Digital Modulation Classification Using the Bees Algorithm and Probabilistic Neural Network Based on Higher Order Statistics

Article · July 2015

2 authors, including:

Saeed Hakimi
Ferdowsi University Of Mashhad

IJICTR
International Journal of Information & Communication Technology Research
ITRC

IJICTR International Journal of Information & Communication Technology Research

Volume 7- Number 4- Autumn 2015

Topics of Interest

Information Technology

Information Systems
IT Applications & Services
IT Platforms: Software & Hardware Technology
IT Strategies & Frameworks

Communication Technology

Communication Devices
Communication Theory
Mobile Communications
Optical Communications
Satellite Communications
Signal / Image / Video Processing

Network Technology

Computer & Communication Networks
Wireless Networks
Network Management
Network Security
NGN Technology
Security Management

IJICTR

This Page intentionally left blank.

IJICTR International Journal of Information & Communication Technology Research

Volume 7- Number 4- Autumn 2015(1-15)

Digital Modulation Classification Using the Bees Algorithm and Probabilistic Neural Network Based on Higher Order Statistics

Saeed Hakimi
Electrical Engineering Department
Ferdowsi University
Mashhad, Iran
hakimi.saeed@stu-mail.um.ac.ir

Ataollah Ebrahimzadeh
Electrical and Computer Engineering Department
Babol University of Technology
Babol, Iran
e_zadeh@nit.ac.ir

Received: May 2, 2015- Accepted: July 16, 2015

***Abstract*— There has been an increasing demand for automatic classification of digital signal formats during the past decades, which seems to be a continouning trend in future too. Most of the previously proposed classifiers can only classify a few kinds of digital signals and/or a low order of digital signals. In addition, They usually require a high level of Signal to Noise Ratio (SNR). This paper presents a hybrid intelligent system for recognition of digital signal types, including three main modules: a feature extraction module, a classifier module, i.e., a Probabilistic Neural Networks (PNN), and an optimization module. Simulation results validate the high recognition accuracy of the proposed system even at low SNRs.**

Keywords- hybrid system, modulation classification, bees algorithm, probabilstic neural network, higher order statistics.

I. INTRODUCTION

Classifiying the modulation type of the received signal is a requisite of many civilian and military applications such as electronic surveillance, signal confirmation, interference identification, monitoring, spectrum management, software radio, intelligent modem, satellite communication, and etc [1]. In general, an automatic modulation classification system works based on one of these two approaches [1]: the Decision Theoretic (DT) approach or the Pattern Recognition (PR) approach. DT method uses probabilistic hypothesis testing arguments to formulate the recognition problem [1-3]. The major drawbacks of DT approaches are their too complex computations and the lack of robustness against the model mismatches. Furthermore, due to DT approaches limitations, they are not efficient when facing different types of digital signals. PR approaches, however, do not need such careful treatment, so they are easy to implement. The PR method can be further divided into two main subsystems: The feature extraction subsystem and the classifier subsystem. The former extracts the features and the latter determines the memberships of signals.

A variety of modulation recognition techniques has been published in the literature. In [4] the authors introduced a modulation classifier based on the zero-crossing characteristic of the intercepted signal. The considered signal types were: BPSK, QPSK, 8PSK, BFSK, 4FSK, and 8FSK. The decision about the modulation type was based on the variance of the zero-crossing interval sequence, the frequency, and the phase difference histograms. In [5], a technique based on the constellation shape was proposed. This technique used a Fuzzy-C means clustering method for classification of PSK4, PSK8, and QAM16 signals. The accuracy rate of the identification exceeded 90% for SNR >5 dB. In [6], the authors proposed a technique for identifying ASK2, ASK4, PSK2, PSK4, FSK2, and FSK4 signals. The classifier was based on a decision flow. These digital signal types have been identified with a success rate around 90% at SNR=10dB. In [7], the authors proposed a technique based on elementary fourth-order cumulants. In [8], the authors proposed a classifier to discriminate among ASK, 4DPSK, 16QAM, and FSK digital signals. The chosen features were: the kurtosis of the signal, the number of peaks in the phase Probability Density Function (PDF), and the mean of the signal frequency absolute value. A fuzzy classifier was used in this technique. For SNR>5dB, the identifier worked properly. For SNR<5 dB, the performance began to deteriorate. In [9], the authors introduced two classifiers for analog and digital modulation recognition: neural network classifier and fixed threshold classifier. They showed that the neural network classifier perform more efficiently than the threshold classifier. In [10], the authors used the mean and the next three moments of the instantaneous characteristics as the features for signal type classification. They applied different classifiers and showed that the Artificial Neural Network (ANN) has better performance than both KNN and the well-known binary decision trees. They reported a success rate of 90% with SNR range 15–25 dB. In [11], the authors proposed an identifier based on cyclic spectral features for identification of AM, USB, LSB, FM, ASK, FSK, BPSK, QPSK, and SQPSK. It was claimed that cyclic spectrum posses more advantages than power spectrum in signal type recognition. The success rate of this identifier was reported around 90% with SNR range 5–25 dB. In [12], the authors used the features that proposed in [6] and a MLP neural network as the classifier. This identifier showed a success rate about 93% at SNR=8dB for identification of ASK2, ASK4, PSK2, PSK2, FSK2, FSK4, FSK4, and QAM16 digital signals. In [13], the authors proposed four features to classify ASK2, ASK4, PSK2, PSK4, FSK2, and FSK4. The features were extracted based on two main processing steps. The first step was the multiplication of two consecutive signal values. In the second step, the mean and the kurtosis of real and imaginary parts of the quantity obtained in the first step were used as the input features of the classifier. Ref. [14] explored the use of Genetic Programming (GP) in combination with KNN for automatic modulation classification. Four modulation types were considered: BPSK, QPSK, QAM16, and QAM64. Cumulants have been used as input features for GP.

In [15], the authors suggested a low complexity minimum distance centroid estimator to estimate the channel gain and carrier phase jointly. The estimation was achieved by minimizing a signal-to-centroid distance. A New nonparametric likelihood function was proposed for fast classification with unknown noise variance and distribution.

In [16], the authors proposed a method to discover unknown digital amplitude-phase modulations over block-fading additive noise channels. The proposed method applied the iterative Richardson-Lucy algorithm to determine the distribution of the transmitted symbols, which completely characterized the underlying signal constellation. The decoding of the received signals can then be carried out based on the estimate of the signal constellation. In [17], the authors addressed the problem of blind digital modulation identification in time-selective Multiple-Input Multiple-Output channels. The proposed identification process was based on blind source separation and feature classification. A specific multi ANN classifier was adopted to improve the recognition of modulation schemes.

From the published works, it can be clearly observed that design of a system for automatic recognition of digital signal types entails some important issues. If they are suitably addressed, the more robust and efficient recognizers can be developed. One of these issues is related to the choice of the classification approach to be adopted. Literature review shows that techniques, using artificial neural networks as classifiers, outperform the others. In ANNs, the suitable threshold at each node is chosen automatically and adaptively. Furthermore, the time order of the key features does not affect the probability of the correct decision about modulation type of a signal. However, many other algorithms, especially those utilizing the decision-theoretic approach, have to choose the threshold for each key feature and perform with different success rates at a same SNR by applying the extracted key features in a different order in the recognition algorithm [6]. Among the ANNs, perhaps the most widespread neural network architecture is the multilayer perceptron network (MLPN), which usually applies the well-known back propagation algorithm as the learning rule. The back-propagation algorithm of the MLPN does not present a structure which can be easily implemented in a completely parallel manner. For this reason, neural networks which operate in parallel have been proposed. The PNN has been developed in order to respect the requirement of high parallelism.

In this work, the PNN neural network is utilized as a classifier. Assuming the true PDF is smooth, the estimated PDF by classifier approaches the true PDF [18]. Choosing the right features set is another challenging issue. In former papers, usually numerous features used for modulation classification leading to improved efficiency. However, by considering different digital signal types (ASK-PSK-QAM) with higher orders (M>16), the performance degrades and even numerous features are not efficient enough anymore. Indeed, one main reason lead to limitations of most techniques on recognition of digital signal types relates to the features that they utilize. In this work, the feature selection is improved in two aspects: by using the combination of moment and cumulant up to eight orders, that has more efficiency than other types of features, and also by selecting the most effective features from features set by BA [19]. The rest of the paper is organized as follows. Feature extraction module will be described in section 2. Section 3 describes the classifier and its optimization. Section 4 presents the BA and feature selection method. Section 5 illustrates some simulation results. Finally, section 6 concludes the paper.

II. Feature Extraction

Different types of communication signals have different characteristics [20]. Therefore, finding the suitable features for their recognition is a critical problem. In this paper, the following radio signals are considered for recognition: ASK4, ASK8, PSK2, PSK4, PSK8, QAM8, QAM16, QAM32, and QAM64. For simplicity, notations of the mentioned signals are substituted with P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, and P_9 respectively.

Based on our research [7,8,12,15], a combination of higher order moments and higher order cumulants up to eighth provides digital classifiers with improved performances. These features can be applied to characterize the signal PDF. The behavior of higher order moments and cumulants against various transformations is an important factor in determining how useful these quantities may be to characterize signals in systems. The only effect of translation on the received signal is changing its mean. The variance and all higher order moments or cumulants remain unaffected. The rotation of the received signal constellation, caused by multipath or other distortions, affects the relative variances, though certain other parameters are invariant to rotation. The following subsection, briefly describes these features.

A. *Higher order moments and higher order cumulants*

Probability distribution moments are a generalization to the concept of the expected value. The general expression for the i^{th} moment of a random variable (R.V.) is given by [20]:

$$\mu_i = \int_{-\infty}^{\infty} (s-m)^i f(s) ds \tag{1}$$

where m is the mean of the random variable, and $f(.)$ is the PDF of the random variable s. The definition of the i^{th} moment for a finite length discrete R.V. is given by [9]:

$$\mu_i = \sum_{k=1}^{N} (s_k - m)^i f(s_k) \tag{2}$$

where N is the data length. In this study, signals are assumed to be zero mean. Thus:

$$\mu_i = \sum_{k=1}^{N} s_k{}^i f(s_k) \tag{3}$$

Next, the auto-moment of a random variable may be defined as follows:

$$M_{pq} = E[s^{p-q}(s^*)^q] \tag{4}$$

where p is called the moment order, p-q and q represent the number of the non conjugated and conjugated terms, respectively. S^* stands for complex conjugation of S. Considering the signal sequence in form of $s_k = a_k + jb_k$, and using the definition of the auto-moments, the expressions for different orders can be easily derived. For example:

$$\begin{aligned}
M_{62} &= E[s^4(s^*)^2] = E[(a+jb)^4(a-jb)^2] \\
&= E[(a^4 + j4a^3b + j^2 6a^2b^2 + j^3 4ab^3 + b^4 j^4) \\
&\quad (a^2 - j2ab - b^2)] \\
&= E[a^6 + j2a^5b - j^2a^4b^2 - j^3 4a^3b^3 \\
&\quad - j^4a^2b^4 + j^5 2ab^5 + j^6b^6] \\
&= E[a^6 + a^4b^2 - a^2b^4 - b^6]
\end{aligned} \tag{5}$$

Consider a scalar zero mean random variable S with characteristic function:

$$\hat{f}(t) = E\left\{e^{jtS}\right\} \tag{6}$$

Expanding the logarithm of the characteristic function as a Taylor series gives:

$$\log \hat{f}(t) = k_1(jt) + \ldots + \frac{k_r(jt)^r}{r!} + \cdots \tag{7}$$

The constants k_r in (7) are called the cumulants (of the distribution) of S. The symbolism for p^{th} order of cumulant is similar to that of the p^{th} order moment. More specifically:

$$C_{pq} = Cum[\underbrace{s,...,s}_{(p-q)terms}, \underbrace{s^*,...,s^*}_{(q)terms}] \tag{8}$$

For example:

$$C_{81} = Cum(s,s,s,s,s,s,s,s^*)$$

Moments may be expressed in terms of cumulants as:

$$M[s_1,..,s_n] = \sum_{\forall v} Cum\left[\{s_j\}_{j \in v_1}\right] .. Cum\left[\{s_j\}_{j \in v_q}\right] \tag{9}$$

where the summation index is over all partitions $v = (v_1,...,v_q)$ for the set of indices $(1,2,...,n)$, and q is the number of elements in a given partition. $M[.]$ represents moments. The n^{th} order cumulant is a function of the moments of orders up to (and including) n:

$$Cum[s_1,..,s_n] = \sum_{\forall v}(-1)^{q-1}(q-1)!E[\prod_{j\in v_1} s_j]..E[\prod_{j\in v_q} s_j] \quad (10)$$

where $Cum[.]$ means cumulant, and the summation performs on all partitions $v=(v_1,...,v_q)$ for the set of indices $(1,2,...,n)$. Assuming $n=2$, the available set of indices is 1 and 2. Therefore, two different types of partitioning are obtained i.e. $v=(v_1,v_2)$. The partitions are (1, 2) with $q=1$, (1), (2) with $q=2$. Therefore, equation (10) becomes:

$$\begin{aligned} Cum[s_1,s_2] &= (-1)^{1-1}(1-1)!E[s_1 s_2] \\ &+(-1)^{2-1}(2-1)E[s_1]E[s_2] \\ &= E[s_1 s_2]-E[s_1]E[s_2] \end{aligned} \quad (11)$$

In the same manner, cumulant expressions up to eighth order can be computed. For example

$$\begin{aligned} C_{61} &= M_{61} - 10M_{20}M_{41} - 5M_{21}M_{40} + 30M_{20}{}^2M_{21} \\ C_{80} &= M_{80} - 35M_{40}{}^2 - 630M_{20}{}^4 + 420M_{20}{}^2M_{40} \end{aligned} \quad (12)$$

The second, fourth, sixth and eighth order of the moments and cumulant are considered as the features. These features are: M_{20}, M_{21}, M_{40}, M_{41}, M_{42}, M_{60}, M_{61}, M_{62}, M_{63}, M_{80}, M_{81}, M_{82}, M_{83}, M_{84}, C_{20}, C_{21}, C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62}, C_{63}, C_{80}, C_{81}, C_{82}, C_{83}, and C_{84}. It should be noted that considering the orders higher than eight did not influence the system performance sensibly, but complicated the system and increased the computation time. The odd order of the higher order moments is zero. Therefore, the total number of the statistical features is 26, because the M_{20} and M_{21} are equal to C_{20} and C_{21} respectively. All these features are computed for the considered digital signals. Table 1 shows some of the theoretical values of the selected features for a number of the considered digital signal types. These values are computed under the constraint of unit variance in noise free, and are normalized by theoretical signal power. Actually, these computed values are obtained assuming that the signal is clean and of infinite length. However, in practice, signals are usually subject to some types of distortion, either inside the transmitter or during transmission, and have finite length. Figure 1 and Figure 2 show some of the higher order features for a number of considered digital signal types.

Table 1. the values of selected features for a number of considered digital signal types

	ASK4	ASK8	PSK2	PSK4	PSK8	QAM8	QAM16	QAM32	QAM64
M_{41}	1.6400	1.7619	1	0	0	1.1111	-0.6800	0	0
M_{81}	5.2496	7.9211	1	0	0	-1.3580	0	0	0
M_{83}	5.2496	7.9211	1	0	0	3.0864	0	0	0
C_{40}	-1.3600	-1.2381	-2	-1	0	-1	-0.6800	-0.1900	-0.6190
C_{60}	-15.560	-21.882	16	0	0	-1.9259	0	0	0
C_{61}	8.3200	7.1889	16	-4	0	4.8889	2.0800	0	1.7972
C_{62}	8.3200	7.1889	16	0	0	4.7464	0	0	0
C_{63}	8.3200	7.1889	16	4	4	4.8809	2.0800	2.1100	1.7972
C_{80}	-30.086	9.2703	-244	-34	1	-69.358	-13.980	-1.9926	-11.502
C_{84}	-30.086	9.2703	-244	-18	-17	-2	17.379	16.6138	24.1104

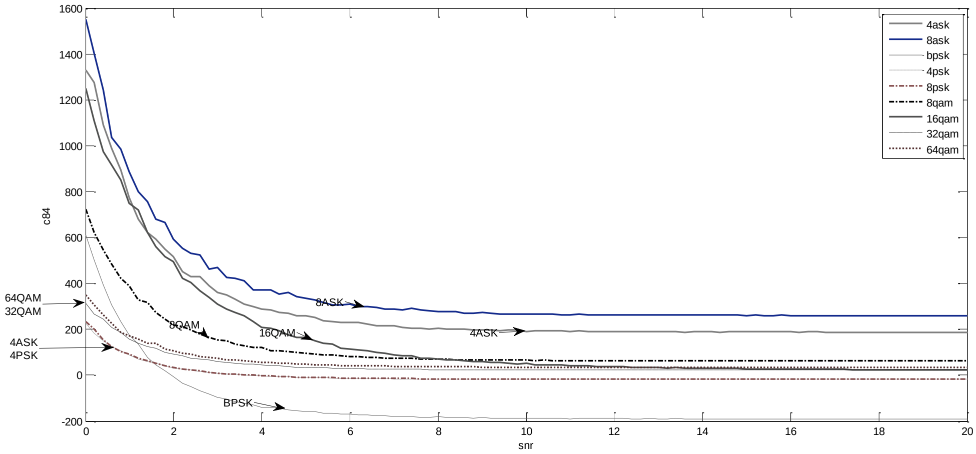

Fig. 1. The amount of C_{84} for some of considered digital signal types.

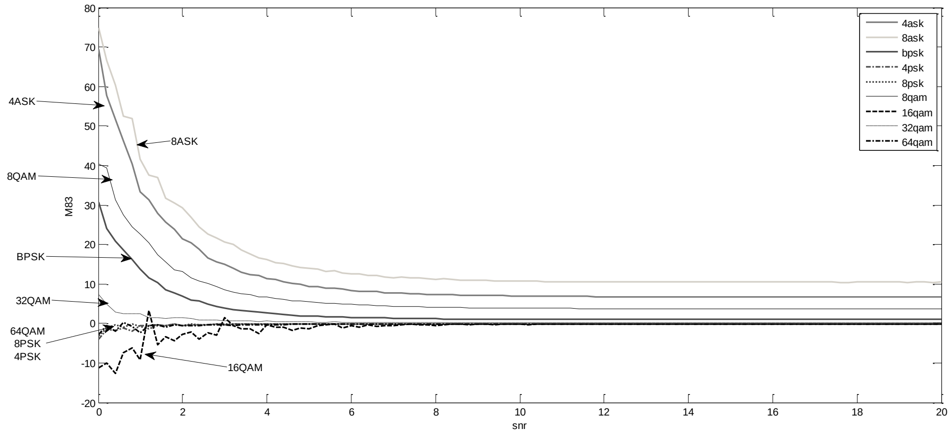

Fig. 2. The amount of M_{83} for some of considered digital signal types.

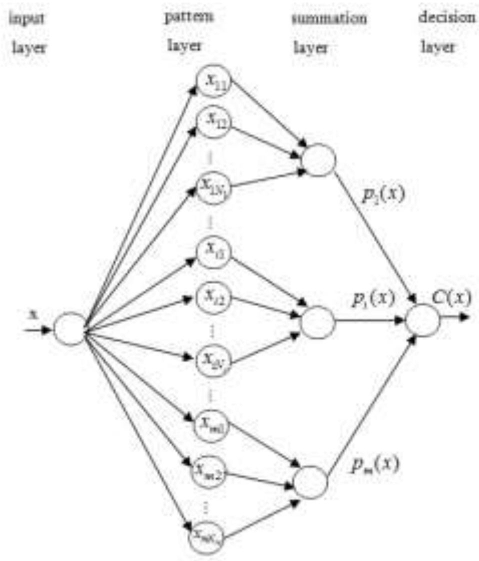

Fig. 3. Block Diagram of a PNN

III. CLASSIFIER

The probabilistic Neural Network (PNN) algorithm is a direct continuation of the work on Bayes classifiers. PNN learns to approximate the PDF of the training examples. More precisely, it is interpreted as a function which approximates the probability density of the underlying examples. The architecture is composed of many interconnected processing units or neurons organized in successive layers. The input layer unit does not perform any computation and simply distributes the input to the neurons in the pattern layer. On receiving a pattern x from the input layer, the neuron x_{ij} of the pattern layer computes its output

$$\phi_{ij}(x) = \frac{1}{(2\pi)^{d/2}\sigma^d}\exp\left[-\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2}\right] \quad (13)$$

where d denotes the dimension of the pattern vector x, σ is the smoothing parameter, and x_{ij} is the neuron vector. The summation layer neurons compute the maximum likelihood of pattern x being classified into C_i by summarizing and averaging the output of all neurons that belong to a same class

$$p_i(x) = \frac{1}{(2\pi)^{d/2}\sigma^d}\frac{1}{N_i}\sum_{j=1}^{N_i}\exp\left[-\frac{(x-x_{ij})^T(x-x_{ij})}{2\sigma^2}\right] \quad (14)$$

where N_i denotes the total number of samples in class C_i. If the apriori probabilities for each class are the same, and also the losses associated with making an incorrect decision for each class are the same, the decision layer unit classifies the pattern in accordance with the Bayes decision rule. This means the classification is based on the output of all the summation layer neurons distribution. A PNN is defined as an implementation of statistical algorithm called Kernel discriminate analysis in which the operations are organized into multilayered feed forward network with four layers. The architecture of a typical PNN is as shown in Fig.3 [21].

$$\hat{C}(x) = \arg\max\{p_i(x)\}, i = 1, 2, ..., m \quad (15)$$

where $\hat{C}(x)$ denotes the estimated class of the pattern x, and m is the total number of classes in the training samples.

The training process of a PNN is essentially the act of determining the value of the smoothing parameter sigma (i.e., the radial deviation of the Gaussian functions). As with RBF networks [22], this parameter needs to be selected to cause a reasonable amount of overlap - too small deviations cause a very spiky approximation which cannot generalize, too large deviations smooth out details. Since, the parameter (the common variance) cannot be determined analytically, the original PNN uses all the training patterns as centers of the Gaussian kernel functions, and assumes a common variance (homoscedastic PNN) [23]. In this method, it is assumed that the smoothing parameter is set to a pre-specified value. However, an appropriate smoothing parameter is often data dependent. Therefore, it requires a proper procedure for smoothing parameter selection as proposed here. When each Gaussian kernel has its own variance, the PNN is heteroscedastic. In the proposed method, depending on input data, an optimal spread value is chosen independently for each class by BA algorithm. This results in low selection error, in this way, the sensitivity of the PNN to smoothing parameter is completely removed.

IV. BEES ALGORITHM

As mentioned, the proposed system uses the BA for optimizing the recognizer. Next phrases describe the BA and optimization of the classifier.

A. *Bees Algorithm*

The BA is an optimization algorithm inspired by the natural foraging behavior of honey bees [19, 24] Figure 4 simply demonstrates the pseudo code for the BA. The BA requires a number of parameters to be set, namely: number of scout bees (n), the number of patches selected out of n visited points (m), the number of elite patches out of m selected patches (e), the number of bees recruited for the best e patches (nep), the number of bees recruited for the other (m-e) selected patches (nsp), the size of patches (ngh), and the stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space. The fitness of the points visited by the scout bees are evaluated in step 2. In step 4, bees that have the highest fitness are designated as “selected bees” and sites visited by them are chosen for neighborhood search. Then, in steps 5 and 6, the algorithm conducts searches in the neighborhood of the selected bees, assigning more bees to search near the best e bees. The bees can be chosen directly according to the fitness associated with the points they are visiting. Alternatively, the fitness values are used to determine the probability of the bees being selected. Searches in the neighborhood of the best e bees, which represent more promising solutions, are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the BA. In step 6, for each site, only the bee with the highest fitness will be selected to form the next bees population. In nature, there is no such a restriction. This constraint is introduced here to reduce the number of points to be explored. In step 7, the remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts to its new population representatives from each selected patch and other scout bees assigned to conduct random searches.

1. Initialize population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) Forming new population.
4. Select elite bees for neighborhood search. Select other bees for neighborhood search.
5. Recruit bees for selected bees and evaluate fitness.
6. Select the fittest bee from each site.
7. Assign remaining bees to search randomly and evaluate their fitness.
8. End While.

Fig. 4. Pseudo code of the basic BA

B. Hybrid Intelligent System (HIS)

As known, the great number of features causes high computational complexity of the recognizer [25]. In addition, some of these features may carry good classification information when treated separately, due to sharing of the same information content there is a little benefit when they combine together. The easiest way to reduce the number of features is a feature selection method. Feature subset selection algorithms can be classified into two categories based on whether or not feature selection is done independently of the learning algorithm used to construct the c1assifier. Filter approaches select salient features only using heuristics based on the intrinsic characteristics of the data. The selection procedure is independent of the estimation or classification process, whereas wrapper approaches embed the estimation classifier as a part of selection procedure and use the estimation of the classifier as feedback to guide the selection direction. Thus, filter approaches are computationally more efficient than wrapper approaches in cases where the original feature number is extremely large. However, for feature selection problems in small and medium size, the filter approaches totally ignore the effects of the selected feature subset on the performance of the classifier used for further estimation (i.e., the feature selection procedure and the classification step do not necessarily optimize the same criterion function). Therefore, in such cases, filter approaches generally result in worse performance than wrapper approaches [25]. In this paper, wrapper type approach has been used.

As mentioned previously, the proposed system employs PNN as classifier. The classification accuracy is used as feedback to the selection process to inform how well a given set of features characterizes patterns. The method requires a data set, comprises patterns, each with N_{tot} features to be utilized in the feature selection process. The classes of all patterns in the training set are known. From the original data sets, new data sets can be constructed in which patterns only contains a subset of the original features. In other words, a pattern in a new data set will have N_s features selected from the original set of N_{tot} features. A bee represents a subset of N_s features and the classifier parameter. A bee can be uniquely identified by a real-binary string (e.g. 52-010110111), where the first part demonstrates the smoothing parameter and the second represents features. The Real part is just one number. The total number of bits in the binary part is N_{tot}, and the total number of non-zero bits is N_s. The position of a bit along the binary string indicates a particular feature. If a feature is selected to form a data set, the corresponding bit is 1. Otherwise, it is zero.

In initial phase, the BA starts with a random generation of a population of real-binary strings that the real number has been generated from a proper interval. The data set is divided in two sets. One data set (training data) is used to train the PNN. The other data set (the test data) is employed to evaluate the classification accuracy of the trained PNN. Neighborhood searching in the BA performs by generating and evaluating neighbors of the fittest bees. It works by just varying slightly a few selected numbers of features and minor changes on the smoothing parameter of best selected answers in that iteration. Various operators could be employed to create neighbors of a given bee, including monadic operators such as mutation, inversion, swap, and insertion (single or multiple). Figure 5 shows the flowchart of the modulation classifier that has been proposed in this work.

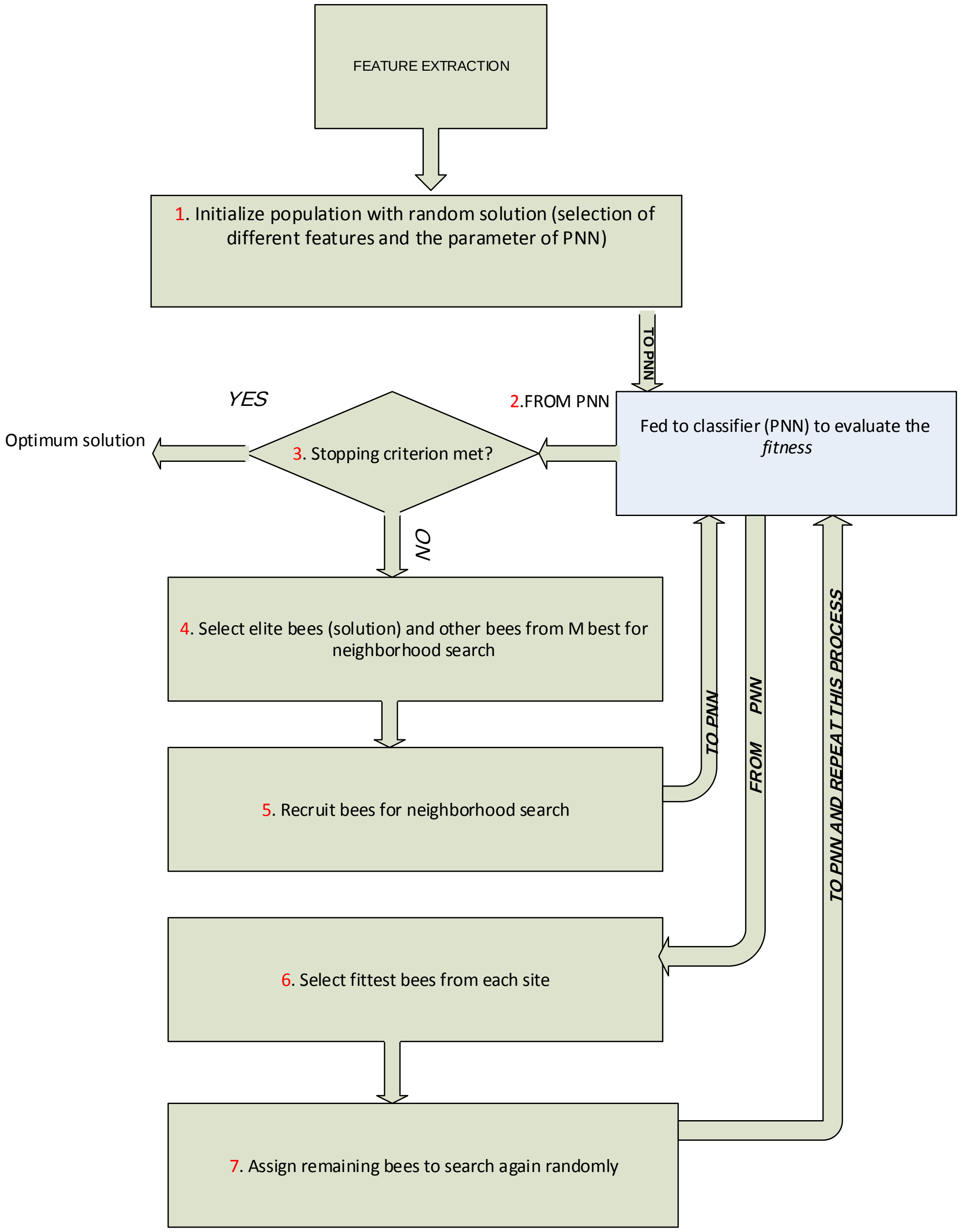

Fig. 5. flow chart of the modulation classifier

V. SIMULATION RESULTS

This section represents simulation results of the considered signals classification. It was supposed that carrier frequencies were estimated correctly, and the signal was heterodyned down. Each signal was generated by MATLAB editor. The digital message was produced randomly for every trial to ensure results would be independent of the message transmitted. Estimating moment and cumulant values for all signal types was based on the theoretical formulas explained in section2. For this process, only the moments and cumulants demonstrating some special characteristics as the class features were selected. The sample frequency was 85.4291 KHz, and the symbol frequency was considered as 10 KHz. A total of 3,000 samples per signal type were created and stored. The estimation was done on a subset of 2000 samples per scheme, out of the total 3,000 samples per scheme dataset. Two different cases were examined. First, the signals were generated noise-free. Second, the signals were distorted by additive white Gaussian noise (AWGN) according to SNRs –2, 0, 2, and 5dB. The PNN parameter (Smoothing parameter) bounded in the interval 10 to 50.

A. *Performance of the Straight Identifier (SI)*

In this subsection, the efficiency of the identifier is evaluated without optimization and feature selection (straight identifier), i.e. full features and the fixed PNN parameter (smoothing parameter) are used. By trial and error method, it was found that optimum value for smoothing parameter is 48. Therefore, this value was considered in simulations. As reported in Table 2, using the MLP neural network with resilient back propagation learning algorithm, 96.28% recognition accuracy has been achieved. While, the recognition accuracy percent of the PNN achieved on the test set was equal to 98.88%. These results were better than those achieved by the SVM-linear and the MLP-back propagation. Indeed, the percent of the MLP-BP and the SVM-Linear methods respectively were equal to 97.15% and 97.22%. Table 3 shows the average efficiency of SI in different SNRs in 10 runs.

recognition accuracy of the recognizer without optimization

classifier	RA(%)
MLP (RPROP)	96.28
MLP (BP)	97.15
SVM (Linear)	97.22
PNN	**98.88**

Table 2. The **average accuracy of si%**

SNR	Efficiency (%)
-2	93.61
0	98.88
2	99.17
5	100

B. *Performance of the identifier using BA optimization*

In this subsection, the efficiency of the proposed identifier using the BA is evaluated. For starting the optimization, one has to specify the number of features that varies from 1 to 26. It is obvious that limiting the number of features will reduce complexity of the identifier and will demand less memory. In this problem, the reduction of feature space was carried out by using the BA feature selection. In general, selecting minimum number of features reduces the recognition accuracy in lower SNRs. However, in this paper, a novel algorithm has been proposed that is the best solution for this problem i.e. reducing the number of features and optimizing the identifier simultaneously. Therewith, the accuracy of the identifier has been improved, and at the same time, its complexity has been reduced. Based on simulation results, it was found out that some features carried useful information in detaching some modulation types, but had weak efficiency in other types. For example, M81 could detach P6, P7, P8, and P9 but confused P3, P4, and P5, whereas C60 denoted P1 ,P2 , P3, and P6 but confused P4, P5, and P7 . BA has given the best identification by assembling the optimum features.

Table 4 demonstrates the selected parameter of PNN in different SNRs, and tables 5-8 show the selected features in different SNRs. These tables illustrate that in SNRs above zero usually algorithms converge to the same features and constant PNN parameter i.e. optimum values. As observed in lower SNRs the average number of features is more than higher SNRs (i.e. the average number of features for SNR=-2 is 14, for SNR=0 is 13, and for SNRs more than two is 10). In addition, the PNN parameter in lower SNRs is smoother than higher SNRs (i.e. the average of smoothing parameter for SNR=-2 is 41.9, for SNR=0 is 27.7, for SNR=2 is 19.9, and for more than SNR=2 is 18).

Table 9 demonstrates the performance of the proposed identifier in different SNRs. The average of the identifier accuracy in SNR=-2 is 94.63%, in SNR=0 is 99.54 %, and for SNRs more than zero is 100%. It is observed that in SNRs above zero the identifier completely detached the modulation types. By increasing SNR from -2 to 0, the accuracy increases about 5%. Tables 10-11 represent the confusion matrix in SNR=-2 and SNR=0. To estimate this matrix, the average of confusion matrices of ten runs was calculated. Table 12 shows the average accuracy of the proposed identifier.

Table 3. The selected parameter of pnn

Run / SNR	1	2	3	4	5	6	7	8	9	10
-2	45	48	44	34	37	44	44	44	45	34
0	11	10	46	38	11	37	11	28	37	48
2	18	18	37	18	18	18	18	18	18	18
5	18	18	18	18	18	18	18	18	18	18

Table 4. The selected features in snr=-2

feature \ run	M_{20}	M_{21}	M_{40}	M_{41}	M_{42}	M_{60}	M_{61}	M_{62}	M_{63}	M_{80}	M_{81}	M_{82}	M_{83}	M_{84}	C_{40}	C_{41}	C_{42}	C_{60}	C_{61}	C_{62}	C_{63}	C_{80}	C_{81}	C_{82}	C_{83}	C_{84}
1	1	0	1	0	1	1	0	1	0	1	1	0	0	1	0	1	1	0	0	0	1	0	0	1	0	1
2	1	0	0	1	1	0	0	0	0	0	1	1	0	1	1	1	1	0	0	1	0	1	0	0	0	0
3	1	0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1
4	1	0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1
5	1	0	0	1	0	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	1	0
6	1	1	1	0	1	0	0	1	1	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	1	1
7	1	0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1
8	1	0	1	1	1	0	0	1	0	1	0	1	0	0	1	1	0	1	0	1	1	1	0	1	0	1
9	1	1	1	1	1	1	0	1	1	1	1	0	0	1	0	1	1	0	0	0	1	0	0	1	1	1
10	1	0	1	0	1	0	1	1	0	1	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1

Table 5. The selected features in snr=0

feature \ run	M_{20}	M_{21}	M_{40}	M_{41}	M_{42}	M_{60}	M_{61}	M_{62}	M_{63}	M_{80}	M_{81}	M_{82}	M_{83}	M_{84}	C_{40}	C_{41}	C_{42}	C_{60}	C_{61}	C_{62}	C_{63}	C_{80}	C_{81}	C_{82}	C_{83}	C_{84}
1	1	0	1	0	1	0	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	0	1
2	0	1	0	1	0	0	1	1	0	1	1	1	0	1	0	0	1	0	1	1	0	0	1	1	0	0
3	1	1	1	1	1	1	1	0	1	1	1	0	1	0	1	1	0	1	1	1	1	1	0	0	1	0
4	0	0	1	0	0	1	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0	1	1	1	0
5	1	0	1	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	1
6	1	0	0	0	0	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	1	0
7	1	0	1	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	1	0	0	1	1	1
8	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1	1	0	0	0	0	1	1	1	1	0	1
9	1	0	0	0	0	1	0	1	1	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	1	0
10	0	1	0	0	0	1	1	1	0	0	0	1	0	1	1	1	1	0	0	1	1	0	1	1	0	0

Table 6. The selected features in snr=2

feature \ run	M_{20}	M_{21}	M_{40}	M_{41}	M_{42}	M_{60}	M_{61}	M_{62}	M_{63}	M_{80}	M_{81}	M_{82}	M_{83}	M_{84}	C_{40}	C_{41}	C_{42}	C_{60}	C_{61}	C_{62}	C_{63}	C_{80}	C_{81}	C_{82}	C_{83}	C_{84}
1	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
2	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
3	1	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	0	0	1	0	0	1	0	1	0	0
4	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
5	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
6	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
7	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
8	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
9	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
10	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1

Table 7. The selected features in snr=5

feature / run	M20	M21	M40	M41	M42	M60	M61	M62	M63	M80	M81	M82	M83	M84	C40	C41	C42	C60	C61	C62	C63	C80	C81	C82	C83	C84
1	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
2	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
3	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
4	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
5	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
6	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
7	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
8	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
9	0	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	1
10	0	0	1	0	0	1	0	0	1	1	0	0	0	1	0	0	1	1	1	1	1	0	0	0	0	1

Table 8. The recognition accuracy of the proposed system

Run / SNR	1	2	3	4	5	6	7	8	9	10
-2	94.72	94.44	95.27	94.16	93.61	94.16	95.55	95.55	92.50	96.38
0	99.44	99.16	99.72	99.44	99.44	99.68	99.44	99.72	100	99.44
2	100	100	100	100	100	100	100	100	100	100
5	100	100	100	100	100	100	100	100	100	100

Table 9. The confusion matrix in snr=-2

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	31.8	8.2							
P_2	10.2	29.8							
P_3			40						
P_4				40					
P_5					39.6				0.4
P_6						40			
P_7					0.2		39.8		
P_8						0.1		39.7	0.2
P_9					0.2			0.1	39.7

Table 10. The confusion matrix in snr=0

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9
P_1	38.8	1.2							
P_2	0.8	39.2							
P_3			40						
P_4				40					
P_5					40				
P_6						40			
P_7							40		
P_8								40	
P_9									40

Table 11. The performance of the proposed system (his)

SNR	Efficiency (%)
-2	94.63
0	99.54
2	100
5	100

C. The effect of the optimization(BA)

In this subsection, the effect of optimization on the proposed system is studied. Figure 6 shows a comparison between performances of the non-optimized classifier and the optimized classifier. It can be seen that the optimization, generally, improves the performances of the classifier for all of SNRs.

In order to compare the performance of BA with other evolutionary algorithms, several algorithms such as Artificial Bee Colony (ABC) [26], Genetic Algorithm (GA) [27], and Particle Swarm Optimization (PSO) [28] are applied to evaluate the proposed method. These results have been obtained in AWGN channel at SNR=0dB. According to results in Table 13, the best accuracy obtained for the test set by BA-PNN is 99.54%. It can be seen that the success rates of BA is higher compared to other nature inspired algorithms. To evaluate the influence of user- defined parameters of BA on its performance and to guide the adequate parameter setting according to the problem at hand, the number of scout bees was changed. It was perceived that by increasing the number of bees the performance would increase. Furthermore, it seemed that there can be an optimum value for this parameter that for each problem it should be found by different methods. Figure 7 shows the performance of the identifier against variations in the number of scout bees at SNR=-4. From simulation results, it is found that the identifier in high SNRs is robust owing to the number of scout bees. Second, the effect of the number of the iterations is surveyed. Figure 8 shows the result of this survey. From figure 8, it can be observed that increasing the number of iterations will improve the accuracy of the recognizer. Other parameters of BA were chosen establishing the same scenario. Table 14 shows parameters values adopted for BA.

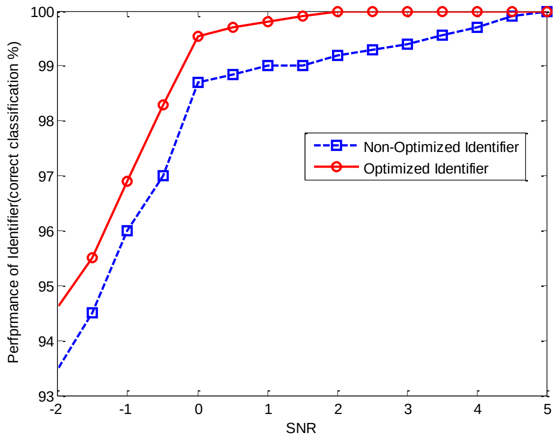

Fig. 6. Comparison between performances of non-optimized identifier and optimized identifier

Table 12. Comparison between performances of ba-pnn and other recognition techniques

Recognition Techniques	Recognition Accuracy (%)
ABC-PNN	94.96
GA- PNN	93.64
PSO- PNN	89.81
BA- PNN	**99.54**

Table 13. The parameters of the BA

BA parameters	Symbol	Value
Number of scout bees	n	25
Number of selected sites	m	6
Number of elite bees	e	4
Initial patch size	ngh	0.1
Number bees around elite points	nep	15
Number of bees around other selected points	nsp	10

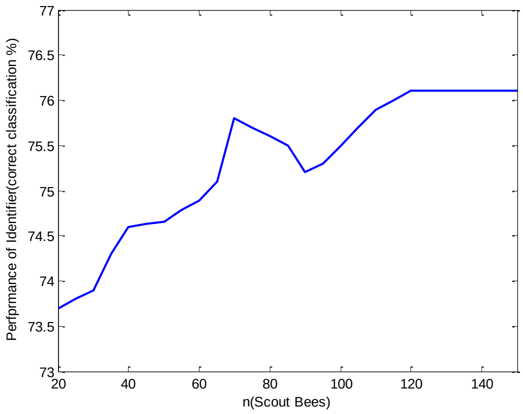

Fig. 7. Sensitivity of accuracy to the number of scout bees in SNR=-4

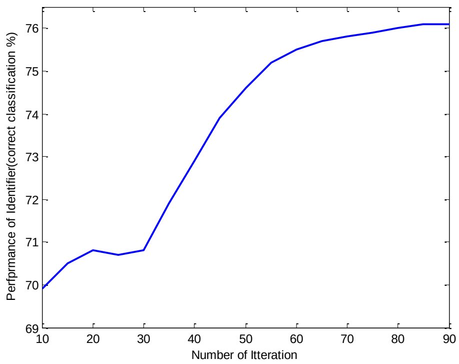

Fig. 8. The effect of iteration on performance of the identifier in SNR=-4

D. Performance comparison

As mentioned in [5], direct comparison with other works is difficult in signal type classification. This is mainly because of the fact that there is no single unified data set available. Table 15 shows the comparison among the important previous papers and the hybrid proposed system. In comparison with other works, the proposed recognizer has many advantages. This system includes a variety of digital signal types. It discloses great generalization ability for classifying the considered digital signal types. The proposed classifier has a success rate of around 95% at SNR =-2 dB. The performance of the classifier is higher than 99% for SNR > 0 dB. In addition, this performance has been achieved with few samples. Results imply that our chosen features manifest efficient properties in signal representation.

Table 14. Comparative study of different works in case of considered modulation, required snr, and recogntion accuracy

Ref	Consider digital signal	SNR(dB)	Recognition accuracy (%)
[4]	BPSK, QPSK, 8PSK, BFSK, 4FSK, 8FSK	15	98
[5]	PSK4, PSK8, QAM16	5	90
[6]	ASK2, ASK4, PSK2, PSK4, FSK2, FSK4	10	90
[7]	BPSK, PAM4, QAM16, PSK8	10	96
[8]	ASK4, 4DPSK, QAM16, FSK	5	90
[11]	AM, DSB, VSB, LSB, USB, FM, PSK2, PSK4, ASK2, ASK4, FSK2, FSK4	15	96
[12]	PSK2, PSK4, PSK8, OQPSK, MSK, QAM16, QAM32, FSK2, FSK4	15-25	93
[13]	AM, USB, LSB, FM, ASK, FSK, BPSK, QPSK, SQPSK	5-15	90
[14]	ASK2, ASK4, PSK2, PSK4, FSK2, FSK4, QAM16	8	93
[29]	PAM4, BPSK, PSK8, QAM16	3	90
[30]	QAM16, QAM32, QAM64, PSK2, PSK8, ASK4, ASK8, FSK4, FSK8, MSK	1	80.2
Proposed hybrid system	ASK4, ASK8, PSK2, PSK4, PSK8, QAM8, QAM16, QAM32, QAM64	0	98

VI. CONCLUSION

Automatic identification of digital signal formats is a crucial task of novel communication systems. In this paper, a hybrid intelligent system was proposed for this purpose with many advantages. In this system, a suitable features set was selected with high efficiency in representing the communication signal formats. As the identifier, a PNN neural network was applied that used the BA as optimizer. Using the mentioned features together with the classifier resulted in a highly efficient recognizer. This recognizer discriminated many various digital signal types with high accuracy even at very low SNRs. However, many features were used for this recognition. In order to reduce the complexity of the recognizer, an optimizer (BA) was applied. This reduced the number of features without trading off the generalization ability and the accuracy. The optimized recognizer also had high performance for recognition of different kinds of digital signals for all considered SNRs. This high efficiency has been achieved with the least number of features, which have been selected using the BA. For future works, this system can be used for recognition of modulation in fading channels. Furthermore, another set of digital signal types can be considered to evaluate the recognition ability of the proposed system.

REFERENCES

[1] Y. Yang, J.-N. Chang, J.-C. Liu, and C.-H. Liu, "Maximum log-likelihood function-based QAM signal classification over fading channels", *Wireless Personal Communications*, vol. 28, pp. 77-94, 2004.

[2] Y. Yang and S. S. Soliman, "A suboptimal algorithm for modulation classification", *IEEE Transactions on Aerospace and Electronic Systems*,vol. 33, pp. 38-45, 1997.

[3] W. Sue, L. X. Jefferson, and Z. Mengchou, "Real-time modulation classification based on maximum likelihood", *IEEE Communications Letters*, vol. 12, pp. 801-803, 2008.

[4] S. Z. Hsue and S. S. Soliman, "Automatic modulation classification using Zero-Crossing", *IEEE Proceedings Radar Sonar and Navigation*, vol. 137, pp. 459-464, 1990.

[5] B. G. Mobasseri, "Digital modulation classification using constellation shape", *Signal Processing*, vol. 80, pp. 251-277, 2000.

[6] E. E. Azzouz and A. K. Nandi, "Automatic identification of digital modulation types", *Signal Processing*, vol. 47, pp. 55-69, 1995.

[7] A.Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants", *IEEE Transactions on Communications*, vol. 48, pp. 416-429, 2000.

[8] J. Lopatka and M. Pedzisz, "Automatic modulation classification using statistical moments and a fuzzy classifier", *in Proceedings of the 5th International Conference on Signal Processing (ICSP'00)*, Beijing, China, vol. 3, pp. 1500-1506, August 2000.

[9] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals", *IEEE Transactions on Communications*, vol. 46, pp. 431-436, 1998.

[10] C. Louis and P. Sehier, "Automatic modulation recognition with neural network", *in Proceedings of the IEEE Military*

Communications Conference (MILCOM'94), Fort Monmouth, NJ ,USA, vol. 3, pp. 713-717, October 1994.

[11] L. Mingquan, X. Xianci, and L. Leming, "Cyclic spectral features based modulation recognition", *in Proceedings of the the International Conference on Communications Technology (ICCT' 96)*, Beijing, China, vol. 2, pp. 792-795, May 1996.

[12] Y. Zhao, G. Ren, X.Wang, Z.Wu, and X.Gu, "Automatic digital modulation recognition using artificial neural networks", *in Proceedings of the International Conference on Neural Networks and Signal Processing (ICNNSP'03)*, vol. 33, pp. 257-260, 2003.

[13] H. Mustafa and M. Dorolovacki, "Digital modulation recognition using SVM classifier", *in Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, vol. 2, pp. 2238-2242, November 2004.

[14] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and KNN", *IEEE Transactions on wireless Communications*, vol. 11, pp. 2742-2750, 2012.

[15] Z. Zhu and A. K. Nandi, "Blind Digital Modulation Classification using Minimum Distance Centroid Estimator and Non-parametric Likelihood Function", *IEEE Transactions on Wireless Communications*, vol. 13, pp. 4483-4494, 2014.

[16] B. Dulek, O. Ozdemir, P. K. Varshney ,and S. Wei, "Modulation Discovery Over Arbitrary Additive Noise Channels Based on the Richardson-Lucy Algorithm", *IEEE Signal Processing Letters*, vol. 21, pp. 756-760, 2014.

[17] S. Kharbech, I. Dayoub, M. Zwingelstein-Colin, E. Simon, and K. Hassan, "Blind Digital Modulation Identification for Time-Selective MIMO Channels", *IEEE Wireless Communications Letters*, vol. 3, pp. 373-376, 2014.

[18] D. F. Specht, "Probabilistic neural networks", *Neural Networks*, vol. 3, pp. 109-118, 1990.

[19] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and S. Zaidi, "The bees algorithm, a novel tool for complex optimization problems", presented at the 2nd int virtual Conference on intelligent Production Machines and Systems (IPROMS 2006), 2006.

[20] J. G. Proakis, *Digital Communications*. New York: McGraw-Hill, 2001.

Saeed Hakimi received the B.Sc. degree from Islamic Azad University, Tehran, Iran, in 2008 and his M.Sc. degree from Babol University of Technology, Babol, Iran, in 2011; both in Communication Engineering. He is currently pursuing the Ph.D degree in the Department of Electrical Engineering at Ferdowsi university of Mashhad, Mashhad, Iran. His current research interests include Distributed Detection & Estimation, Statistical Signal Processing, Resource Allocation, and Information Theoretic Signal Processing.

Ataollah Ebrahimzadeh received his Ph.D. degree in electrical (telecommunication) engineering. Now he is an associate professor in the Faculty of Electrical and Computer Engineering at Babol University of Technology. His research interests are: Signal Processing, Pattern Recognition, Artificial Intelligence, and Modern Control. He is the reviewer of a large number of international journals and conferences.

[21] K. Z. Mao, K. C. Tan, and W. Ser, "Probabilistic neural-network structure determination for pattern classification", *IEEE Transactions on Neural Networks*, vol. 11, pp. 1009-1016, 2000.

[22] S. Chen, C .F. N. Cowan, and P. M. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Transactions on Neural Networks*, vol. 2, pp. 302-309, 1991.

[23] Y. Zheng Rong, M. Zwolinski, C. D. Chalk, and A. C. Williams, "Applying a robust heteroscedastic probabilistic neural network to analog fault detection and classification", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, pp. 142-151, 2000.

[24] D. T. Pham, A. J. Soroka, A. Ghanbarzadeh, E. Koc, S. Otri, and M. Packianather, "Optimising Neural Networks for Identification of Wood Defects Using the Bees Algorithm", *in Proceedings of the IEEE International Conference on Industrial Informatics*, vol. 1, pp. 1346-1351, 2006.

[25] Y. Wang, L. Li, J. Ni, and S. Huang, "Feature selection using tabu search with long-term memories and probabilistic neural networks", *Pattern Recognition Letter*, vol. 30, pp. 661-670, 2009.

[26] D. Karabog and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm", *Applied Soft Computing*, vol. 8, pp. 687-697, 2008.

[27] K. S. Tang, K. F. Man, S. Kwong, and Q. He, "Genetic algorithms and their applications", *IEEE Signal Processing Magazine*, vol. 13, pp. 22-37, 1996.

[28] J. Kennedy and R. Eberhart, "Particle swarm optimization", *in proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[29] F. C. B. F. Muller, C. Cardoso, and A. Klautau, "A Front End for Discriminative Learning in Automatic Modulation Classification", *IEEE Communications Letters*, vol. 15, pp. 443-445, 2011.

[30] K. Hassan, I. Dayoub, W. Hamouda, and M. Berbineau, "Automatic modulation recognition using wavelet transform and neural networks in wireless systems", *EURASIP Journal on Advances in Signal Processing*, vol. 2010, pp. 1-13, 2010.

IJICTR

This Page intentionally left blank.