

Application of Artificial Neural Network and Support Vector Regression in Cognitive Radio Networks for RF Power Prediction Using Compact Differential Evolution Algorithm

Sunday Iliya, Eric Goodyer, John Gow, 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, jgow@dmu.ac.uk
jethros@dmu.ac.uk, mgongora@dmu.ac.uk

Abstract—Cognitive radio (CR) technology has emerged as a promising solution to many wireless communication problems including spectrum scarcity and underutilization. To enhance the selection of channel with less noise among the white spaces (idle channels), the a priori knowledge of Radio Frequency (RF) power is very important. 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) and Support Vector Regression (SVR) models for the prediction of real world RF power within the GSM 900, Very High Frequency (VHF) and Ultra High Frequency (UHF) FM and TV bands. Sensitivity analysis was used to reduce the input vector of the prediction models. The inputs of the ANN and SVR consist of only time domain data and past RF power without using any RF power related parameters, thus forming a nonlinear time series prediction model. 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. Since CR are embedded communication devices with memory constrain limitation, the models used, implemented a novel and innovative initial weight optimization of the ANN's through the use of compact differential evolutionary (cDE) algorithm variants which are memory efficient. This was found to enhance the accuracy and generalization of the ANN model.

Index Terms—Cognitive Radio; Primary User; Artificial Neural Network; Support Vector Machine; Compact Differential Evolution; RF Power; Prediction.

I. INTRODUCTION

DUE TO the current static spectrum allocation policy, most of the licensed radio spectrum are not maximally utilized and often free (idle) while the unlicensed spectrum are overcrowded. Hence the current spectrum scarcity is the direct consequence of static 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] [3]. In

this paper, we focused on the study of real world RF power distribution in some 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, FM band, and the GSM 900 band. The problem of spectrum scarcity and underutilization, can be minimized by adopting a new paradigm of wireless communication scheme. Advanced Cognitive Radio (CR) network or Adaptive Spectrum Sharing (ASS) is one of the ways to optimize our wireless communications technologies for high data rates in a dynamic environment while 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 [4]. CR is an intelligent radio where many of the digital signal processing that were traditionally done in static hardware are implemented via software. Irrespective of the definition of CR, it has the followings basic features: observation, adaptability and intelligence. CR is the key enabling tool for dynamic spectrum access and a promising solution for the present problem of spectrum scarcity and underutilization. Cognitive radio network is made up of two users i.e. the license owners called the primary users (PU) who are the incumbent legitimate owners of the spectrum and the cognitive radio commonly called the secondary users (SU) who intelligently and opportunistically access the unused licensed spectrum based on some agreed conditions. CR access to licensed spectrum is subject to two constrains i.e on no interference base, this implies that CR can use the licensed spectrum only when the licensed owners are not using the channel (the overlay CR scheme). The second constrain is on the transmitted power, in this case, 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) [5] [1]. There are four major steps involved in cognitive radio network, these are: spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility [6] [7].

In spectrum sensing, the CR senses the PU spectrum using either energy detector, cyclostationary features detector, cooperative sensing, match filter detector, eigenvalue detector, etc to sense the occupancy status of the PU [8]. Based on the sensing results, the CR will take a decision using a binary classifier to classify the PU channels (spectrum) as either busy or idle there by identifying the white spaces (spectrum holes or idle channels). Spectrum sharing deals with efficient allocation of the available white spaces to the 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 channels when the PU reclaimed ownership of the channel and search for another spectrum hole to communicate. During the withdrawal or search 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 enhanced with cognitive capability for optimum spectrum utilization. CRs are embedded wireless communication devices with limited memory, thus in this paper, we utilized the power of compact differential evolutionary (cDE) algorithm which is memory efficient, to develop an optimized ANN and SVR model for the prediction of real world radio frequency (RF) power. RF power traffics is a function of time, geographical location (longitude and latitude), height above the sea level (altitude) and the frequency or channels properties. Since our experiment is conducted at a fixed geographical location and at constant height, the inputs of the ANN and SVR consist of only past RF power samples, current time domain information and frequency (channel) while the output is the predicted current RF power in decibel (dB) (i.e. the current RF power is modelled as a function of time, frequency and past RF power samples) hence forming a nonlinear time series prediction model. ANN and SVR models were adopted because of the dynamic nonlinearity often associated with RF traffic pattern, coupled with random interfering signals or noise resulting from both artificial and natural sources. The use of sensitivity analysis as detailed in Section VIII for the determination of the optimum number of past recent RF power samples to be used as part of the input of the ANN or SVR for prediction of current RF power, results into a more compact, robust, accurate, and well generalized models. The proposed algorithm used a priori data to enable the system to avoid noisy channels. The prior 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, and 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 for scripting and C++ for signal processing blocks; while the 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.

Many prediction models used in CR radio uses known RF

related parameters as their inputs of which licensed owners will not be willing to dispose such information to CR users. Some of the models are based on explicit mathematical model which may be different from real world situation as highlighted in Section II. Some of the prediction models aim at prediction of spectrum holes, but the fact that spectrum holes (vacant channels) are known does not depict any information about the best channel to be used among the idle channels as the noise level is not flat for all the channels. Thus the major contribution of our model is that it can be used for Rf power prediction where the CR has no prior knowledge of any RF power related parameter. This will enable the CR to avoid noisy channels. The model is trained and tested using real world data. Also instead of training the ANN using back propagation algorithms (BPA) which often lack optimality due to premature convergent, the weights of the ANN are initially evolves using cDE and then fine tune using BPA, this was found to produce a more accurate and generalized model as compared with the one trained using only BPA. SVR was also examined using different kernels and we come up with the model that is more appropriate for our studied location.

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

II. RELATED WORK

There are different types and variants of Computational Intelligence (CI) and machine learning algorithms that can be used in CR such as genetic algorithms for optimization of transmission parameters [9], swarm intelligence for optimization of radio resource allocation [10], fuzzy logic system (FLS) for decision making [11] [12], neural network and hidden Markov model for prediction of spectrum holes; game theory, linear regression and linear predictors for spectrum occupancy prediction [13], Bayesian inference based predictors, etc. Some of the CI methods are used for learning and prediction, some for optimization of certain transmission parameters while others for decision making [14]. TV idle channels prediction using ANN was proposed in [15], however, data were collected only for two hours everyday day (5pm to 7pm) within a period of four weeks, this is not sufficient to capture all the various trends associated with TV broadcast. Also, identifying the idle channels does not depict any spatial or temporal information of 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 idle channels. Spectrum hole prediction using Elman recurrent artificial neural network (ERANN) was proposed in [16]. 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 modelled using ideal multivariate time series equations, which are often different from real life RF traffics where PU signals can be embedded in noise and/ or interfering signals. Traffic pattern

prediction using seasonal autoregressive integrated moving-average (SARIMA) was proposed for reduction of CRs hopping rate and interference effects on PU while maintaining a fare blocking rate [17]. The model (SARIMA) does not depict any information about the expected noise power.

Fuzzy logic (FL) is a CI method that can capture and represent uncertainty. As a result it has been used in CR research for decision making processes. In [11] an FL based decision-making system with a learning mechanism was developed for selection of optimum spectrum sensing techniques for a given band. Among these techniques are matched filtering, correlation detection, features detection, energy detection, and cooperative sensing. Adaptive neural fuzzy inference system (ANFIS) was used for prediction of transmission rate [18]. This model was designed to predict the data rate (6, 12, 24, 36, 48 and 54 Mbps) that can be achieved in wireless local area network (WLAN) using a 802.11a/g configuration as a function of time. The training data set was obtained by generating a random data rate with an assigned probability of occurrence at a given time instance, thus forming a time series. In this study, real world RF data wasn't used. More importantly, the research did not take into account the dynamic nature of noise or interference level which can affect the predicted data rates. Semi Markov model (SMM) and continuous-time Markov chain (CTMC) models have also been used for the prediction of packet transmission rates [19]. This avoids packet collisions through spectrum sensing and prediction of temporal WLAN activities combined with hopping to a temporary idle channel. However, SMM are not memory efficient, neither was there any reference made to the expected noise level among the inactive (idle) channels to be selected. An FL based decision system was modeled for spectrum hand-off decision-making in a context characterized by uncertain and heterogeneous information [12] 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 [20]. 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 [21]. The EMA achieved a prediction average mean square error (MSE) of 0.2436 with the assumption that the channel utilization follow exponential distribution with rate parameter $\lambda = 0.2$ 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 SVR and an ANN trained using cDE 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 capabilities of cDE algorithm variants and the local search advantages of back-propagation algorithms (BPA). The initial weights of the ANN were evolved using cDE after which the ANN was trained (fine tune) 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. The meta parameters that govern the accuracy and generalization SVR model were evolves using cDE.

III. ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks (ANN) are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems [22], [23]. Due to the dynamic 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 are nonlinear hyperbolic tangent functions (1), and a linear symmetric straight line (2) is used for the output activation function. Implementation with other activation functions were also adopted, but this choice gave a better promising results. The nonlinear hyperbolic tangent 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 compact differential evolutionary algorithms variants after which the weights are further fine tuned using backpropagation algorithm (BPA). The training objective function is the minimization of the mean square error (MSE) i.e. the synaptic weights and biases were updated every epoch to minimize the 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. In this study, the back propagation algorithm is used as a local searcher, thus the learning rate was kept low at 0.01. The inputs of the ANN consist of seven past recent RF power, and 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 while the output gives the power in Decibels (dB). Each input of the time domain, enables the ANN to keep track with the trend of RF power variation as a function of that particular input. The current RF power is modelled as a nonlinear function of recent past RF power samples and current time, thus forming a nonlinear time series model. The number of past samples to be used (in this study 7) for reliable prediction and efficient memory management was obtained experimentally as detailed in Section VIII. The actual past RF power (not the predicted RF power) samples fed at the input of the ANN, coupled with the long time training information captured via the time domain inputs, results in a robust ANN model that adapt well to the present trend of RF power variation. 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 RF power prediction of only one channel which implies that each channel will have its own dedicated ANN. To circumvent this problem, we designed two models for RF power prediction in multiple channels. The second model depicted in Fig 2 is used for prediction of RF power in many channels (for this study is 20 channels) but one at a time. It has only one output neuron, but in addition to the time and past RF power samples inputs, it has another inputs representing the channels. The output neurons of the third (parallel) model is equal to the number of channels to be considered Fig 3. The parallel model is used for simultaneous prediction of RF power in multiple channels given the current time instant and past RF power samples as inputs. For the parallel model, if 7 recent past samples of each of the channels

were used as distinct feedback inputs, there will be a total of $7N$ feedback inputs; where N is the number of channels Fig 3; and the training will be computationally expensive. These large feedback inputs were reduced to 7 by using their average. The data used in this study were obtained by capturing real world RF signals within the GSM 900, VHF and UHF TV 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.

Artificial neural network architecture can be broadly classified as either feed forward or recurrent type [22]. Each of these two classes can be structured in different configurations. A feed forward network is one in which the output of one layer is connected to input of the next layer via a synaptic weight, while the recurrent type may have at least one feedback connection or connections between neurons within the same layer or other layers depending on the topology (architecture). The training time of the feed forward is less compared to that of the recurrent type but the recurrent type has better memory capability for recalling past events. Four ANN topologies were considered: feed forward (FF), cascaded feed forward (CFF), feed forward with output feedback (FFB), and layered recurrent (LR) 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 compact differential evolutionary algorithm variants. The first generation initial weights and biases were randomly generated and constrained within the decision space of -2 to 2 . After 1000 generations, the ANN weights and biases were initialized using the elite i.e. the most fittest solution (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 the training Section VI-A. Thus producing the final optimized ANN model.

$$F(x) = b \cdot \tanh(ax) = b \left(\frac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}} \right) \quad (1)$$

$$F(x) = mx + c \quad (2)$$

Where the intercept $c = 0$ and the gradient m is left at Matlab default while the constants a and b are assigned the value 1.

IV. SUPPORT VECTOR MACHINE

Support vector machine (SVM) used for regression is often known as support vector regression (SVR). In SVR, the input space x is first mapped onto a high m dimensional feature space by means of certain non-linear transformation (mapping), after which a linear model $f(x, w)$ is constructed in the feature space as shown in (4), [22]. Many time series regression prediction models uses certain loss functions during the training phase for minimization of the empirical risk, among these loss functions are mean square error, square error

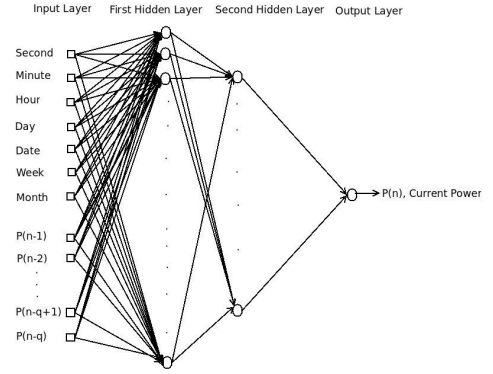


Fig. 1: Dedicated ANN model for one channel

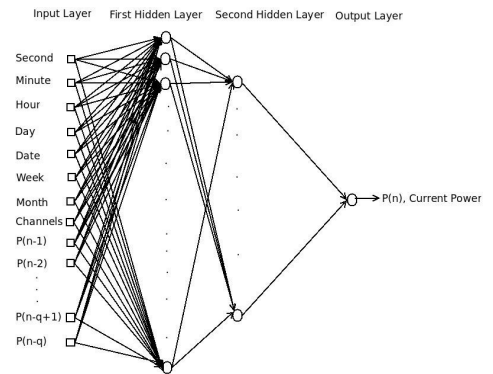


Fig. 2: Multiple channels, single output ANN model

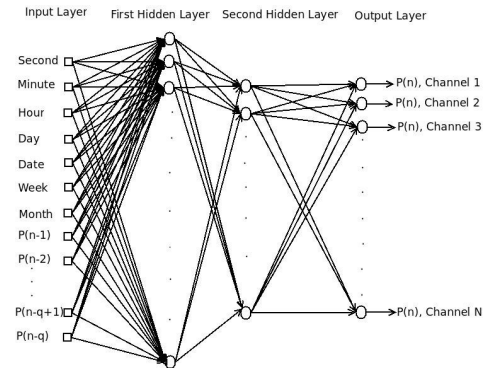


Fig. 3: Multiple channels, parallel outputs ANN model

Where n is a time index, $P(n-1), P(n-2), \dots, P(n-q)$ are the past q RF power samples while $P(n)$ is the current predicted RF power.

and absolute error. In SVM regression, a different loss function called ε -insensitive loss proposed in [24] [25], is used. When the error is within the threshold ε , it is considered as zero, beyond the threshold ε , the loss function (error) is computed as the difference between the actual error and the threshold as depicted in (5). The empirical risk function of support vector regression is as shown in (6). The goal of SVR model is to

approximate an unknown real-value function depicted by (3). Where x is a multivariate input vector while y is a scalar output, and δ is independent and identically distributed (i.i.d.) zero mean random noise or error. The model is estimated using a finite training samples (x_i, y_i) for $i = 1, \dots, n$ where n is the number of training samples. For this study, the input vector x of the SVR model consist of past recent RF power, current time and frequency while the scalar output y is the current power in Decibels (dB).

$$y = r(x) + \delta \quad (3)$$

$$f(x, w) = \sum_{j=1}^m w_j g_j(x) + b \quad (4)$$

Where $g_j(x)$, $j = 1, \dots, m$ refer to set of non-linear transformations, w_j are the weights and b is the bias.

$$L_\varepsilon(y, f(x, w)) = \begin{cases} 0 & \text{if } |y - f(x, w)| \leq \varepsilon \\ |y - f(x, w)| - \varepsilon & \text{otherwise} \end{cases} \quad (5)$$

$$R_{emp}(w) = \frac{1}{n} \sum_{i=1}^n L_\varepsilon(y_i, f(x_i, w)) \quad (6)$$

Support vector regression model is formulated as the minimization of of the following objective functions, [22]:

$$\text{minimise } \frac{1}{2} \|W\|^2 + C \sum_{i=1}^n (\xi_i + \xi_i^*) \quad (7)$$

$$\text{subject to } \begin{cases} y_i - f(x_i, w) - b \leq \varepsilon + \xi_i^* \\ f(x_i, w) + b - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, n \end{cases} \quad (8)$$

The non-negative constant C is a regularization parameter that determined the trade off between model complexity (flatness) and the extend to which the deviations larger than ε will be tolerated in the optimization formulation. It controls the trade-off between achieving a low training and validation error, and minimizing the norm of the weights. Thus the model generalization is partly dependent on C . The parameter C enforces an upper bound on the norm of the weights, as shown in (9). Very small value of C will lead to large training error while infinite or very large value of C will lead to over-fitting resulting from large number of support vectors, [26]. The slack variables ξ_i and ξ_i^* represent the upper and lower constrains on the output of the system. These slack variables are introduced to estimate the deviation of the training samples from the ε -sensitive zone thus reducing model complexity by minimizing the norms of the weights, and at the same time performing linear regression in the high dimensional feature space using ε -sensitive loss function. The parameter ε controls the width of the ε -insensitive zone used to fit the training data. The number of support vectors used in constructing the support vector regression model (function) is partly dependent on the parameter ε . If ε -value is very large, few support vectors will be selected, on the contrary, bigger ε -value results in a more generalized model (flat estimate). Thus both the complexity and the generalization capability of the network depend on its value. One other parameter that can affect the generalization

and accuracy of a support vector regression model is the kernel parameter and the type of kernel function used as shown in (11) to (14).

There are three meta-parameters or hyperparameters that determine the complexity, generalization capability and accuracy of support vector machine regression model, these are the C Parameter, ε and the kernel parameter γ , [27], [28], [29]. Optimal selection of these parameters is further complicated due to the fact that they are problem dependent and the performance of the SVR model depends on all the three parameters. There are many proposals how these parameters can be chosen. It has been suggested that these parameters should be selected by users based on the users experience, expertise and a priori knowledge of the problem, ([24], [25], [30], [31]). This leads to many repeated trial and error attempts before getting the optimums if possible, and it limit the usage to only experts. In this study, we used cDE to evolves the three meta parameters of the SVR model. SVR optimization problem constitute a dual problem with a solution given by

$$f(x) = \sum_{i=1}^s (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (9)$$

The dual coefficients in (9) are subject to the constrains $0 \leq \alpha_i \leq C$ and $0 \leq \alpha_i^* \leq C$. Where s is the number of support vectors, $K(x, x_i)$ is the kernel function, b is the bias, while α and α^* are Lagrange multipliers. The training samples x with non-zero coefficients in (9) are called the support vectors. The general expression for the kernel is depicted in (10). Any symmetric positive definite function, which satisfies Mercers Conditions [22] can be used as kernel function. In this study, four kernel were used, i.e. the Radial Basis Function (RBF), Gaussian Radial Basis Function, Exponential Radial Basis Function kernel and Linear kernel given by (11), (12), (13), and (14) respectively, [22]. In this study, we designed two SVR models for each kernel, one of the model shown in Fig. 4 is dedicated for prediction of RF power of only one channel or resource block which implies that each channel will have it own model; the second model shown in Fig. 5 has an additional channel input thus it can be used for prediction of RF power in many channels but one at a time.

$$K(x, x_i) = \sum_{j=1}^m g_j(x) g_j(x_i) \quad (10)$$

$$K(x, x_i) = e^{(-\gamma \|x - x_i\|^2)} \quad (11)$$

$$K(x, x_i) = e^{(-\frac{\|x - x_i\|^2}{2\sigma^2})} \quad (12)$$

$$K(x, x_i) = e^{(-\frac{\|x - x_i\|}{2\sigma^2})} \quad (13)$$

$$K(x, x_i) = x^T x_i + c \quad (14)$$

The adjustable constant parameter c of the linear kernel was set to 0 while the kernel parameters σ and γ were evolved using cDE algorithm variants.

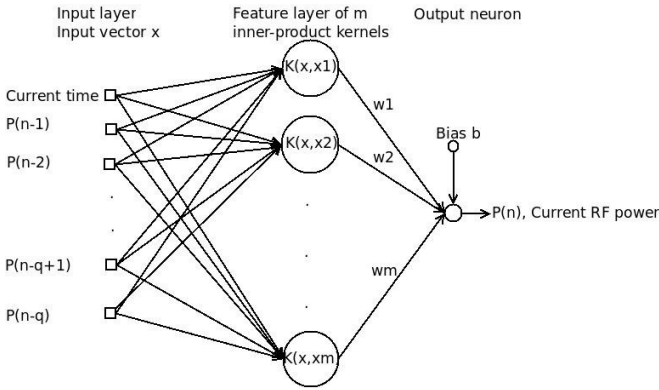


Fig. 4: Dedicated SVR model for one channel

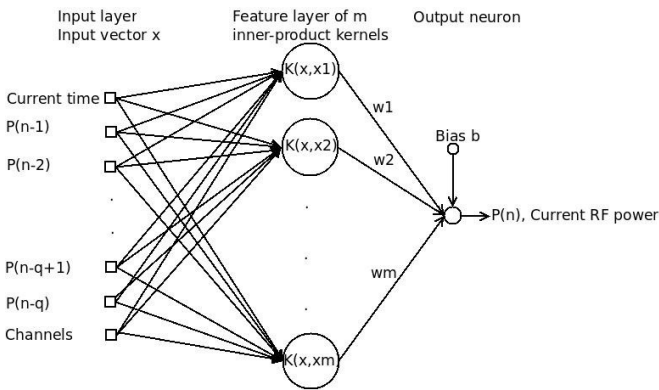


Fig. 5: Multiple channels, single output SVR model

Where n is a time index, $P(n-1), P(n-2), \dots, P(n-q)$ are the past q RF power samples while $P(n)$ is the current predicted RF power.

V. OPTIMIZATION ALGORITHMS

A brief description of the optimization algorithms implemented are presented in this section. We combine the global search capability of compact differential evolutionary algorithms with the single solution local search advantages of BPA to evolve the weights and biases of the optimized ANN model as described in the training, Section VI-A.

A. Differential Evolution

Base on the original definition, DE are population based direct search algorithms used to solve continuous optimization problems [32] [33]. 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 (15), [32].

$$X_{i,G} = X_{min} + rand(1,0) \cdot (X_{max} - X_{min}) \quad (15)$$

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 candidates solution (individuals or 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 as shown in (16) to (20), [32]:

- DE/rand/1:

$$V_{i,G} = X_{r_1,G} + F \cdot (X_{r_2,G} - X_{r_3,G}) \quad (16)$$

- DE/best/1:

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) \quad (17)$$

- DE/rand-to-best/1:

$$V_{i,G} = X_{i,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - X_{r_2,G}) \quad (18)$$

- DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \quad (19)$$

- DE/rand/2

$$V_{i,G} = X_{r_5,G} + F \cdot (X_{r_1,G} - X_{r_2,G}) + F \cdot (X_{r_3,G} - X_{r_4,G}) \quad (20)$$

Where the indexes r_1, r_2, r_3, r_4 and r_5 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 cDE algorithm variants implemented [34]. 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 by (21), [32]:

$$U_{i,G}^j = \begin{cases} V_{i,G}^j & \text{if } (rand(0,1) \leq Cr \text{ or } j = j_{rand}) \\ X_{i,G}^j & \text{otherwise} \end{cases} \quad (21)$$

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 Algorithm 1, [32].

Algorithm 1: Exponential Crossover

```

1:  $U_{i,G} = X_{i,G}$ 
2: generate  $j = randi(1, D)$ 
3:  $U_{i,G}^j = V_{i,G}^j$ 
4:  $k = 1$ 
   while  $rand(0, 1) \leq Cr$  AND  $k < D$  do
6:    $U_{i,G}^j = V_{i,G}^j$ 
    $j = j + 1$ 
8:   if  $j == n$  then
      $j = 1$ 
10:  end if
    $k = k + 1$ 
12: end while
end

```

D. Selection Process

After every generation, the fitness function of each offspring $U_{i,G}$ and the corresponding parent $X_{i,G}$ are computed. A greedy selection schemes is used in which if the fitness function of the offspring is less than or equal to that of its 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) among the final evolved solutions is selected. The DE algorithm pseudocode is depicted in Algorithm 2.

Algorithm 2: Differential Evolution

```

Generate an initial population  $X_{G=0}$  of  $Np$  individuals.
2: Evaluate fitness of each individuals (solutions).
   while termination condition is not met (Generation) do
4:   for  $i = 1$  to  $Np$  do
     Evaluate parent ( $X_{i,G}$ ) fitness .
6:   Generate trial offspring  $V_{i,G}$  by mutation using (16).
     Generate offspring  $U_{i,G}$  by either binomial crossover or exponential crossover.
8:   Evaluate offspring ( $U_{i,G}$ ) fitness
   end for
10:  for  $i = 1$  to  $Np$  do
     Selection Process:
12:   Form the next generation solutions by selecting the best between parents and their offspring
   end for
14: end while
end

```

VI. COMPACT DIFFERENTIAL EVOLUTION

Compact differential evolution (cDE) algorithm is achieved by incorporating the update logic of real values compact genetic algorithm (rcGA) within DE framework [35] [36] [37]. The steps involved in cDE is as follows: A $(2 \times n)$ probability vector \mathbf{PV} consisting of the mean μ and standard deviation σ is generated. where n is the dimensionality of the problem (in this case the number of weights and biases). At initialization, μ was set to 0 while σ was set to a very large value 10, in order to simulate a uniform distribution. A solution called the elite is sampled from the \mathbf{PV} . At each generation (step) other candidate solutions are sampled from the \mathbf{PV} according

to the mutation schemes adopted as described in Section V-B, e.g. for DE/rand/1 three candidate solutions X_{r_1} , X_{r_2} and X_{r_3} are sampled. Without loss of generality, each designed variable $X_{r_1}[i]$ belonging to a candidate solution X_{r_1} , is obtained from the \mathbf{PV} as follows: For each dimension indexed by i , a truncated Gaussian probability density function (PDF) with mean $\mu[i]$ and standard deviation $\sigma[i]$ is assigned. The truncated PDF is defined by (22). The CDF of the truncated PDF is obtained. A random number $rand(0,1)$ is sampled from a uniform distribution. $X_{r_1}[i]$ is obtained by applying the random number $rand(0,1)$ generated to the inverse function of the CDF. Since both the PDF and CDF are truncated or normalized within the range $[-1, 1]$; the actual value of $X_{r_1}[i]$ within the true decision space of $[a, b]$ is obtained as $(X_{r_1}[i] + 1) \frac{(b-a)}{2} + a$. The mutant (provisional offspring) is now generated using the mutation schemes. The offspring is evolved by performing a crossover operation between the elite and the provisional offspring as described in Section V-C. The fitness value of the offspring is computed and compared with that of the elite. If the offspring outperforms the elite, it replaces the elite and declares the winner while the elite is the loser; otherwise the elite is maintained and declares the winner while the offspring is the loser. In this study, the fitness function is the MSE obtained using the test data. The weights and the biases of the ANN are initialized with the offspring and the MSE is obtained, this is repeated using the elite. The one with the least MSE is the winner. The \mathbf{PV} is updated using (23) and (24). Hence in cDE, instead of having a population of individuals (candidate solutions) for every generation as in normal DE, the population are represented by their probability distribution function (i.e. their statistics), thus minimizing the computational complexity, amount of memory needed, and the optimization time. The pseudocode of cDE is as shown in Algorithm 3, [35].

$$PDF(\mu[i], \sigma[i]) = \frac{e^{-\frac{(x-\mu[i])^2}{2\sigma[i]^2}} \sqrt{\frac{2}{\pi}}}{\sigma[i] \left(erf\left(\frac{\mu[i]+1}{\sqrt{2}\sigma[i]}\right) - erf\left(\frac{\mu[i]-1}{\sqrt{2}\sigma[i]}\right) \right)} \quad (22)$$

$$\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_P} (Winner[i] - loser[i]) \quad (23)$$

$$\sigma^{t+1}[i] = \sqrt{(\sigma^t[i])^2 + \delta[i]^2 + \frac{1}{N_P} (Winner[i]^2 - loser[i]^2)} \quad (24)$$

where $\delta[i]^2 = (\mu^t[i])^2 - (\mu^{t+1}[i])^2$, t = steps or generations, N_P is a virtual population and erf is the error function.

Algorithm 3: Compact Differential Evolution Pseudocode

```

generation t=0
2: ** PV Initialization **
   for  $i = 1$  to  $n$  do
4:   Initialize  $\mu[i] = 0$ 
     Initialize  $\sigma[i] = 10$ 
6: end for
   Generate the elite by means of  $\mathbf{PV}$ 
8: while budget condition do
   ** Mutation **

```

```

10:  Generate 3 or more individuals according to the
      mutation schemes e.g.  $X_{r_1}$ ,  $X_{r_2}$  and  $X_{r_3}$  by means
      of PV
      Compute the mutant  $V = X_{r_1} + F \cdot (X_{r_2} - X_{r_3})$ 
12:  ** Crossover **
       $U = V$ , where  $U =$  offspring
14:  for  $i = 1 : N$  do
      Generate rand(0,1)
16:  if  $\text{rand}(0, 1) > Cr$  then
       $U[i] = \text{elite}[i]$ 
18:  end if
      end for
20:  ** Elite Selction **
      [ Winner Loser] = compete( $U$ , elite)
22:  if  $U == \text{Winner}$  then
       $\text{elite} = U$ 
24:  end if
      ** PV Update **
26:  for  $i = 1 : n$  do
       $\mu^{t+1}[i] = \mu^t[i] + \frac{1}{N_P}(\text{Winner}[i] - \text{loser}[i])$ 
28:   $\sigma^{t+1}[i] = \sqrt{(\sigma^t[i])^2 + \delta[i]^2 + \gamma[i]^2}$ 
      Where:  $\delta[i]^2 = (\mu^t[i])^2 - (\mu^{t+1}[i])^2$ 
30:   $\gamma[i]^2 = \frac{1}{N_P}(\text{Winner}[i]^2 - \text{loser}[i]^2)$ 
      end for
32:   $t = t + 1$ 
      end while

end

```

A. Training of ANN and SVM

The objective function in this study is the MSE of the optimized ANN computed using the test data. After every generation, the offspring U_G and the *elite* 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 the elite, the offspring will replace the elite in the next generation otherwise the elite will be maintained. At the end of the generations, the most fittest candidate solution i.e. the final evolved elite; 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 cDE is run 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 satisfied. One of the desirable feature of BPA is it simplicity but it often converges slowly and lack optimality as it can easily be trapped in a local optimum leading to premature convergent. Many approaches has been adopted to solve the problem of premature convergent associated with BPA such as the introduction of momentum constant, varying of the learning rate and retraining of the network with new initial weights. To circumvent the problem of premature convergent, and to have a robust ANN that is well generalized, we combine the global search advantages of cDE optimization algorithm and the local search capability of single solution BPA to evolve the weights

and biases of the ANN. The combination of the global search capabilities of cDE 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 cDE algorithm pseudocode is depicted in Algorithm 3.

In construct to the training of ANN using BPA, the training of SVM is optimal with the optimality rooted in convex optimization. This desired feature of SVM is obtained at the cost of increased computational complexity. The fact that the training of SVM is optimal does not implies that the evolved machine will be well generalized or have a good performance. The optimality here is based on the chosen meta parameters (i.e. C parameter, ε and the kernel parameter γ), the type of kernel function used and the training data. We used the same randomization cDE optimization algorithm variants to evolve the SVM meta parameters while the weights and bias of the SVM were evolves via convex optimization. At each generation, the meta parameters are set using each candidate solution, and the corresponding weights and bias are computed. In order to estimate how the SVM will generalize to an independent dataset (test data), we use two fold cross validation commonly known as holdout method. This has the advantage of having both large training and validation datasets, and each data point is used for both training and validation on each fold. The training data is randomly divided into two sets e.g. A and B of equal size. The SVM was trained on A and test on B , after which it is trained on B and test on A , the average of the MSE for the two test was used as the fitness function for the given sets of meta parameters. At the end of the generations, the SVM is reconstructed using the most fittest meta parameters and tested on the test datasets (data not known by the SVM nor used to train it).

VII. 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-defined radios with one motherboard and interchangeable daughter board modules for various ranges of frequencies. The daughter board modules serve as the RF front end. Two daughter boards, SBX and Tuner 4937 DI5 3X7901, having a continuous frequency ranges of 4MHz to 4.4GHz and 50 MHz to 860 MHz respectively, were used in this research. The daughterboard perform analog operations such as up/down-conversion, filtering, and other signal conditioning while the motherboard perform the functions of clock generation and synchronization, analog to digital conversion (ADC), digital to analog conversion (DAC), host processor interface, and power control. It also 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 8 MHz, we divided the channels into subchannels (resource block) 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 to a lower sampling rate that can easily be processed by the host computer at run time using the inbuilt decimation block in gnu-radio. 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 calls, 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 rather 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++.

VIII. DELAYED INPUTS SENSITIVITY ANALYSIS

In order to examine how many numbers of recent past RF power samples are needed as feedback inputs for reliable prediction, and to have a model with reduced dimensionality of input vector, we carried out a sensitivity analysis. One way of evaluating the importance (significance) of an input in ANN is to measure the Change Of MSE (COM) when the input is deleted or added to the neural network [38]. In this study, the COM method is adopted with the time domain inputs unaltered, and the actual past RF power are added to the input one after the other starting from the most recent one. The ANN is trained with i delay inputs (past RF power samples) and the MSE MSE_i is evaluated. The network is retrained with $i + 1$ delayed inputs, the MSE MSE_{i+1} is obtained and the change in the MSE, $\delta_{mse} = MSE_{i+1} - MSE_i$ is computed as a means of evaluating the importance of the $i + 1$ delay input, for $i = [0 \dots q]$, where q is the total number of past samples used; see Fig 1. Note, δ_{mse} is not computed relative to the MSE obtained when all the q delayed inputs are used as in normal COM method, due to the fact that we don't know the required number of delay inputs q at the start of the experiment; in this case q is obtained by setting a constrain on δ_{mse} . The importance of the inputs are ranked base on the one whose addition causes the largest decrease in MSE as the most important since they are most relevant to the construction of a network with the smallest MSE. In order to justify the importance or ranking of the inputs statistically, for every i

inputs delay, the ANN is trained 20 times, each time with a randomly generated initial weights and biases, the average of the 20 MSE_i is used. The ranking using the normalized values of change in average MSE δ_{mse} as delayed inputs were added is as shown in Fig 7. The graph of the average MSE against number of delayed inputs is as depicted in Fig 6. From Fig 6 and Fig 7, it is obvious that when the number of delay inputs is > 7 , the change in MSE δ_{mse} , is very small. Thus in this study we decided to use 7 past recent RF power samples as part of the ANN inputs for current RF power prediction, taken into cognition the memory constrain of CR as an embedded device.

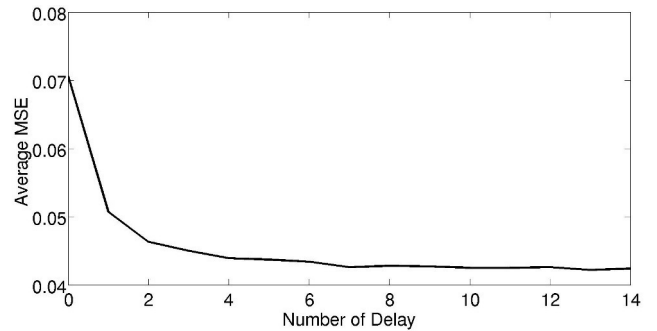


Fig. 6: Sensitive analysis curve

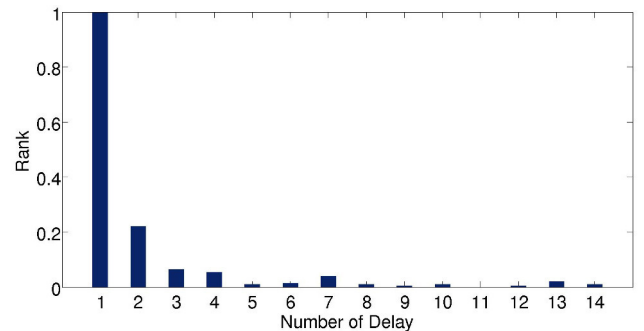


Fig. 7: Past RF power sensitivity ranking

IX. 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 for 1000 generations. After a manual tuning of the parameters, the following parameters are used in this study:

- cDE/rand/1/bin, cDE/rand/1/exp, cDE/rand/2/bin, cDE/rand/2/exp, cDE/best/1/bin, cDE/best/1/exp, cDE/best/2/bin, cDE/best/2/exp, cDE/rand-to-best/1/bin and cDE/rand-to-best/1/exp has been run with $F = 0.1$ and $Cr = 0.3$
- 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	5	15	15
Second Hidden Neurons	3	10	10
Output Neurons	1	1	20
Number of Channels	1	20	20

Tables IV and III 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, the combination of cDE/rand/1/exp and back propagation algorithm (BPA) outperform all the other algorithms with reference to the FFB ANN model while the combination of cDE/rand/1/bin and BPA is the best for the FF ANN model. These two bests are used as the reference for the Wilcoxon test [39]. A '+' indicate that the reference algorithm outperform the other algorithm while "-" mean that the other algorithm outperform the reference. For this problem, when the Wilcoxon test was perform by changing the reference algorithm, the second best algorithm for the FFB ANN model are cDE/rand-to-best/1/exp and cDE/rand/2/exp; both having the same AMSE and STD of 0.0290 and 0.0005 respectively while for FF ANN model, the second best algorithm is cDE/rand/1/exp. For this problem, the FF ANN model trained using cDE/rand/1/bin and BPA emerge as the best compared with other models (FFB, CFF, LR) and algorithms implemented. This implies that the feedback information may have been captured through the inputs assigned to the 7 recent past RF power samples. Comparing the results depicted in tables IV and III with our previous work detailed in [40]; the use of some of the most recent RF power samples as part of the input vector of the ANN, produces a more accurate and robust ANN model with reduced number of neurons. This form a non-linear time series predictive model. The neurons in the model adopted in this study is approximately half of the ones used in [40], thus it has less parameters (weights and biases) to be optimized. To validate the fact that the combination of these cDE 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 depicted in tables IV and III. For the hybridized training i.e combining cDE with BPA, the cDE is run for 1000 generations and the final best solution (elite) was used to reinitialized the ANN weights and further train using BPA constrained within 200 epoch and 6 validation fails. In almost all cases, the hybridize training outperform the training with only BPA. Fig 8, 9 and 10 shows the prediction graphs of some selected channels using test data. These results depict a good generalization of the three models. For this

problem, the combination of the global search capabilities of cDE algorithm variants, 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.

The prediction results of the dedicated SVR model shown in Fig. 4 is depicted in Table IV. From this result, the exponential kernel with meta parameters evolved using cDE/rand/2/exp seem to be more promising with an average MSE of 0.0226, the next best kernel is the linear kernel with AMSE of 0.0301 using cDE/rand/1/bin variant. For multiple channel, single output SVR model Fig. 5, trained for prediction of RF power of 20 resource blocks or channels, the linear kernel emerge as the best with AMSE of 0.0682 this is followed by the RBF kernel with AMSE of 0.0819. the best hyperparameters are evolved using cDE/rand/1/exp for linear kernel and cDE/rand/1/bin and cDE/rand/2/bin for RBF. The results for FF ANN, multiple channel, single output model is also shown in Table with best model having an AMSE of 0.0818 with weights and biases evolved using cDE/best/2/exp and BPA.

TABLE II: Test Results Using FFB ANN Model With DE/best/1/bin as Reference

Algorithms	Algorithms		Algorithms + BPA	
	AMSE	STD	AMSE	STD
cDE/rand/1/bin	0.4055	0.0780	0.0403	(0.0879+)
cDE/rand/1/exp	0.1950	0.0964	0.0242	(0.0007)
cDE/best/2/bin	0.1496	0.0354	0.1999	(0.2405+)
cDE/best/2/exp	0.3243	0.0400	0.1635	(0.2272+)
cDE/best/1/bin	0.1436	0.0335	0.1644	(0.2267+)
cDE/best/1/exp	0.3376	0.0419	0.1808	(0.2352+)
cDE/rand-to-best/1/bin	0.1575	0.0212	0.0359	(0.0275+)
cDE/rand-to-best/1/exp	0.5233	0.1233	0.0290	(0.0005+)
cDE/rand/2/bin	0.0928	0.0119	0.0312	(0.0112+)
cDE/rand/2/exp	0.2087	0.0302	0.0290	(0.0005+)
BPA (constant learning rate)			0.1345	(0.2029+)
BPA (varying learning rate)			0.2508	(0.0332+)

TABLE III: Test Results Using FF ANN With DE/best/1/bin as Reference

Algorithms	Algorithms		Algorithms + BPA	
	AMSE	STD	AMSE	STD
cDE/rand/1/bin	0.3294	0.1664	0.0203	(0.0004)
cDE/rand/1/exp	0.1720	0.1490	0.0204	(0.0006+)
cDE/best/2/bin	0.1181	0.0447	0.0240	(0.0134+)
cDE/best/2/exp	0.1656	0.0833	0.0205	(0.0005+)
cDE/best/1/bin	0.1145	0.0314	0.0206	(0.0015+)
cDE/best/1/exp	0.2121	0.1189	0.0205	(0.0005+)
cDE/rand-to-best/1/bin	0.3102	0.1136	0.0205	(0.0006+)
cDE/rand-to-best/1/exp	0.1646	0.1147	0.0205	(0.0008+)
cDE/rand/2/bin	0.3039	0.1448	0.0214	(0.0042+)
cDE/rand/2/exp	0.1627	0.1050	0.0207	(0.0009+)
BPA (constant learning rate)			0.0267	(0.0004+)
BPA (varying learning rate)			0.0476	(0.0180+)

X. CONCLUSION

This paper demonstrates the power of ANN to produce a robust time series prediction models for RF power traffics in some selected channels. The combination of the global search capabilities of memory efficient cDE and the local search advantages of single solution BPA to evolves the weights and biases of the ANN prediction models, proved to produce a more robust, accurate and well generalise ANN models than

TABLE IV: SVR results using one channel dedicated model

Algorithms	RBF		Gaussian RBF		Exponential		Linear	
	AMSE	STD	AMSE	STD	AMSE	STD	AMSE	STD
cDE/rand/1/bin	0.4055	0.0780	0.0614	0.0198	0.0243	0.0006	0.0301	0.0005
cDE/rand/1/exp	0.1950	0.0964	0.1333	0.0916	0.0298	0.0111	0.0313	0.0012
cDE/best/2/bin	0.0508	0.0174	0.2043	0.1437	0.0350	0.0181	0.0303	0.0003
cDE/best/2/exp	0.0416	0.0078	0.2114	0.0788	0.0437	0.0190	0.0313	0.0005
cDE/best/1/bin	0.0474	0.0015	0.2670	0.0385	0.0266	0.0041	0.0310	0.0007
cDE/best/1/exp	0.0511	0.0035	0.2115	0.1355	0.0323	0.0015	0.0323	0.0015
cDE/rand-to-best/1/bin	0.0477	0.0045	0.2062	0.0856	0.0243	0.0001	0.0314	0.0007
cDE/rand-to-best/1/exp	0.0507	0.0037	0.1346	0.0806	0.0239	0.0014	0.0306	0.0008
cDE/rand/2/bin	1.6718	0.0000	0.1937	0.0946	0.0248	0.0008	0.0306	0.0005
cDE/rand/2/exp	0.2087	0.0302	0.1912	0.0625	0.0226	0.0009	0.0313	0.0015

TABLE V: SVR and ANN results using one output, multiple channel model for 20 channels

Algorithms	Linear		Gaussian RBF		Exponential RBF		RBF		ANN FF	
	AMSE	STD	AMSE	STD	AMSE	STD	AMSE	STD	AMSE	STD
cDE/rand/1/bin	0.0700	0.0033	0.1274	0.0035	0.2422	0.0296	0.0819	0.0093	0.0842	0.0029
cDE/rand/1/exp	0.0682	0.0016	0.1238	0.0025	0.2609	0.0651	0.0848	0.0016	0.0835	0.0028
cDE/best/2/bin	0.1104	0.0012	0.3565	0.3051	0.1472	0.0225	0.1056	0.0084	0.0875	0.0033
cDE/best/2/exp	0.1078	0.0020	0.1609	0.0155	0.1434	0.0136	0.0855	0.0119	0.0818	0.0017
cDE/best/1/bin	0.1112	0.0017	0.6258	0.4953	0.1466	0.012	0.1057	0.0087	0.0869	0.0018
cDE/best/1/exp	0.1112	0.0016	0.3395	0.2010	0.3020	0.2732	0.0947	0.0087	0.0832	0.0032
cDE/rand/best/1/bin	0.1096	0.0019	0.2854	0.0425	0.1361	0.0060	0.0910	0.0040	0.0839	0.0028
cDE/rand/best/1/exp	0.1086	0.0016	0.4062	0.0433	0.1357	0.0035	0.0868	0.0036	0.0851	0.0012
cDE/rand/2/bin	0.1094	0.0019	0.1281	0.0095	0.2345	0.0162	0.0819	0.0074	0.0844	0.0035
cDE/rand/2/exp	0.0687	0.0011	0.1255	0.0036	0.2257	0.0104	0.0902	0.0156	0.0859	0.0042

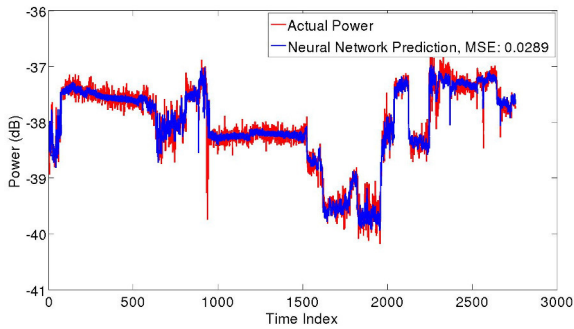


Fig. 9: Cascaded feed forward, parallel output model prediction

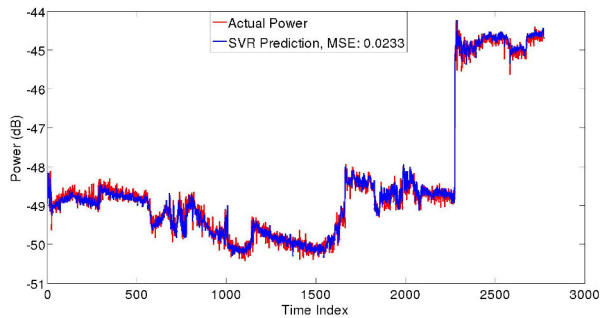


Fig. 10: Multiple channels, single output SVR model prediction

TABLE VI: Best SVR model parameters for Table IV

	Kernel			
	RBF	Gaussian RBF	Exponential RBF	Linear
MSE	0.0416	0.0614	0.0226	0.0301
C	7.72	7.62	22.28	3.76
ϵ	0.000653	0.0003331	0.000228	0.000472
γ or σ	7.38	1.49	7.57	-
Algorithms cDE/	best/2/exp	rand/1/bin	rand/2/exp	rand/1/bin

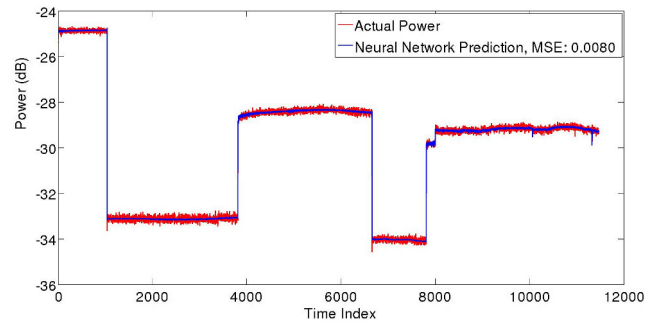


Fig. 8: Feed forward dedicated model prediction

using only BPA for this problem. For the dedicated one channel model, the ANN outperform the SVR model for all the kernels implemented while for the multiple channels, single output model, only the linear kernel SVR model outperform the ANN. The a priori knowledge of the RF power resulting from either communication signals, noise and/or interferences, is not only applicable to cognitive radio network, but in any wireless communication system for noisy channels avoidance.

ACKNOWLEDGMENT

This work is supported by Petroleum Technology Development Fund (PTDF) Scholarship, Nigeria and De Montfort University, United Kingdom.

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] V. Valenta, R. Marsalek, G. Baudoin, M. Villegas, M. Suarez, and F. Robert, "Survey on spectrum utilization in europe: Measurements, analyses and observations," in *Cognitive Radio Oriented Wireless Networks Communications (CROWNCOM), 2010 Proceedings of the Fifth International Conference on*, June 2010, pp. 1–5.
- [3] S. Haykin, D. J. Thomson, and J. H. Reed, "Spectrum sensing for cognitive radio," in *IEEE Transactions on Cognitive Radio*, May 2009.
- [4] J. Oh and W. Choi, "A hybrid cognitive radio system: A combination of underlay and overlay approaches," in *IEEE Transactions on Cognitive Radio*, 2009.
- [5] 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.
- [6] 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.
- [7] A. M. Wyglinski, M. Nekovee, and Y. T. Hou, *Cognitive Radio Communications and Networks*, 2009.
- [8] M. Subhedar and G. Birajdar, "Spectrum sensing techniques in cognitive radio networks: A survey," *International Journal of Next-Generation Networks*, vol. 3, no. 2, pp. 37–51, 2011.
- [9] T. W. Rondeau, B. Le, C. J. Rieser, and C. W. Bostian, "Cognitive radios with genetic algorithms: Intelligent control of software defined radios," in *©2004 SDR Forum, Proceeding of the SDR 2004 Technical Conference and Product Exposition*, 2004.
- [10] S. K. Udgata, K. P. Kumar, and S. L. Sabat, "Swarm intelligence based resource allocation algorithm for cognitive radio network," in *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on*, Oct 2010, pp. 324–329.
- [11] M. Matinmikko, J. Del Ser, T. Rauma, and M. Mustonen, "Fuzzy-logic based framework for spectrum availability assessment in cognitive radio systems," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 11, pp. 2173–2184, November 2013.
- [12] L. Giupponi and A. Perez, "Fuzzy-based spectrum handoff in cognitive radio networks," 2008.
- [13] Y. Chen and H.-S. Oh, "A survey of measurement-based spectrum occupancy modelling for cognitive radios," in *1553-877X ©2013 IEEE IEEE Communications Surveys and Tutorials*, 2013.
- [14] R. Azmi, "Support vector machine based white space predictors for cognitive radio," Master's thesis, 2011.
- [15] O. Winston, A. Thomas, and W. OkelloOdongo, "Optimizing neural network for tv idle channel prediction in cognitive radio using particle swarm optimization," in *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on*, June 2013, pp. 25–29.
- [16] 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.
- [17] X. Li and S. A. Zekavat, "Traffic pattern prediction and performance investigation for cognitive radio systems," in *IEEE Communication Society, WCNC Proceedings*, 2008.
- [18] 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.
- [19] 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.
- [20] Z. Tabakovic, S. Grgic, and M. Grgic, "Fuzzy logic power control in cognitive radio," in *IEEE transactions*, 2009.
- [21] 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.
- [22] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed., 2008.
- [23] Z. Jianli, "Based on neural network spectrum prediction of cognitive radio," in *978-1-4577-0321-8/11/\$26.00 ©2011 IEEE*, 2011.
- [24] V. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York Inc, 1999.
- [25] V. Vapnik, *Statistical learning theory*. New York: Wiley, 1998.
- [26] E. Alpaydin, *Introduction to Machine Learning*, ser. Adaptive computation and machine learning. MIT Press, 2004.
- [27] V. Kecman, *Learning and soft computing*. MIT Press Cambridge, Mass, 2001.
- [28] C. Vladimirov and Y. MA, "Selection of meta-parameters for support vector regression," pp. 687–693, August 2002.
- [29] W. Wenjian, Z. Xu, W. Lu, and X. Zhang, "Determination of the spread parameter in the gaussian kernel for classification and regression," vol. 55, no. 3, pp. 643–663, October 2003.
- [30] V. S. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, 1st ed. New York, NY, USA: John Wiley and Sons, Inc., 1998.
- [31] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2001.
- [32] K. V. Price, R. Storn, and J. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2005.
- [33] 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.
- [34] 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.
- [35] E. Mininno, F. Neri, F. Cupertino, and D. Naso, "Compact differential evolution," *Evolutionary Computation, IEEE Transactions on*, vol. 15, no. 1, pp. 32–54, Feb 2011.
- [36] G. Harik, F. Lobo, and D. Goldberg, "The compact genetic algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 287–297, Nov 1999.
- [37] C. W. Ahn and R. Ramakrishna, "Elitism-based compact genetic algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 4, pp. 367–385, Aug 2003.
- [38] A. H. Sung, "Ranking importance of input parameters of neural networks," *Expert Systems with Applications*, vol. 15, no. 3, pp. 405–411, November 1998.
- [39] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [40] S. Iliya, E. Goodyer, J. Shell, J. Gow, and M. Gongora, "Optimized neural network using differential evolutionary and swarm intelligence optimization algorithms for rf power prediction in cognitive radio network: A comparative study," in *978-1-4799-4998-4/14/\$31.00 ©2014 IEEE International Conference on Adaptive Science and Information Technology*, 2014.