Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

ELSEVIER

Signal Processing 84 (2004) 351–365

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Automatic digital modulation recognition using artificial neural network and genetic algorithm

M.L.D. Wong, A.K. Nandi*

Signal Processing and Communications Group, Department of Electrical Engineering and Electronics, University of Liverpool, Brownlow Hill, Liverpool L69 3GJ, UK

Received 9 August 2001; received in revised form 5 June 2003

Abstract

Automatic recognition of digital modulation signals has seen increasing demand nowadays. The use of artificial neural networks for this purpose has been popular since the late 1990s. Here, we include a variety of modulation types for recognition, e.g. QAM16, V29, V32, QAM64 through the addition of a newly proposed statistical feature set. Two training algorithms for multi-layer perceptron (MLP) recogniser, namely Backpropagation with Momentum and Adaptive Learning Rate is investigated, while resilient backpropagation (RPROP) is proposed for this problem, are employed in this work. In particular, the RPROP algorithm is applied for the first time in this area. In conjunction with these algorithms, we use a separate data set as validation set during training cycle to improve generalisation. Genetic algorithm (GA) based feature selection is used to select the best feature subset from the combined statistical and spectral feature set. RPROP MLP recogniser achieves about 99% recognition performance on most SNR values with only six features selected using GA.

Keywords: Modulation recognition; Artificial neural networks; Genetic algorithm; Feature subset selection

1. Introduction

Automatic modulation recognition (AMR) has its roots in military communication intelligence applications such as counter channel jamming, spectrum surveillance, threat evaluation, interference identification, etc. Whilst most methods proposed initially were designed for analogue modulations, the recent contributions in the subject focus more on digital communication [6,17,23,24]. Primarily, this is due to the increasing usage of digital modulations in many novel applications, such as mobile telephony, personal dial-up network, indoor wireless network, etc. With the rising developments in software-defined radio (SDR) systems, automatic digital modulation recognition (ADMR) has gained more attention than ever. Such units can act as front-end to SDR systems before demodulation takes place, thus a single SDR system can robustly handle multiple modulations.

In the early days, modulation recognition relied heavily on human operator's interpretation of measured parameters to classify signals. Signal properties such as IF waveform, signal spectrum, instantaneous amplitudes and instantaneous phase are often used in conventional methods. A latter form of recogniser consists of demodulator banks, each

* Corresponding author. Tel.: +44-151-794-4525; fax: +44-151-794-4540.
E-mail address: a.nandi@liv.ac.uk (A.K. Nandi).

0165-1684/$ - see front matter
doi:10.1016/j.sigpro.2003.10.019

Nomenclature

$s(t)$	modulated signal	σ_{aa}	SD of the AV of the centred normalised $a(i)$
$A_m(t)$	message amplitude	σ_{af}	SD of the AV of the centred normalised $f(i)$
$f_m(t)$	message frequency	C	number of non-linear components
$\phi_m(t)$	message phase	$\mathbf{X}_i$	signal vector
f_c	carrier frequency	C	cumulant
A_{mc}, A_{ms}	QAM amplitude modulators	$y(t)$	received signal
g_T	pulse shaping function	$\tilde{y}(t)$	Hilbert transform of $y(t)$
f_s	sampling frequency	H_y	complex envelope of $y(t)$
R_b	symbol baud rate	R	real part of H_y
N_s	number of samples	I	imaginary part of H_y
$a(i)$	instantaneous amplitude	$y_k(\mathrm{n})$	output of neuron k
$f(i)$	instantaneous frequency	w_{ij}	weight value for neuron i from neuron j
$\phi(i)$	instantaneous phase	u_i	weighted sum of the inputs of neuron i
$\gamma_{\max}$	maximum value of power spectral density of normalised $a(i)$	E	error function
σ_{ap}	standard deviation (SD) of the absolute value (AV) of the centred non-linear components of $\phi(i)$	ε	learning rate parameter
σ_{dp}	SD of the direct value of the centred non-linear components of $\phi(i)$	μ	momentum parameter
		$\eta^{-,+}$	update parameter
		Δ	next update value of the weight

designed for a particular modulation type. This is considered as semi-automatic since a human operator is still required to 'listen' to the output, but it is impractical for digital communications.

Since mid-1980s, new classes of modulation recognisers which automatically determine incoming modulation type have been proposed. Generally, these methods fall into two main categories, decision theoretic and statistical pattern recognition. Decision theoretic approaches use probabilistic and hypothesis testing arguments to formulate the recognition problem. The major drawback of this approach are the difficulties of forming the right hypothesis as well as careful analyses that are required to set the correct threshold values. Examples of decision theoretic approaches include Azzouz and Nandi [2] who proposed a global procedure for analogue and digital modulation signals, Swami and Saddler [13] who examines the suitability of cumulants as features for digitally modulated signal, and Dohono and Huo [6] who proposed a new method using Hellinger representation.

Pattern recognition approaches, however, do not need such careful treatment, although choosing the right feature set is still an issue. This classification method can be further divided into two: the feature extraction subsystem and the recognition subsystem. The feature extraction subsystem is responsible for extracting prominent characteristics which are called features from the raw data. Examples of features used are higher-order cumulants [23], signal spectral [20], constellation shape [17], power moments [8], etc.

The second subsystem, the pattern recogniser, is responsible for classifying the incoming signal based on the features extracted. It can be implemented in many ways, e.g. K-nearest neighbourhood classifier (KNN), probabilistic neural network (PNN), support vector machine (SVM), etc. [15,16,18,22] chose multi-layer perceptron (MLP) as their modulation recogniser system.

Louis and Sehier [22] proposed a hierarchical neural network which uses backpropagation training. They also gave a performance analysis on backpropagation with other algorithms such as cascade correlation, binary decision tree and KNN. Lu et al. [16] proposed a novel MLP-based modulation neural network recogniser using instantaneous frequency and bandwidth features of signals. In [15], Lu et al. enhanced their techniques through the usage of cyclic spectrum features. Nandi and Azzouz [18] proposed MLP neural networks with spectral feature set for analogue, digital and combined modulation recognition. Their algorithms had inspired the foundation of a couple of commercial product or prototype; examples of hardware implementation had been reported in a variety of applications, e.g. 4G software radio wireless networks [14], spectrum monitoring hardware [4,5], etc.

The work presented in this paper is a continuation of the works of Azzouz and Nandi [2,3,18,20] with focus on digital modulations and artificial neural network (ANN) based recognisers. Here we present the use of two updated ANN training algorithms, which improve the performance and epoch times by a large margin. On the other hand, additional modulation and a new feature set based on higher-order statistics (HOS) of the signal are studied. The performance of spectral, statistical and combined feature sets have also been reviewed. Furthermore, we investigate the effects of features subset selection on the combined feature set using genetic algorithm (GA), which clearly demonstrates the reduction in feature set with enhanced performance.

This paper is organised as follows: Section 2 states the problem of automatic recognition of digital modulated signals. Section 3 briefly discusses the extraction of different feature sets; MLP-based recognisers and its training algorithms are briefly discussed in Section 4. Section 5 presents two methods of feature subsets selection using GA. Some simulation results are shown in Section 6. Finally, conclusions and future plans are presented in Section 7.

2. Problem statement

In digital communication, a modulated signal can be generally represented as

$$s(t) = A_m g_{\mathrm{T}}(t)\cos(2\pi f_m(t) + \phi_m(t)), \tag{1}$$

where A_m, f_m and ϕ_m are the message amplitude, message frequency and message phase, respectively, in accordance with appropriate modulation techniques and g_{T} is the pulse shaping function. We consider the four main types of digital modulation techniques, i.e. amplitude shift key (ASK), phase shift key (PSK), frequency shift key (FSK) and quadrature amplitude modulation (QAM).

QAM (e.g. QAM16, V29, V32, QAM64, etc.), being a newer type among the four, takes a slightly different form

$$s_m(t) = A_{m\mathrm{c}} g_{\mathrm{T}}(t)\cos 2\pi f_{\mathrm{c}} t + A_{m\mathrm{s}} g_{\mathrm{T}}(t)\sin 2\pi f_{\mathrm{c}} t. \tag{2}$$

As seen in Eq. (2), QAM employs two quadrature carriers, instead of one. Each carrier is independently modulated by $A_{m\mathrm{c}}$ and $A_{m\mathrm{s}}$, where m takes the value from 1 to M, with M corresponding to the number of signal levels. For example a QAM16 signal is by nature of combination of two ASK4 signals with different quadrature carriers. QAM can also be written as

$$s_{m1m2}(t) = A_{m1} g_{\mathrm{T}}(t)\cos(2\pi f_{\mathrm{c}} t + \phi_{m2}), \tag{3}$$

where QAM can be seen as a combined digital amplitude and digital phase modulation.

The function of ADMR is to act as an intermediate stage between channel equalisation and demodulation. It is not meant to recover the embedded information in the sampled data. Ten digital modulation types are considered in this paper:

- ASK2
- ASK4
- BPSK
- QPSK
- FSK2
- FSK4
- QAM16
- V29
- V32
- QAM64

3. Feature extraction

All signals are digitally simulated according to Eqs. (1) and (3) in MATLAB environment. Random integer up to M-level ($M = 2, 4, 16, 29, 32, 64$) are first generated by a uniform random number generator (URNG). These are then resampled at baud rate for pulse shaping before passing through respective modulators. Parameters used for modulation are shown in Table 1.

The simulated signals were also band-limited and Gaussian noise were added according the predefined signal-to-noise ratio (SNR), i.e. −5, 0, 5, 10, and 20 dB.

Table 1
Modulation parameters

Parameters	Values
Sampling frequency, f_s	307.2 kHz
Carrier frequency, f_c	17 kHz
Baud rate, R_b	9600 baud
No. samples, N_s	4096 samples

Typical pattern recognition systems often reduce the size of a raw data set by extracting some distinct attributes called features. These features, which can be represented as d-dimensional vectors, define a particular pattern.

Azzouz and Nandi [3, Chapter 3] proposed a spectral-based feature set for digital modulations. These features were demonstrated to be suitable for signals which contain hidden information in either instantaneous amplitude (OOK, ASK4), instantaneous frequency (M-ary FSKs) or instantaneous phase (BPSK, PSK4).

However, modulation schemes such as QAM contain information in both amplitude and phase. In this work, two feature sets—a spectral feature set (see Section 3.1) and a new feature set based on higher-order cumulants of the signal (see Section 3.2) are used.

3.1. Spectral feature set

The main motivation for this spectral feature set is that, the information content for digital modulations is hidden either in the signal instantaneous amplitude, instantaneous phase or instantaneous frequency, while, for the previously proposed modulation types [3, Chapter 3], information is only hidden in a single domain for a particular modulation type. The five features proposed are described as below:

- Maximum value of the power spectal density of the normalised-centred instantaneous amplitude

$$\gamma_{\max} = \max |DFT(a_{cn}(i))|^2/N_s, \tag{4}$$

where N_s is the number of samples, $a_{cn}(i)=a_n(i)-1$ and $a_n(i) = a(i)/m_a$, $a(i)$ is the ith instantaneous amplitude and m_a is the sample mean value.

- Standard deviation of the absolute value of the centred non-linear components of the instantaneous phase

$$\sigma_{ap}=\sqrt{\frac{1}{C}\left(\sum_{a_n(i)>a_t}\phi_{NL}^2(i)\right)-\left(\frac{1}{C}\sum_{a_n(i)>a_t}|\phi_{NL}(i)|\right)^2}, \tag{5}$$

where C is the number of samples in $\{\phi_{NL}(i)\}$ for which $a_n(i) > a_t$ and a_t is the threshold value for $a(i)$ below which the estimation of the instantaneous phase is very noise sensitive.

- Standard deviation of the direct value of the centred non-linear component of the instantaneous phase

$$\sigma_{dp}=\sqrt{\frac{1}{C}\left(\sum_{a_n(i)>a_t}\phi_{NL}^2(i)\right)-\left(\frac{1}{C}\sum_{a_n(i)>a_t}\phi_{NL}(i)\right)^2}. \tag{6}$$

- Standard deviation of the absolute value of the normalised-centred instantaneous amplitude

$$\sigma_{aa}=\sqrt{\frac{1}{N_s}\left(\sum_{i=1}^{N_s}a_{cn}^2(i)\right)-\left(\frac{1}{N_s}\sum_{i=1}^{N_s}|a_{cn}(i)|\right)^2}. \tag{7}$$

- Standard deviation of the absolute value of the normalised-centred instantaneous frequency

$$\sigma_{af}=\sqrt{\frac{1}{C}\left(\sum_{a_n(i)>a_t}f_N^2(i)\right)-\left(\frac{1}{C}\sum_{a_n(i)>a_t}|f_N(i)|\right)^2}. \tag{8}$$

Note that the availability of instantaneous amplitude, phase and frequency depends on the availability of carrier frequency f_c. In this paper, we assume that f_c is known a priori; in practical applications it would need to be estimated. Besides the frequency estimation (see Appendix B in [5]), other concerns such as channel information are assumed to have been adequately addressed either with a priori knowledge or through blind estimation algorithms. The above features were used in previous works of Nandi and Azzouz.

3.2. Statistical feature set

Present modulation schemes, e.g. QAM16, V29, etc., often contain information in both amplitude and phase spectra. In communications, these modulation are regarded as complex signal and often be illustrated as in-phase and quadrature (I–Q) components. HOS provide a good way to obtain features to illustrate the two-dimensional probability density function (PDF), i.e. the constellation diagram. Normalised HOS provide invariant properties that are of interest, e.g. amplitude invariant. Besides, HOS are not affected by additive white Gaussian noise (AWGN) as its higher cumulants (order > 2) are always zero.

In [23], Swami and Saddler reported a detailed account on ADMR using normalised complex HOS. Here, the authors proposed a different scheme by using the cumulants of the real and imaginary parts of the analytic signal.

3.2.1. Cumulants

Let $\mathbf{X}_i$ be a signal vector, $\{x_i^1, x_i^2, \ldots, x_i^N\}$, and $\langle\,\rangle$ denote the statistical expectation. The second-, third- and fourth-order cumulants at zero lag is then

$$\mathrm{C}_{\mathbf{X}_1,\mathbf{X}_2} = \langle \mathbf{X}_1, \mathbf{X}_2 \rangle = \frac{1}{N}\sum_{n=1}^{N} x_1^n x_2^n, \tag{9}$$

$$\mathrm{C}_{\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3} = \langle \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \rangle = \frac{1}{N}\sum_{n=1}^{N} x_1^n x_2^n x_3^n, \tag{10}$$

$$\begin{aligned}\mathrm{C}_{\mathbf{X}_1,\mathbf{X}_2,\mathbf{X}_3,\mathbf{X}_4} &= \langle \mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \mathbf{X}_4 \rangle - \langle \mathbf{X}_1, \mathbf{X}_2 \rangle\langle \mathbf{X}_3, \mathbf{X}_4 \rangle \\ &\quad - \langle \mathbf{X}_1, \mathbf{X}_3 \rangle\langle \mathbf{X}_2, \mathbf{X}_4 \rangle - \langle \mathbf{X}_1, \mathbf{X}_4 \rangle\langle \mathbf{X}_2, \mathbf{X}_4 \rangle \\ &= \frac{1}{N}\sum_{n=1}^{N} x_1^n x_2^n x_3^n x_4^n - \mathrm{C}_{\mathbf{X}_1,\mathbf{X}_2}\mathrm{C}_{\mathbf{X}_3,\mathbf{X}_4} \\ &\quad - \mathrm{C}_{\mathbf{X}_1,\mathbf{X}_3}\mathrm{C}_{\mathbf{X}_2,\mathbf{X}_4} - \mathrm{C}_{\mathbf{X}_1,\mathbf{X}_4}\mathrm{C}_{\mathbf{X}_2,\mathbf{X}_3}.\end{aligned} \tag{11}$$

3.2.2. Feature set

Let H_y be the complex envelope of the sampled signal $y(t)$ which is defined by

$$H_y = [y(t) + \mathrm{j}\hat{y}(t)]\exp(-\mathrm{j}2\pi f_{\mathrm{c}} t), \tag{12}$$

where $\tilde{y}(t)$ is the Hilbert transform of $y(t)$ and f_{c} is the carrier frequency.

We then define R to be the real part of H_y, $\Re(H_y)$, and I to be the imaginary part of $H_y, \Im(H_y)$. Thus, we propose the following feature set:

$$\begin{aligned}&\mathrm{C}_{R,R}, \mathrm{C}_{R,I}, \mathrm{C}_{I,I}, \\ &\mathrm{C}_{R,R,R}, \mathrm{C}_{R,R,I}, \mathrm{C}_{R,I,I}, \mathrm{C}_{I,I,I}, \\ &\mathrm{C}_{R,R,R,R}, \mathrm{C}_{R,R,R,I}, \mathrm{C}_{R,R,I,I}, \mathrm{C}_{R,I,I,I}, \mathrm{C}_{I,I,I,I},\end{aligned} \tag{13}$$

which are the second-, third- and fourth-order cumulants and cross cumulants of the real and imaginary parts of the signal. These features are then combined with the original five features to give the final combined feature set.

Each modulation scheme has a unique constellation diagram in its I–Q plane (see Fig. 1). Examples of some HOS features to distinguish among different constellations are shown in Figs. 2 and 3.

In this paper, it is assumed that the phase recovery process is perfect. In practice, the clock recovery process often introduces phase offset or non-coherency in the recovered signal. Prior to the extraction of these features, the phase error must be corrected. This is a part of the authors' on going work and will be presented in another forthcoming paper. For dense modulation ($M > 64$), care is to be taken so that the sample size is large enough to include every symbol in the constellation map. Otherwise, the estimates of the higher-order cumulants will not be accurate.

3.3. Data preprocessing and normalisation

Although feature extraction and selection are essentially a form of data preprocessing, we regard any work done prior to feature extraction as data preprocessing. These include data sampling, pulse shaping, estimation of carrier frequencies, f_{c}, and recovery of complex envelope through Hilbert transform.

Normalisation is carried out before feeding the extracted features into the MLP recogniser for training, validation and testing. The features are normalised to ensure that they are zero mean and unit variance. Fig. 4 shows the complete process of the proposed digital modulation recogniser.

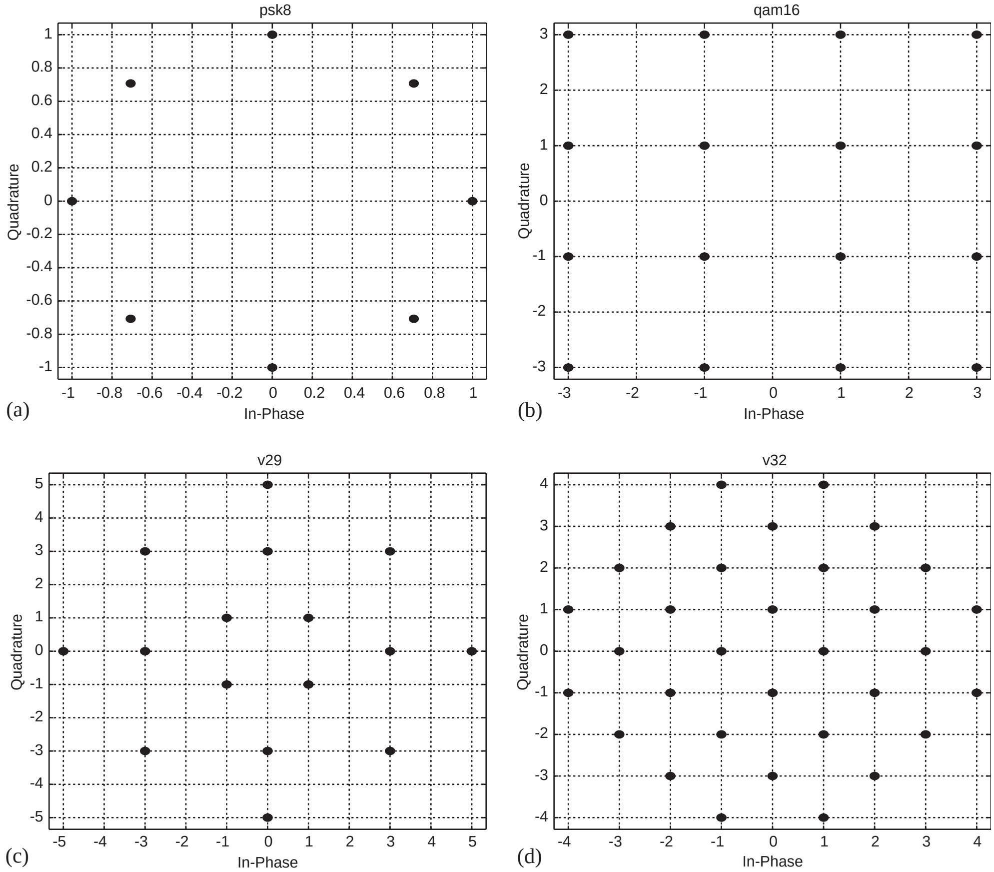

Fig. 1. Various constellation signatures: (a) PSK8; (b) QAM16; (c) V29; (d) V32.

4. MLP classifier

ANN gained their popularity in recent decades due to their abilities to learn complex input–output mapping of non-linear relationships, self-adaptivity to environment changes and fault tolerance. A popular family of ANN is the feed-forward networks which include the MLP and the radial basis function (RBF) networks. We have chosen MLP as the classifier in this work; however, there is nothing to prevent one from choosing other types of neural networks. Ibnkahla [9] surveyed a wide variety of neural networks in digital communications.

MLP is chosen due to its simplicity and efficient hardware implementation. In many ways, MLP can be likened to a posteriori probability estimation and non-linear discriminant analysis in statistical pattern recognition. In fact, MLP hides the details of background statistics and mathematics from the users, and at the same time provides an easily understandable mathematical model of a biological concept for the solution.

MLP is a feed-forward structure of interconnection of individual non-linear parallel computing units called neurons. Inputs are propagated through the network layer by layer and MLP gives an non-linear mapping of the inputs at the output layers.

We can write MLP mathematically as

$$y_k(n) = \phi\left(\sum_{i=1}^{q} w_{ki}\,\phi\left(\sum_{j=0}^{p} w_{ij}x_j(n)\right)\right), \qquad (14)$$

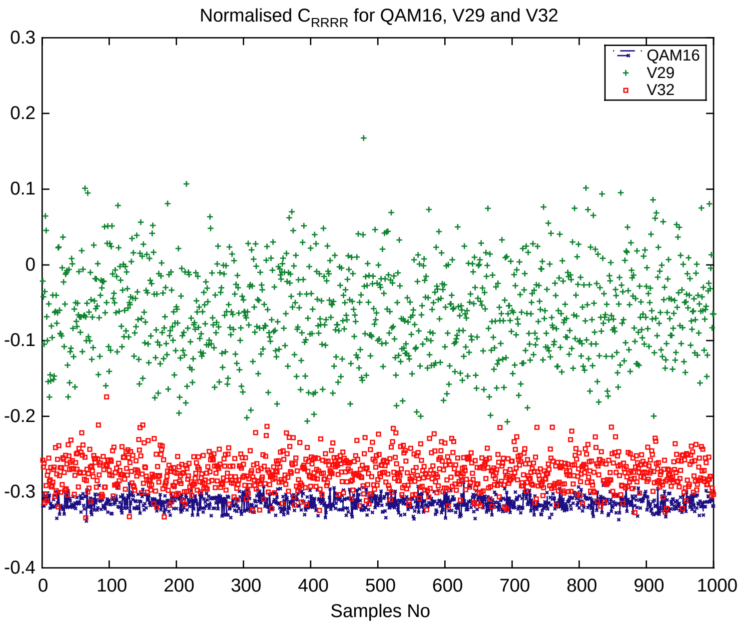

Fig. 2. Fourth-order cumulant of real part of the analytical signals of three different modulation types (SNR at 20 dB).

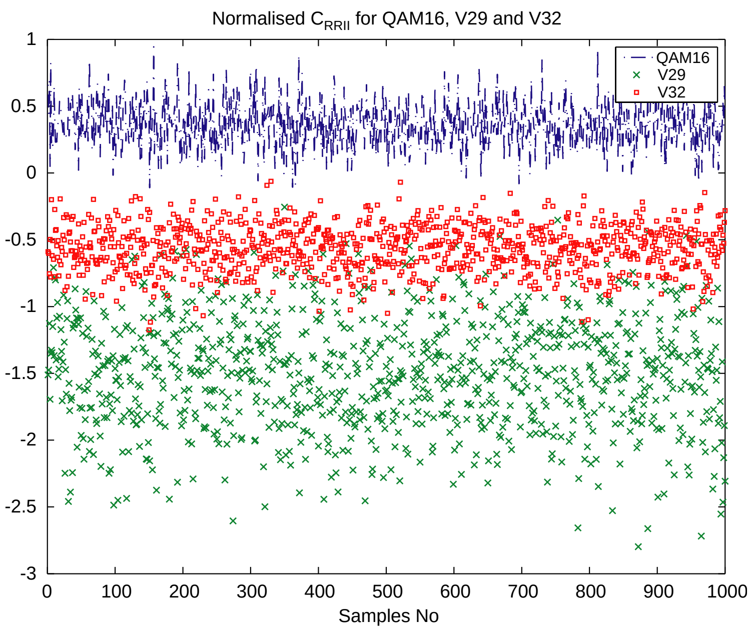

Fig. 3. Fourth-order cross cumulant of the analytical signals of three different modulation types (SNR at 20 dB).

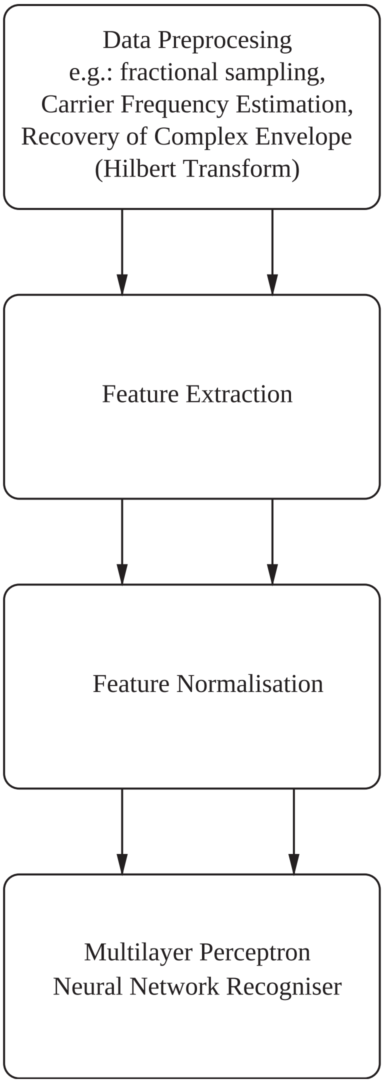

Fig. 4. Proposed digital modulation recogniser.

where n is the sample number, subscript k now denotes the output nodes, subscripts i and j denote hidden and inputs nodes, respectively. Note that the activation functions, ϕ, are allowed to vary for different layers of neurons.

The recognition basically consists of two phases—training and testing. A paired training input and target output are presented at each training process, and weights are calculated according to the chosen learning algorithm. For a batch training algorithm, weights are updated once every training epoch, meaning a full run of training sample, and for an adaptive training, weights are updated every training sample. We shall discuss learning algorithms more in depth in the next section.

4.1. Learning algorithms

MLP is relatively mature branch of ANN and there are a number of efficient training algorithms. Azzouz and Nandi [19] adopted the standard backpropagation algorithm (BP). They have also considered BP with momentum and adaptive learning rate to speed up the training time required.

BP algorithms implement generalised chained rules repetitively to calculate the changes of each weight with respect to the error function, E. Examples of some common choices are the sum squared error (SSE) and mean squared error (MSE):

$$\frac{\delta E}{\delta w_{ij}} = \frac{\delta E}{\delta y_i}\frac{\delta y_i}{\delta u_i}\frac{\delta u_i}{\delta w_{ij}}, \tag{15}$$

where w_{ij} represents the weight value from neuron j to neuron i, y_i is the output and u_i is the weighted sum of the inputs of neuron i.

The weight values are then updated by a simple gradient descent algorithm

$$w_{ij}(t+1) = w_{ij}(t) - \varepsilon\frac{\delta E}{\delta w_{ij}}(t). \tag{16}$$

The learning rate parameter, ε, is analogous to the step size for a least mean square (LMS) adaptive filter, where a higher learning rate means a faster convergence, but with risk of oscillation. On the other hand, a value too small will take too much time to achieve convergence. An adaptive learning rate variant of BP takes account of this problem by updating the learning rate adaptively. By doing so, one also avoids trapping in a local minima.

A BP algorithm with momentum adds an extra momentum parameter, μ, to the weight changes:

$$\Delta w_{ij}(t+1) = -\varepsilon\frac{\delta E}{\delta w_{ij}}(t) + \mu\frac{\delta E}{\delta w_{ij}}(t-1). \tag{17}$$

This takes account of the previous weight changes and leads to a more stable algorithm and accelerates convergence in shallow areas of the cost function.

In recent years, new algorithms have been proposed for network training. However, some algorithms require much computing power to achieve good training, especially when dealing with a large training set. Although these algorithms require very small numbers of training epochs, the actual training time for a epoch is much longer compared with BP algorithms. An example would be the Levenberg–Marquardt Algorithm (LM).

In this paper, we consider the BP algorithms and the resilient backpropagation algorithm (RPROP) proposed by Riedmiller and Braun [21] in 1993. Basically, unlike BPs, RPROP only considered the sign of derivatives as the indication for the direction of the weight update. In doing so, the size of the partial derivative does not influence the weight step.

The following equation shows the adaptation of the update values of Δ_{ij} for the RPROP algorithm. For initialisation, all Δ_{ij} are set to small positive values:

$$\Delta_{ij}(t) = \begin{cases} \eta^{+} * \Delta_{ij}(t-1) \\ \quad \text{if } \dfrac{\delta E}{\delta w_{ij}}(t-1)\dfrac{\delta E}{\delta w_{ij}}(t) > 0, \\ \eta^{-} * \Delta_{ij}(t-1) \\ \quad \text{if } \dfrac{\delta E}{\delta w_{ij}}(t-1)\dfrac{\delta E}{\delta w_{ij}}(t) < 0, \\ \eta^{0} * \Delta_{ij}(t-1) \quad \text{otherwise,} \end{cases} \tag{18}$$

where $\eta^0 = 1$, $0 < \eta^- < 1 < \eta^+$ and $\eta^{-,0,+}$ are known as the update factors. Whenever the derivative of the corresponding weight changes its sign, it implies that the previous update value is too large and it has skipped a minimum. Therefore, the update value is then reduced (η^-) as shown above. However, if the derivative retains its sign, the update value is increased (η^+). This will help to accelerate convergence in shallow areas. To avoid over-acceleration, in the epoch following the application of η^+, the new update value is neither increased nor decreased (η^0) from the previous one. Note that values of Δ_{ij} remain non-negative in every epoch.

This update value adaptation process is then followed by the actual weight update process, which is governed by the following equations:

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t) & \text{if } \dfrac{\delta E}{\delta w_{ij}}(t) > 0, \\ +\Delta_{ij}(t) & \text{if } \dfrac{\delta E}{\delta w_{ij}}(t) < 0, \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t). \tag{20}$$

4.2. Generalisation and early stopping

Early stopping is often used to avoid over-training of a recogniser and to achieve better generalisation. When training a recogniser, the error measured generally decreases as the number of training epochs increases. However, if an error value is measured against an independent data set (the validation set) after each training epoch, the error will first decreases until a point where the network starts to become over-trained, the error then starts to increase with the epoch value.

We can, therefore, stop the training procedure at the smallest error value of the validation set since it gives the optimal generalisation of the recogniser.

5. Feature selection

Pattern recognition applications often face the problem of the curse of dimensionality. The easiest way to reduce the input dimension is to select some inputs and discard the rest. This method is called feature selection.

The advantage of feature selection is that one can use the least possible number of features without compromising the performance of the MLP recogniser. The effect can be seen best when there are highly correlated inputs or more than one input share the same information content. Situations like this often occur when addressing pattern recognition problems.

There exists two main paradigms of feature selection, i.e. the filter-type approach (e.g. [1,13]) and the wrapper type approach (e.g. [11]). For the filter-type approach the feature set is evaluated without the aid of the application, in this case, the MLP recogniser. In

Table 2
An overview of generalised GA

(1)	Initialise a population P_0 of N individuals
(2)	Set generation counter, $i = 1$
(3)	Create a intermediate population, P_i', through selection function
(4)	Create a current population, P_i through reproduction functions
(5)	Evaluate current population
(6)	Increment the generation counter, $i = i + 1$
(7)	Repeat step 3 until termination condition reached
(8)	Output the best solution found

other word, the features were selected based on some predefined functions of the members of the features set. An example of these functions is the correlation among the features. The wrapper type approach, however, uses the performance of recogniser to evaluate the selected feature subset. This has the advantage of guaranteed performance but often takes a longer time.

There are many methods for feature selection, but they generally consist of two main parts—a selection phase and evaluation phase. In pattern recognition, the selection criterion is normally the minimisation of recognition errors. This leaves us with choosing a suitable selection procedure. In this paper, we shall focus on a stochastic search method using GA. Similar wrapper type feature selection method with GA were also proposed in [10,25]. Conventional brute force searching method like forward selection, backward selection, etc. are also available, but are not discussed in the context of this paper. A comparison among these methods can be found in [12].

5.1. Genetic algorithm

GA is a stochastic optimisation algorithm which adopts Darwin's theory of survival of the fittest. The algorithm runs through an iteration of individual selection, reproduction and evaluation as illustrated in Table 2. In recent years, GA has found many application in engineering sector, such as adaptive filter design, evolutionary control systems, feature extraction, evolutionary neural networks, etc.

Two important issues in GA are the genetic coding used to define the problem and the evaluation function. Without these, GA is nothing but a meaningless repetition of procedures. The simple GA proposed by Goldberg [7] in his book uses binary coding, but other methods such as real coding sometimes are more meaningful to some problems. This is discussed in further details in Section 5.2.

The evaluation function estimates how good an individual is in surviving the current environment. In function minimisation problems, the individual that gives the smallest output will be given the highest score. In this case, the evaluation function can be the reciprocal of output plus a constant. In the modulation recognition problem, we define the evaluation as the overall performance of the recogniser, which is an average performance of training, validation and testing. Therefore, the GA implemented here is designed to seek the maximum score returned by each individual.

5.2. String representation

Each individual solution in GA is represented by a genome string. This string contains specific parameters to solve the problem. In our application, two different methods of coding are investigated, i.e. the binary genome coding and the list genome coding.

In the binary genome coding, each string has a length N, where N is the total number of input features available. A binary '1' denotes the presence of the feature at the corresponding index number. Similarly, a binary '0' denotes an absence. The advantage of this coding is that it searches through the feature subspace dynamically without user defined number of subset features. No constraint is needed with this coding method.

The second coding used is the real numbered list genome string. Each genome in this category is of length M, where M is the desired number of features in a feature subset. To initialise, the GA chooses randomly M numbers from a list of integers ranging from 1 to N. However, we do not desire any repetition of the integers as this means that the same feature is selected more than once. Therefore a constraint, $1 < f_i < N$ is applied, where f_i denotes ith input feature. In practice, we randomise the list sequence and choose the first M features in the list.

An extra parameter can be appended to the genome for choosing the number of hidden units in the MLP recogniser. However, from previous simulations, we learned that the optimum number of hidden units is around ten neurons using RPROP training algorithm. Since this is practical and reasonable for a hardware implementation, we shall not complicate our problem of feature selection with an extra parameter.

5.3. Basic genetic operators

Basic genetic operators are used for reproduction and selection. The following gives a brief description of each operator:

Crossover: Crossover occurs with a crossover probability of P_c. A point is chosen for two strings where their genetic informations are exchanged. There are also variation of two- or multi-point crossover. For our purpose, we shall use one-point crossover, and typical value of P_c of 0.75.

Mutation: Mutation is used to avoid local convergence of the GA. In binary coding, it just means the particular bit chosen for mutation is inverted to its complement. In list coding, the chosen index is replaced with a new index without breaking the constraint. Mutation occurs with typical mutation probability of 0.05. This probability value is kept at such a low value to prevent unnecessary oscillation.

Selection: There are several ways to select a new intermediate population. Based on the performance of individual strings, roulette wheel selection assigns a probability to each string according to their performance. Therefore, poor genome strings will have a slight chance of survival. Unlike roulette wheel, selection by rank just orders the individuals according to their performance and select copies of best individuals for reproduction.

Other genetic operators like elitism, niche, diploidy are often classified as advanced genetic operators. For the purpose of investigation, we shall apply only elitism. Elitism comes in various forms. In our application, we require that the best two strings are always to be included in the new population. This gives a chance to reevaluate their capabilities and improves GA convergence.

6. Simulation results

6.1. MLP and BP with momentum and adaptive learning rate

In this section, the performance of the proposed recogniser is investigated with ten different modulation types, i.e. ASK2, ASK4, BPSK, QPSK, FSK2, FSK4, QAM16, V29, V32 and QAM64. Random numbers up to M-level ($M = 2$, 4, 8, 16, 32, etc.) are first generated using a URNG. These are then re-sampled at baud rate for pulse shaping. Each modulation type has 3000 realisations of 4096 samples each which makes a total of 30,000 realisations. These are then equally divided into three data sets for training, validation and testing. The MLP recogniser is allowed to run up to 10,000 training epochs. However, training is normally stopped by the validation process long before this maximum epoch is reached.

A single hidden layer MLP feed-forward network was chosen as the recogniser. As described in the previous section, the number of neurons in the hidden layer needs to be determined manually at this stage. Generally, any number in the vicinity of 40 neurons seems to be adequate for reasonable classification. In a noiseless setup, the recogniser can show up to 100% accuracy as shown in Table 3.

The effect of noise on recogniser performance is also studied through training and testing with different SNRs. Table 4 shows the performance of the recogniser for various SNR values. Performance is generally good even with low SNR values. For example at 0 dB SNR, a performance success of 98% is recorded.

6.2. MLP and RPROP

Using a similar setup as in the previous section, the MLP recogniser is trained using the RPROP algorithm as described in Section 4.1. The task is again to recognise 10 different modulation types grouped by SNR values.

Table 5 presents a comparison of performance and number of epochs. As seen from the results, RPROP reduced the epoch training to about one-fifth of BP with momentum and adaptive learning rate.

Table 3
Training performance for BP MLP with 40 hidden layer neurons (SNR $= \infty$)

Simulated modulation type	Deduced modulation type									
	ASK2	ASK4	PSK2	PSK4	FSK2	FSK4	QAM16	V.29	V.32	QAM64
ASK2	1000	0	0	0	0	0	0	0	0	0
ASK4	0	1000	0	0	0	0	0	0	0	0
BPSK	0	0	1000	0	0	0	0	0	0	0
QPSK	0	0	0	1000	0	0	0	0	0	0
FSK2	0	0	0	0	999	1	0	0	0	0
FSK4	0	0	0	0	0	1000	0	0	0	0
QAM16	0	0	0	0	0	0	1000	0	0	0
V.29	0	0	0	0	0	0	0	1000	0	0
V.32	0	0	0	0	0	0	0	0	1000	0
QAM64	0	0	0	0	0	0	0	0	0	1000

All 17 features were used.

Table 4
Performance of BP MLP Recogniser with 40 hidden layer neurons at different SNR values

Performance (%)	SNR (dB)				
	−5	0	5	10	20
Training	90.38	98.11	99.32	99.86	100.00
Validation	89.18	98.00	99.35	99.82	100.00
Testing	89.35	97.91	99.21	99.90	99.96
Overall	89.64	98.01	99.33	99.86	99.99

All 17 features were used.

Table 5
Overall performance and number of epochs used for RPROP MLP with 10 hidden layer neurons vs. BP MLP with 10 hidden layer neurons

SNR (dB)	BP		RPROP	
	Overall perf. (%)	Elasped epochs	Overall perf. (%)	Elasped epoch
−5	73.62	441	94.79	490
0	87.02	421	99.07	129
5	99.26	2101	99.53	483
10	89.86	995	99.95	285
20	99.93	1593	99.97	257

All 17 features were presented at the input.

Computationally, the time difference between both algorithms is negligible.

Another point worth noting is that the RPROP needs less hidden neurons than the conventional BP. Table 6 shows results obtained using only 10 hidden layer neurons as opposed to 40 hidden layer neurons in the previous section. Performance are notably better for the lower SNR values.

RPROP also seems to perform better in lower SNR range compared to BP with momentum and adaptive learning rate. The simulations were carried out in MATLAB environment using neural network toolbox.

Table 6
Performance of RPROP MLP Recogniser with 10 hidden layer neurons at different SNR values

	SNR (dB)				
Performance (%)	−5	0	5	10	20
Training	95.42	99.32	99.58	99.95	99.97
Validation	94.5	98.92	99.45	99.97	99.98
Testing	94.45	98.98	99.55	99.94	99.96
Overall	94.79	99.07	99.52	99.95	99.97

All 17 features were presented.

6.3. Feature selection using GA

Experiments were again carried out in the MATLAB environment. By choosing the wrapper-type approach, the RPROP MLP recogniser (as seen in Section 6.2) was chosen as the evaluation function and the overall performance was returned as the fitness value of the genome string.

Convergence of a GA can be defined in several ways. In our application, the performance convergence is chosen. Therefore we define convergence as the ratio between present fitness function value and the average of fitness function values from the past five generations. The GA is allowed to run until it reaches convergence.

6.3.1. Binary coding

For binary coding, 20 binary genome strings were created randomly with uniform probability distribution during initialisation. This implies that each bit in the string is independent of other bits. One-point crossover and single bit mutation were used as reproduction operators. The GA was run until it reached convergence. Each feature subset selected was allowed to train for a maximum of 250 epochs unless it was not stopped by the validation process.

Table 7 shows the best genome strings selected in different SNR environment and their performance. These are then compared with the performance of the MLP recogniser trained using RPROP.

Generally, perfomance from reduced feature sets selected in different SNRs show improvement over the performance shown by the RPROP MLP recogniser. Moreover, feature subsets contain typically four or five features less than the original feature set.

6.3.2. Fixed length list coding

With list genome encoding method, one has to specify the length of the genome string, which represents the number of features in the reduced feature set. As seen from the previous section, without any compromise in performance, the feature set can typically be reduced down to 12 features. Therefore, in this section, experiments were carried out to investigate the possibility of even smaller data set and the compromise in performance that one might observe.

Table 8 shows the performance of the recogniser using only six features selected by the GA and the performance using the original full feature set. At −5 dB, the recogniser records a performance degradation of 2% only. For other SNRs, the difference is negligible.

6.4. Performance comparison

As mentioned in [17], direct comparison with other works is difficult in modulation recognition. This is mainly because of the fact that there is no single unified data set available. Different setup of modulation types will lead to different performance. Besides, there are many different kinds of benchmarking systems used for signal's quality. This causes difficulties for direct numerical comparison.

As for neural network-based modulation recognisers, Louis and Sehier reported a generalisation rate of 90% and 93% of accuracy of data sets with SNR of 15–25 dB. However, the performance for lower SNRs are reported to be less than 80% for a fully connected network, and about 90% for a hierarchical network.

Lu et al. [16] show an average accuracy around 90% with 4096 samples per realisation and SNR ranges

Table 7
Performance of binary string representation GA RPROP MLP with 10 hidden layer neurons with dynamic feature selection

SNR (dB)	Binary string	Features chosen	Perf. with GA (%)	RPROP Perf. (%)
−5	1101111111111100	15	94.37	94.79
0	1111010001110111	12	99.15	99.07
5	1111010001011110	11	99.97	99.53
10	1111010001110111	12	99.97	99.95
20	1101100000111011	11	99.99	99.97

Table 8
Performance of list string representation GA RPROP MLP with 10 hidden layer neurons

SNR (dB)	Genome string	Features chosen	Perf. with GA (%)	RPROP Perf. (%)
−5	1, 2, 7, 8, 9, 12	6	92.93	94.79
0	6, 7, 8, 9, 12, 15	6	99.02	99.07
5	7, 8, 13, 14, 15, 16	6	99.34	99.53
10	5, 6, 7, 12, 13, 15	6	99.83	99.95
20	6, 7, 12, 13, 15, 16	6	99.80	99.97

Only six features were selected.

between 5 and 25 dB. By increasing the number of samples, an increase of 5% in average performance is achieved. In [15], Lu et al. show through computer simulation the average recognition rate is 83%, and it reaches over 90% for SNR value of over 20 dB. However, if SNR is less than 10 dB, the performance drops to less than 70%.

The MLP recogniser proposed in this work shows a steady performance with different SNR values, e.g. over 90% for all SNR values. Through feature selection, the input dimension is reduced to less than half the original size without trading off the generalisation ability and accuracy. The proposed recogniser is also fast in terms of training time. If it were known that changes have occurred, for example, the network can easily be retrained.

7. Conclusion

AMR is an important issue in communication intelligence (COMINT) and electronic support measure (ESM) systems. Previous works on ANN recognisers have been encouraging but with a limited type of modulations; here additional digital modulations like QAM16, V29, V32 and QAM64 are included. We propose, in this paper, a combination of a spectral and a new statistical-based feature set with an MLP recogniser implementation. Also, the authors have compared two ANN training algorithms, BP with momentum and adaptive learning and RPROP, which, as far as the authors are aware, is the first application of the RPROP algorithm in ADMR. Simulations show satisfactory results, about 99% accuracy at SNR value of 0 dB with good generalisation. RPROP also demonstrated better and faster performance than BP.

GA was used to perform feature selection to reduce the input dimension and increase performance of the RPROP MLP recogniser. Significant improvements can be seen at lower SNR values using the selected feature subset. Using only six features in list coding method, the RPROP MLP managed to deliver a performance of 99% at 0 dB SNR and 93% at −5 dB SNR.

Acknowledgements

The authors would like to thank Overseas Research Studentship (ORS) Awards Committee UK and the

University of Liverpool for funding this work. They are also grateful for the comments from the anonymous reviewers for this paper.

References

[1] S. Abe, R. Thawonmas, Y. Kobayashi, Feature selection by analyzing class regions approximated by ellipsoids, IEEE Trans. Systems Man Cybernet. Part C Appl. Rev. 28 (2) (1998) 282–287.

[2] E.E. Azzouz, A.K. Nandi, Procedure for automatic recognition of analogue and digital modulations, IEE Proc. Commun. 143 (5) (1996) 259–266.

[3] E.E. Azzouz, A.K. Nandi, Automatic Modulation Recognition of Communication Signals, Kluwer Academic Publishers, Dordrecht, 1996.

[4] D. Boudreau, C. Dubuc, F. Patenaude, M. Dufour, J. Lodge, R. Inkol, A fast modulation recognition algorithm and its implementation in a spectrum monitoring application, in: MILCOM 2000 Proceedings. 21st Century Military Communications. Architectures and Technologies for Information Superiority, Vol. 2, IEEE Press, New York, 2000, pp. 732–736

[5] Communications Research Centre Canada, Spectrum Explorer Software, 2002, http://www-ext.crc.ca/spectrum-explorer/spectrum-explorer6/en/index.htm.

[6] D. Donoho, X. Huo, Large-sample modulation classification using Hellinger representation, Signal Process. Adv. Wireless Commun. (1997) 133–137.

[7] D.E. Goldberg, Genetics Algorithms: in Search Optimisation and Machine Learning, Addison-Wesley, Reading, MA, 1989.

[8] A. Hero, H. Hadinejad-Mahram, Digital modulation classification using power moment matrices, in: Proceedings of the IEEE 1998 International Conference on Acoustics, Speech and Signal Processing (ICASSP), Washington, Vol. 6, 1998, pp. 3285–3288.

[9] M. Ibnkahla, Applications of neural networks to digital communications—a survey, Signal Processing 80 (2000) 1185–1215.

[10] L.B. Jack, A.K. Nandi, Genetics algorithms for feature selection in machine condition monitoring with vibration signals, IEE Proc.: Vision Image Signal Process 147 (3) (2000) 205–212.

[11] R. Kohavi, D. Sommerfield, Feature subset selection using the wrapper method: overfitting and dynamic search space topology, in: Proceedings of First International Conference on Knowledge Discovery and Data Mining (KDD-95), Montreal, 1995, pp. 192–197.

[12] M. Kudo, J. Sklansky, Comparison of algorithms that select features for pattern classifiers, Pattern Recognition 33 (33) (2000) 25–41.

[13] P.L. Lanzi, Fast feature selection with genetic algorithms: a filter approach, in: Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, 1997, pp. 537–540.

[14] K. Nolan, L. Doyle, D. O'Mahony, P. Mackenzie, Modulation scheme recognition techniques for software radio on a general purpose processor platform, Proceedings of the First Joint IEI/IEE Symposium on Telecommunication Systems, Dublin, 2001.

[15] L. Mingquan, X. Xianci, L. Lemin, AR modeling based features extraction for multiple signals for modulation recognition, in: Signal Processing Proceedings, Fourth International Conference, Beijing, Vol. 2, 1998, pp. 1385–1388.

[16] L. Mingquan, X. Xianci, L. LeMing, Cyclic spectral features based modulation recognition, in: Communications Technology Proceedings, International Conference, Beijing, Vol. 2, 1996, pp. 792–795.

[17] B.G. Mobasseri, Digital modulation classification using constellation shape, Signal Processing 80 (2000) 251–277.

[18] A.K. Nandi, E.E. Azzouz, Modulation recognition using artificial neural networks, Signal Processing 56 (2) (1997) 165–175.

[19] A.K. Nandi, E.E. Azzouz, Modulation Recognition Using Artificial Neural Networks, Signal Processing 56 (2) (1997) 165–175.

[20] A.K. Nandi, E.E. Azzouz, Algorithms for automatic modulation recognition of communication signals, IEEE Trans. Commun. 46 (4) (1998) 431–436.

[21] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: the RPROP algorithm, in: Proceedings of the IEEE International Conference on Neural Networks, San Francisco, CA, 1993, pp. 586–591.

[22] C.L.P. Sehier, Automatic modulation recognition with a hierarchical neural network, in: Military Communications (MILCOM) Conference Records 1993: Communications on the move, Vol. 1, IEEE Press, New York, 1993, pp. 111–115.

[23] A. Swami, B.M. Sadler, Hierachical digital modulation classification using cumulants, IEEE Trans. Commun. 48 (3) (2000) 416–429.

[24] M.L.D. Wong, A.K. Nandi, Automatic digital modulation recognition using spectral and statistical features with multi-layer perceptrons, in: Proceedings of International Symposium of Signal Processing and Application (ISSPA), Kuala Lumpur, Vol. II, 2001, pp. 390–393.

[25] J. Yang, V. Honavar, Feature subset selection using a genetic algorithm, Intell. Systems IEEE 13 (2) (1998) 44–49.