi, "A hybrid cognitive radio system: A combination of underlay and overlay approaches," in *IEEE Transactions on Cognitive Radio*, 2009.
- [4] C. Stevenson, G. Chouinard, Z. Lei, W. Hu, J. Stephen, and W. Caldwell, "The first cognitive radio wireless regional area network standard," in *IEEE 802.22*, 2009.
- [5] X. Xing, T. Jing, W. Cheng, Y. Huo, and X. Cheng, "Spectrum prediction in cognitive radio networks," in *1536-1284/13/\$25.00 ©2013 IEEE Transactions on Wireless Communications*, April 2013.
- [6] A. M. Wyglinski, M. Nekovee, and Y. T. Hou, *Cognitive Radio Communications and Networks*, 2009.
- [7] R. Azmi, "Support vector machine based white space predictors for cognitive radio," Master's thesis, 2011.
- [8] J. Proakis and D. Manolakis, *Digital Signal Processing Principles, Algorithms, and Applications*, 4th ed. Upper Saddle River, New Jersey 07458: Pearson Prentice Hall, 2007.
- [9] J. Proakis and M. Salehi, *Digital Communications*, 5th ed. Mcgraw-Hill, 2008.
- [10] M. I. Taj and M. Akil, "Cognitive radio spectrum evolution prediction using artificial neural networks based multivariate time series modelling," in *European Wireless, Vienna Austria*, April 2011.
- [11] X. Li and S. A. Zekavat, "Traffic pattern prediction and performance investigation for cognitive radio systems," in *IEEE Communication Society, WCNC Proceedings*, 2008.
- [12] S. Hiremath and S. K. Patra, "Transmission rate prediction for cognitive radio using adaptive neural fuzzy inference system," in *IEEE 5th International Conference on Industrial and Information Systems (ICIIS), India*, Aug 2010.
- [13] S. Geirhofer, J. Z. Sun, L. Tong, and B. M. Sadler, "Cognitive frequency hopping based on interference prediction: Theory and experimental results," vol. 13, no. 2, march 17, 2009.
- [14] L. Giupponi and A. Perez, "Fuzzy-based spectrum handoff in cognitive radio networks," 2008.
- [15] Z. Tabakovic, S. Grgic, and M. Grgic, "Fuzzy logic power control in cognitive radio," in *IEEE transactions*, 2009.
- [16] Z. Lin, X. Jian, L. Huang, and Y. Yao, "Energy prediction based spectrum sensing approach for cognitive radio network," in *978-1-4244-3693-4/09/\$25.00 ©2009 IEEE*, 2009.
- [17] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., 2008.
- [18] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.
- [19] A. K. Qin, V. L. Huang, and P. N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization," in *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, April 2009.
- [20] D. Zaharie, "A comparative analysis of crossover variants in differential evolution," in *Proceedings of the International Multiconference on Computer Science and Information Technology*, 2007, pp. 171–181.
- [21] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 1998.
- [22] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [23] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.