Support Vector Machine Assisted Genetic Programming for MQAM Classification

Zhechen Zhu, Muhammad Waqar Aslam, Asoke Kumar Nandi
Department of Electrical Engineering and Electronics, the University of Liverpool, Liverpool, United Kingdom
{z.zhu, m.w.aslam, a.nandi}@liverpool.ac.uk

***Abstract*—Automatic modulation classification is used to identify automatically the modulation type of an incoming signal with limited or no prior knowledge to it. Various classifier systems have been developed to solve this problem. However, for certain types of modulations such as 16QAM and 64QAM, the classification performance under noisy condition still needs to be improved. In this paper, we propose a new AMC scheme by combining genetic programing (GP) with support vector machine (SVM) for the classification of 16QAM and 64QAM signals. The benchmark result shows that SVM assisted GP can produce better accuracy than some other existing methods.**

Keywords: QAM, modulation classification, support vector machine, genetic programming

I. INTRODUCTION

Automatic modulation classification (AMC) is a key step in many military and civilian communication applications [1]. Recent development in software defined radio (SDR) and cognitive radio (CR) has largely motivated the research in AMC [2]. Adaptive modulation is one of the essential tasks of both SDR and CR where AMC plays the role of connecting the detection and demodulation of incoming signals. A proper adaptive modulation scheme helps to improve the communication efficiency in varying channel conditions as well as to minimize the radio hardware design. As the selection of modulation type used by the transmitter is based on the channel condition, the receiving end needs to identify the modulation type correctly so that the signals can be accurately demodulated for further processing. Among all the modulation types, quadrature amplitude modulation (QAM) has been widely used in many communication systems. For example, 16QAM is used in modems, 4QAM and 64QAM used in WLAN, DVB, and WiMAX systems [21]. Also as compare to other modulation types, higher order QAM is much more difficult to classify [3]-[4]. Therefore, this study is focused on two high order QAM modulations namely 16QAM and 64QAM.

In general, there are two groups of AMC methods: the likelihood based solutions and feature based pattern recognition solutions. Maximum Likelihood (ML), which has been previously proposed in [5]-[6], formulates the AMC problem with probabilistic and hypothesis testing arguments and works to minimize the probability of false classification. However, it needs an accurately matched underlying probability density function model to achieve optimum results which makes it vulnerable to phase and frequency offsets, residual channel effect, timing errors and non-Gaussian noises. Also it has a high computational complexity demand, which Wong and Nandi attempted to address in [7] by using minimum distance classifier. Different from ML, pattern recognition methods identifies the modulation by extracting features from the signals and then processing these features with pre-designed rules for a decision. It has been favored by many recent studies in AMC because of its simplicity, robustness and high performance when properly designed. There are many possible combinations of features used and ways to set the classification rules. In [8] Azzouz and Nandi used features calculated from instantaneous amplitude and phase for AMC with neural network taking the role of classification decision making. Different features were proposed by Swami and Sadler in [3], where features were extracted based on high order statistics. They used fourth order cumulants with a hierarchical classification scheme. Wong and Nandi [9] also used the same features with a combination of artificial neural network (ANN) and genetic algorithm (GA). Similarly, Shermeh and Ghazalian [10] used GA for feature selection but combined with a SVM classifier. The selection of features can optimize the efficiency of the classifier and may improve the performance by a small margin. However, it is very likely that some information might be lost with features of less contribution to the overall performance being neglected.

Recently, we proposed to use GP as a feature generator which combines existing features into a single new feature. By doing so, the efficiency of the classifier can be improved as there is only one input feature to be processed. In addition, the classification performance can be improved as all features are being considered while only be given different weights in the formulation of the new feature. In [11] such GP feature generator is combined with K-nearest neighbor to form a complete classification scheme. Good performance was achieved between BPSK, QPSK, 16QAM and 64QAM. However, the discrimination of 16QAM and 64QAM was not very successful especially under noisy condition. As KNN is a relatively simple classifier, we replaced it with SVM in this paper for a possible performance improvement.

The paper is organized as follows. Section II introduces the signal model and the statistical characterization. It also explains the features used by GP. The basic structure of GP and a complete model of AMC system have been given in Section III. Experimental results and comparison of performance with other results are given in Section IV and V respectively, while the conclusions are drawn in Section VI.

978-1-61284-943-0

II. SIGNAL MODEL AND FEATURE EXTRACTION

A. Signal Model

For any communication systems, the baseband waveform at the receiver can be written as

$$y(n) = Ae^{j(2\pi f_0 nT+\theta_n)} \sum_{I=-\infty}^{\infty} x(I)h(nT - IT + \in_T T) + g(n) \quad (1)$$

where $x(I)$ is the symbol sequence, A is unknown amplitude factor, T is the symbol spacing, $\in_T$is timing error, f_0 represents carrier frequency offset, h(.) is the residual baseband channel effects, θ_n is the phase jitter which may vary from sample to sample, and $g(n)$ is additive, white, Gaussian noise (AWGN). We assume ideal working condition with presence of white Gaussian noise only with the assumption that channel had been equalized and the residual channel effect is $h(.)$ negligible. Also other parameters, such as T (the symbol timing), f_0 (the carrier frequency offset), etc., are assumed to be known.

B. Feature Extraction

The feature based recognition approach has been used for AMC in this paper. Fourth and sixth order cumulants of received signals have been used as the underlying features. Cumulants are made up of moments of received signals, so various moments have been used as features too. For a complex valued stationary signal the cumulants can be defined as shown below [12]

$$\begin{aligned}
C_{40} &= cum(y(n), y(n), y(n), y(n)) = M_{40} - 3{M_{20}}^2 \\
C_{41} &= cum(y(n), y(n), y(n), y^*(n)) = M_{40} - 3M_{20}M_{21} \\
C_{42} &= cum(y(n), y(n), y^*(n), y^*(n)) \\
&= M_{42} - |M_{20}|^2 - 2{M_{21}}^2 \\
C_{60} &= cum(y(n), y(n), y(n), y(n), y(n), y(n)) \\
&= M_{60} - 15M_{20}M_{40} + 30{M_{20}}^3 \\
C_{61} &= cum(y(n), y(n), y(n), y(n), y(n), y^*(n)) \\
&= M_{61} - 5M_{21}M_{40} - 10M_{20}M_{41} + 30{M_{20}}^2 M_{21} \\
C_{62} &= cum(y(n), y(n), y(n), y(n), y^*(n), y^*(n)) \\
&= M_{62} - 6M_{20}M_{42} - 8M_{21}M_{41} - M_{22}M_{40} + 6{M_{20}}^2 M_{22} + \\
&\quad 24{M_{21}}^2 M_{20} \\
C_{63} &= cum(y(n), y(n), y(n), y^*(n), y^*(n), y^*(n)) \\
&= M_{63} - 9M_{21}M_{42} + 12{M_{21}}^3 - 3M_{20}M_{43} - 3M_{22}M_{41} + \\
&\quad 18M_{20}M_{21}M_{22} \qquad (2)
\end{aligned}$$

M_{pq} represents the moment of a signal which is defined as

$$M_{pq} = E[y(k)^{p-q}(y^*(k))^q]] \quad (3)$$

III. METHOD

A. Genetic Programming

Genetic Programming is a machine learning methodology which is inspired by biological evolution on the development of computer programs [13]. In most classification applications, these programs are actually expressions represented by tree structures. The tree structure, as shown in Fig. 1, has a single output computed based on the arrangement of inputs at different terminals and functions at different nodes. This single output is used for classification in different ways according to the classification strategy. The entire GP process starts with a randomly generated initial generation. The evolution of these tree structures is essentially directed by the fitness evaluation which evaluates the performance of a tree in solving a specific problem. During reproduction, trees with better fitness are selected for different operations including crossover and mutation to generate new trees. Crossover takes two parent trees and selects one branch from each then swaps the selected branches to make two new children trees. An example of crossover operation is shown in Fig. 1. Mutation on the other hand, takes only one parent tree and one branch from it. Instead of using an existing branch from another tree, mutation replaces the original branch with a randomly generated new branch. With such a reproduction approach, the new trees are expected to inherit some of the useful traits from the last generation and manage to achieve better fitness with gradual partial changes. At the end of the evolution, the best tree ever produced in the whole evolution process is selected as a final solution.

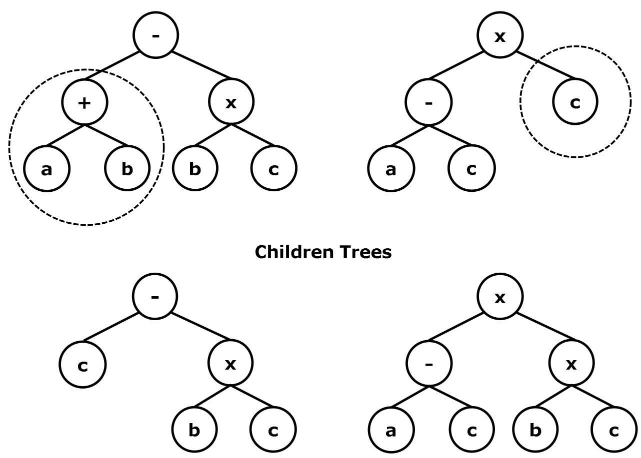

Figure 1. Crossover operation with two parent tress and two children Trees.

GP has been applied in some classification problems [14] – [17]. In [14] a survey is given on the application of genetic programming for classification purpose. Previous study [18] by Loveard and Ciesielski has suggested that GP has the advantage in classification that the more time given for evolution the better performance can be achieved and for different runs, due to the randomness in parent selection and reproduction operation, different final solutions could be generated which lends GP classifiers well to a voting strategy to improve the performance.

In this paper, GP is used as a feature generator for the SVM classifier. During the GP evolution, different tree structures were formed with different combination of existing high order cumulant features to output a single value as the new feature. While one can simply use the existing features for SVM classification, a combination of selected features should return a better performance. GP in this case automates this complex process in an intelligent way. Once GP returns the best combination of features in the form of a tree, the tree is evaluated with the assistance of SVM.

B. Support Vector Machine

Support vector machine, developed by Vapnik [19], is also a relatively new machine learning technique. As a classifier, SVM separate the binary labeled training date with a hyper-plane that maximizes the distance from one class to another. When no linear separation is possible, different 'kernels' can be introduced to realize a non-linear mapping of a feature space. The hyper-plane found by the SVM in feature space corresponds to a non-linear decision boundary in the input space.

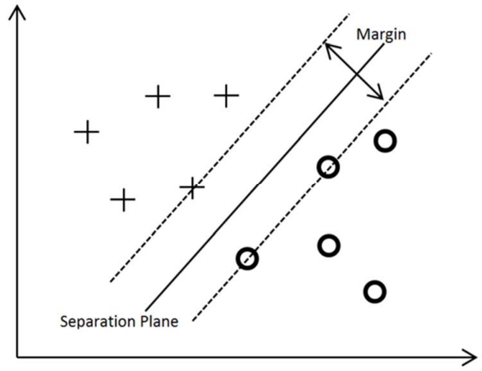

Figure 2. Two class feature plane with optimal separation plane

In this paper SVM was used for fitness evaluation in genetic programming as well as the end classifier. When GP is used for feature generation, the outcome is normally affected by different functions used and the randomness of the process. The resulting feature distribution can be very complex. Therefore, it is very difficult to use a simple linear boundary to separate samples from different classes. Even if an optimal linear can be established, it can hardly return a fair evaluation of the feature generated. That is why SVM is adopted in this paper to accurately evaluate the feature. The example shown in Figure 3 is a good demonstration of how SVM can be implemented on a new feature. As can be seen in this figure, the 16QAM samples are divided into two areas by the 64QAM samples. However, if one looks closer at the distribution, still the samples from both classes can be separated by a non-linear boundary. After applying SVM with a 3rd order polynomial kernel function, a non-linear boundary is easily established for a good separation.

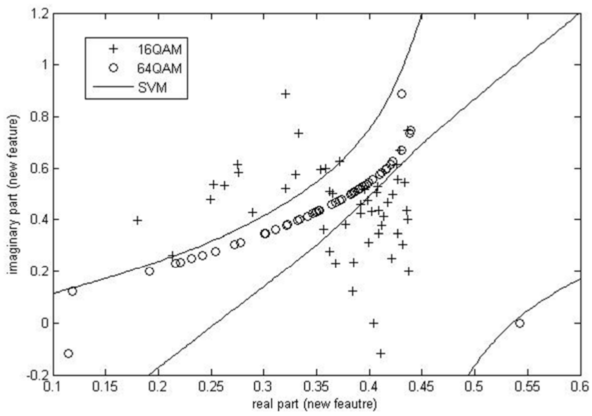

Figure 3. 16QAM and 64QAM samples on the new feature plane.

1) SVM for Fitness Evaluation

During the fitness evaluation, new features of 200 signal samples from each modulation class were calculated from every individual tree. These samples were then fed into the SVM for obtaining the optimum separation plane between the two classes. After the boundary is established, the total number of error samples e which were located on the wrong side of the separation plane is counted as the raw value for the fitness calculation. The fitness function is given below where $e_{(16QAM,64QAM)}$ is the total number of 16QAM samples being located on the 64QAM side of the separation plane and $e_{(64QAM,16QAM)}$ is the total number of 64QAM samples being located on the 16QAM side.

$$\mathrm{f} = e_{(16QAM,64QAM)} + e_{(64QAM,16QAM)} \tag{4}$$

As for a better feature, the value would be lower which means a better separation between the two classes in the feature space. Then the trees which produce these better features became more likely to be picked for further genetic reproduction. Because of the complexity of genetic programming, SVM was trained with a 3rd order polynomial kernel function for a more efficient feature evolution process. The resulting SVM model is shown in Figure 4.

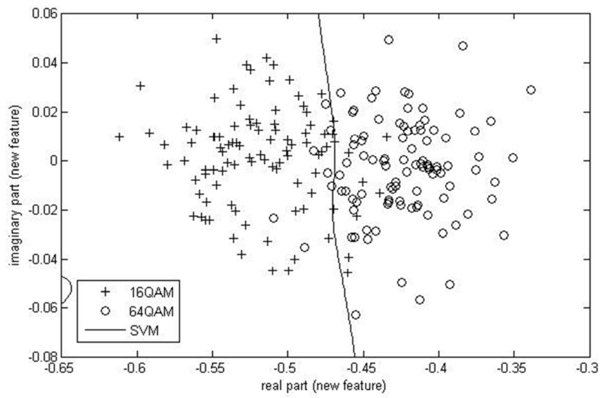

Figure 4. New feature space with 200 signal realizations from 16QAM and 64QAM. Each realization consists of 512 samples with SNR of 10 dB. The boundary shown in the figure is the trained separation plane from a SVM with 3rd order polynomial kernel function.

2) SVM for Classification

In the testing stage, where the new feature has been established and the SVM is used for classification test, a new SVM was trained to be the final classifier. As only a single SVM is to be trained in this stage, it could be designed to be more complex than the ones used in the fitness evaluation process so that better classification performance can be achieve. Here, 400 samples of signals were generated as training data for the new SVM with a fifth order polynomial kernel function. The resulting SVM model is shown in Figure 5.

After the training for the SVM classifier is finished, only the new feature and trained separation plane was passed on to the final classification test. Training samples were not used in the testing stage where new testing data were generated. Details of

978-1-61284-943-0

performance tests conducted on the proposed system will be given in the next section.

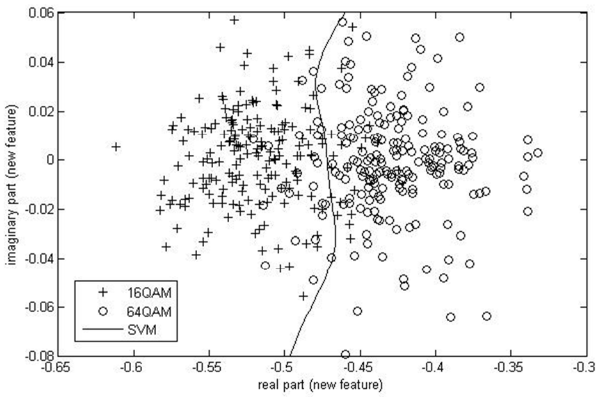

Figure 5. New feature space with 400 signal samples from 16QAM and 64QAM under. Each realization consists of 512 samples with SNR of 10 dB. The boundary shown in the figure is the trained separation plane from a SVM of 5th order polynomial kernel function.

C. GP and SVM Parameters

All experiments were conducted under MATLAB environment with Bioinformatics Toolbox by Mathworks and a third party GPlab Toolbox (http://gplab.sourceforge.net/). For the GP programs used in this paper, 100 generations with 25 individuals in each generation were generated. However, the program can be terminated if any of the individuals reached perfect fitness. Other parameters are listed in Table I. More information of the parameters in GPlab Toolbox can be found on the toolbox website.

TABLE I. GENETIC PROGRAMMING PARAMETERS

Parameters	Values
Number of Generation	100
Population Size	25
Function Pool	plus, minus, times, reciprocal, negator, abs, sqrt, sin, cos, tan, asin, acos, tanh, mylog[a]
Terminal Pool	C_{40}, C_{41}, C_{42}, C_{60}, C_{61}, C_{62}, C_{63}
Genetic Operator	crossover, mutation
Operator Probability	0.9, 0.1
Tree Generation	ramped half-and-half
Initial Maximum Depth	28
Selection Operator	lexictour
Elitism	replace

a. mylog is a protected $\log_e x$ function which ignores the input x if it is zero.

For the SVM used in this paper, 200 samples of both 16QAM and 64QAM signals are used for fitness evaluation with the kernel function set to 3rd order polynomial. In the testing stage, 400 samples were used for training of the classifier with the order of polynomial kernel increased to 5th order. In both stages, quadratic programming was used as the method to find separation plane. All the parameters are listed in Table II.

TABLE II. SUPPORT VECTOR MACHINE PARAMETERS

Parameters	Values	
	Fitness Evaluation	***Testing Stage***
Number of Training samples	200	400
Number of Testing samples	N/A	10,000
Kernel Function	Polynomial	Polynomial
Polynomial Order	3	5
Method	Quadratic Programming	Quadratic Programming

IV. EXPERIMENTS AND RESULTS

In this study, experiments were conducted for signals with two different sample lengths (512 and 2048) and SNR values (10 dB and 20 dB). The noise is assumed to be AWGN and samples are assumed to be adequately equalized.

In the first stage, each GP program was run for 100 generations with a population of 25. The parameters used in GP have been given in section III. The fitness function used has been explained in Section III. In total, each GP run was repeated for 10 times to obtain 10 best individuals. While a total number 25000 GP trees were created in the entire evaluation process, only a single tree was picked from the 10 best individuals. The selection decision is based on results from tests conducted which will be described in the next paragraph but with a smaller testing sample size of 1000 realizations.

After a best new feature was selected, it is tested for its classification performance with SVM classifier and a large number of signal samples. For each value of SNR and the number of samples, 10,000 realizations of noise corrupted data for each modulation type were generated. So in total 2 sets were generated and as there are 2 modulation types, so each set had 20,000 realizations. These 20,000 realizations were tested with the best tree and results are presented in Table III.

TABLE III. CLASSIFICATION ACCURACY FOR DIFFERENT SNRS AND NUMBER OF SAMPLES

SNRs	512	2048
10 dB	90.3%	99.8%
20 dB	99.9%	100.0%

V. DISCUSSION

With similar higher order cumulants feature based AMC scheme Swami and Sadler [4] reported a classification accuracy of 90% between 16QAM and 64QAM. However, signals of more than 10,000 samples and noise free condition is needed to achieve such performance. Compared to that, we used only 2048 samples under a SNR of 20 dB. Dobre, Ness and Su used cyclic cumulants in [20] for classifying 16QAM and 64QAM. The result they achieved was an accuracy of 70% at 10 dB with 2000 samples. Under the similar condition, our result is about 99.8%.

By applying Naïve Bayes and SVM with the high order cumulants, Wong and Nandi [4] reported a performance of 89.98% and 91.23% for classifying BPSK, QPSK, 16QAM and 64QAM signals with 512 samples and SNR of 10 dB. In [11], when GP is combined with KNN, it gives a classification accuracy of 94.7% between the same modulations considered in

[4]. Considering the fact that the identification of BPSK and QPSK can easily achieve 100% accuracy, the classification results between 16QAM and 64QAM is recreated and to be compared with performance of GP-SVM in Table IV.

TABLE IV. ACCURACY OF PERFORMANCE RESULTS IN 4-CLASS CLASSIFICATION AT AN SNR OF 10 DB

Number of samples	GP-KNN	GP-SVM
512	89.4%	**90.3%**
2048	99.6%	**99.8%**

VI. CONCLUSION

For the purpose of better utilizing the existing AMC features and achieving better classification performance, this paper implemented a scheme using GP for feature generation with SVM assisting its fitness evaluation and final classification. While GP contributes more to the final performance, the results also showed that the use of a more sophisticated classification tool such as SVM is still beneficial to the final classification results. In addition, this combination is not fully justified as the setup for SVM is yet fully optimized. Also, other advanced classification tool can be tried for performance enhancement.

ACKNOWLEDGEMENTS

Zhechen Zhu would like to acknowledge the financial support from the University of Liverpool (U.K.). Muhammad Waqar Aslam would like to thank the University of Azad Jammu & Kashmir Pakistan for financial support.

REFERENCES

[1] E. E. Azzouz and A.K. Nandi, Automatic Modulation Recognition of Communication Systems. Norwell, MA: Kluwer, 1996.

[2] K. E. Nolan, L. Doyle, P. Mackenzie, and D. O'Mahony, "Modulation scheme classification for 4G software radio wireless networks," in Proc. IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 25-31, 2002.

[3] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," IEEE Trans. Commun., vol. 48, pp. 416-429, 2000.

[4] M. L. D. Wong and A. K. Nandi, "Naïve Bayes classification of adaptive broadband wireless modulation types with higher order cumulants," in Proc. of the International Conference on Signal Processing and Communication Systems, 2008.

[5] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," IEEE Trans. Commun., vol. 48, pp. 189-193, 2000.

[6] T.Yucek and H. Arslan, "A Novel Sub-optimum Maximum Likelihood Modulation Classification Algorithm for Adaptive ODFM System," in Proc. of IEEE Wireless Communications and Networking Conference (WCNC), pp.739-744, 2004

[7] M. L. D. Wong and A. K. Nandi, "Semi-blind algorithms for automatic classification of digital modulation types," Digital Signal Processing, vol. 18, pp. 209-227, 2008.

[8] 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.

[9] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," Sig. Proc., vol. 84, pp. 351-365, 2004.

[10] A.E. Shermeth, R. Ghazalian, "Recognition of communication signal types using genetic algorithm and support vector machine based on the higher order statistics," Digitial Signal Process, vol. 20, pp. 1748-1757, 2010

[11] Z. Zhu, M.W. Aslam, and A.K. Nandi, "Augmented genetic programming for automatic digital modulation classification," in Proc. IEEE MLSP 2010, pp. 391-396, 2010

[12] M. R. Mirarab, M. A. Sobhani, "Robust modulation classification for PSK/QAM/ASK using higher order cumulants," in Proc. of the Sixth International Conference on Information, Communications and Signal Processing, 2007.

[13] J. R. Koza. Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[14] P. G. Espejo, S. Ventura, F. Herrera, "A Survey on the Application of Genetic Programming to Classification," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 40, pp. 121-144, 2010.

[15] J. Eggermont, A. E. Eiben and J. I. van Hemert, "A comparison of genetic programming variants for data classification," In Proc. on the Third Symposium on Intelligent Data Analysis (IDA), 1999.

[16] C. Gathercole and P. Ross. "Dynamic training subset selection for supervised learning in genetic programming," In Parallel Problem Solving from Nature III, pp. 312-321, 1994.

[17] H. Gray, "Genetic programming for classification of medical data," In Late Breaking Papers at the Genetic Programming Conference, pp. 291-297, 1997

[18] M. Zhang, V. B. Ciesielski, and P. Andreae, "A domain independent window approach to multiclass object detection using genetic programming," EURASIP Journal on Applied Signal Processing8, pp. 841-859, 2003.

[19] V. Vapnik. Statistical Learning Theory, Wiley, 1998.

[20] O. A. Dobre, Y. Bar-Ness, and W. Su, "Higher-order cyclic cumulants for high order modulation classification," in Proc. IEEE MILCOM, pp. 112-117, 2003.

[21] Todor Cooklev, "Wireless Communication" Standards: A study of IEEE 802.11, 802.15, and 802.16, IEEE Standards Association, 2004. ISBN 073814066X

978-1-61284-943-0

This Page Was Intentionally Left Blank