Modulation Recognition in the 868 MHz Band Using Classification Trees and Random Forests

Ken Lau, Matias Salibián-Barrera, and Lutz Lampe

***Abstract*—Automatic modulation recognition (AMR) enables the detection of different data-transmission formats sharing the same frequency band. One such band is the European 868 MHz band that is dedicated for short-range devices. Recently, an AMR method that applies a feature-based tree has been proposed for this band. In this paper, we present alternative feature-based classifiers that enable a more accurate AMR. In particular, we propose the use of classification tree and random forest classifiers, and we devise an extended set of features for the modulation classification problem at hand. Through simulation experiments we demonstrate a significant improvement in recognition success rate for typical transmission types in the 868 MHz band.**

***Index Terms*—Automatic modulation recognition, Classification Tree, Random Forest.**

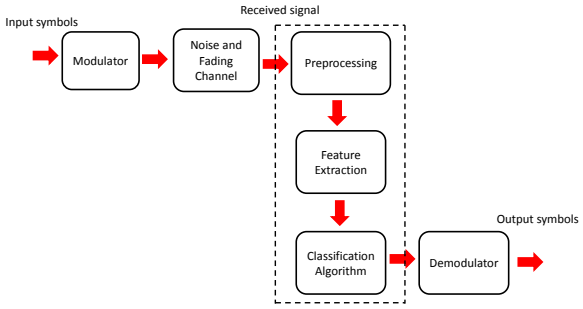


Fig. 1: System block diagram of an AMR task.

I. Introduction

THE process of automatic modulation recognition (AMR) concerns the detection of an unknown transmission format or transmit-signal type based on a received signal. AMR is used in various civilian and military communication applications for tasks such as signal confirmation and interference identification [1], [2]. AMR has also gained popularity due to the development of software-defined radio (SDR) and cognitive radio (CR) concepts [2], [3]. For those applications, an effective AMR classifier can be used to support multiple transmission formats over a white noise or fading channel and to avoid interference.

The integration of AMR in a communication system is illustrated in Figure 1. AMR is an intermediate step between reception and demodulation that involves preprocessing the received signal, extracting features, and feeding the information to a classification algorithm. This process is marked with a black dashed rectangle in the figure. The role of AMR is to identify the correct transmit-signal parameters, including the use of spreading, so that subsequent demodulation and decoding can properly be employed.

An important application case for AMR is the transmission in the European 868 MHz band for short-range devices. Since different communication standards and systems use this band, [4] proposed the application of AMR to enable a communication gateway that translates waveforms from different devices. In particular, AMR based on a feature-based classifier was applied in [4], [5]. Feature-based classifiers extract relevant features from the received signal, which are then used in a classifier to determine the transmission format. For example, [6] incorporates higher-order statistical (HOS) features in conjunction with a classification tree (CT) method to classify M-ary phase shift keying (PSK) and M-ary quadratic amplitude modulation (QAM). A very similar approach to AMR is shown in [7], which also uses HOS features, but constructs artificial neural networks (ANNs) for classification. ANNs have long been applied for AMR for both analog and digital modulation formats, cf. e.g., [8]. Furthermore, support vector machine (SVM) (e.g. [9]) and genetic algorithms (GAs) (e.g. [1], [10]) have also been used for AMR. Training SVMs requires extensive and careful tuning, which includes choosing a kernel function and its parameters, the regularization penalty (soft-margin constant), and also selecting a strategy to perform multi-class classification [11]. For example, [9] uses particle swarm optimization to select the tuning parameters of SVMs to classify digitally modulated signals. Training ANNs presents similar challenges, and decision trees have been shown to perform very well compared with other alternatives [8], [12]. Simpler classifiers, like Naïve Bayes, have generally not done better than decision trees in the context of AMR [13], [14]. This is not surprising given that many features used to classify modulations can exhibit strong correlations [15], [16].

The inclusion of a large number of features often poses additional challenges due to the increased complexity of the prediction model. The work done in [7] handles this problem by using principle component analysis (PCA) to select the most useful features for classification. For an overview of common feature variables and classification algorithms applied to AMR tasks we refer to [2], [17].

In this paper, we extend the approach of [4], [5] considering AMR for transmission in the European 868 MHz band. The classifier presented in [4] and [5] is a feature-based tree

K. Lau and M. Salibián-Barrera are with the Department of Statistics, University of British Columbia, Vancouver, British Columbia, Canada (Email: matias@stat.ubc.ca). L. Lampe is with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, British Columbia, Canada (Email: lampe@ece.ubc.ca).

This work was supported by the Natural Sciences and Engineering Council of Canada (NSERC).

(FBT) using two and five features, respectively. In addition, the features appear exactly once in the tree. We improve this approach by including an extended feature set and utilizing a CT [18] and a random forest (RF) [19]. In particular, we include HOS features and use variable importance (instead of PCA as in [7]) to deal with highly complex models. We chose RF classifiers, which to the best of our knowledge have not been studied for AMR yet, as they provide superior predictive accuracy in many applications due to its utilization of the consensus of individual trees as weak learners. It is shown in Section 8.7 of [20] that using an ensemble of trees improves accuracy and precision of predictions. Furthermore, because of the ensemble method together with the way the RF algorithm selects only a subset of features at each decision threshold of a tree, allows the algorithm to handle many features without overfitting. Numerical results demonstrate that the proposed CT and RF classifiers outperform the FBT approach over a wide range of signal-to-noise ratios (SNRs), and yield significantly higher success rates for AMR.

II. System Model

Different communication standards specify a variety of communication protocols for the European 868 MHz band for short-range communication. We adopt the system model from [5] and consider six important transmission formats. The first three formats are on-off keying (OOK), binary phase-shift keying (BPSK) and offset quaternary phase-shift keying (OQPSK), and all of them have a carrier frequency of 868.3 MHz. The other three formats are binary frequency-shift keying (BFSK) operating at carrier frequencies of 868.3 MHz, 868.95 MHz, and $868.03 + i \cdot 0.06$ MHz, where $0 \leq i \leq 9$, respectively. These are used in the wireless meter-bus specification and, according to this specification, we denote the three signal types as BFSK-A, BFSK-B, and BFSK-R2, respectively. Further details about the six communication signals are provided in [5].

Since the signals are narrowband, we apply a frequency-flat slowly fading channel model. At the AMR receiver, the signals received in the 868 MHz band are mixed to a low intermediate frequency. For our numerical results presented in Section V, we apply down-conversion by 867.3 MHz, i.e., the intermediate frequency for OOK modulation would be 1 MHz. The signal at intermediate frequency is sampled with sampling frequency f_s, resulting in the discrete-time signal $r[k]$, where k is the discrete-time index. As in [5], we apply a sampling frequency $f_s = 6.25$ MHz. The received signal can be expressed as

$$r[k] = gs[k] + n[k] \,, \tag{1}$$

where $s[k]$ is the down-converted and sampled signal component, g is the channel gain, which varies very slowly compared to the signaling rate, and $n[k]$ is white Gaussian noise with variance σ_n^2. Since AMR is done before any signal demodulation and detection (see Figure 1), no time synchronization or filtering are applied, and thus $s[k]$ is an arbitrarily time and phase shifted version of the transmitted signal. Denoting the variance of $s[k]$ by σ_s^2, the instantaneous SNR for AMR processing is $g^2\sigma_s^2/\sigma_n^2$. The AMR results only depend on the SNR, and not on the absolute values of g, σ_s^2, or σ_n^2.

The task of the AMR method is to determine from the received signal $r[k]$ which of the six transmission formats has been used in $s[k]$. To this end, features are extracted from $r[k]$ and then used in a classifier. The features and classifiers applied in previous and our AMR approaches will be introduced in the following two sections.

III. Feature Extraction

In this section, we describe the preprocessing and feature extraction required for classification. First, the preprocessing and features utilized in the FBT of [5] are briefly introduced. Next, we present a set of commonly used engineered features that we deem to contribute positively for classification with CT and RF.

A. Preprocessing

The task of AMR begins by preprocessing N samples, $r[0], \ldots, r[N-1]$, of the sampled received signal in (1). Following [5], we adopt $N = 512$ and generate via discrete Fourier transform (DFT) the normalized spectral representations

$$R[k] = \sum_{i=0}^{N-1} \frac{r[i]}{\mu_r} \mathrm{e}^{-\mathrm{j}2\pi \frac{ik}{N}} \,, \quad k = 0, \ldots, N-1 \tag{2}$$

and

$$A[k] = \sum_{i=0}^{N-1} \frac{|a[i]|}{\mu_a} \mathrm{e}^{-\mathrm{j}2\pi \frac{ik}{N}} \,, \quad k = 0, \ldots, N-1 \,, \tag{3}$$

where

$$a[k] = r[k] + \mathrm{j}\mathcal{H}\{r\}[k] \,, \quad k = 0, \ldots, N-1 \tag{4}$$

is the analytical received signal and

$$\mu_a = \frac{1}{N} \sum_{k=0}^{N-1} |a[k]| \,, \quad \mu_r = \frac{1}{N} \sum_{k=0}^{N-1} |r[k]| \,. \tag{5}$$

In (4), $\mathcal{H}\{\cdot\}$ denotes Hilbert transformation.

B. Features from [5]

In [5], five features, referred to as m_1, m_2, …, m_5, were derived based on $R[k]$ and $A[k]$ from (2) and (3), respectively. For example, m_1 is computed as

$$m_1 = \max_{1 \leq k_1 \leq 7} (|A[k_1]|) + \max_{25 \leq k_2 \leq 27} (|A[k_2]) \,. \tag{6}$$

The features were designed to discriminate the transmission formats introduced in Section II. To avoid being repetitive, we refer to [5] for the derivation of these features.

C. Additional Features

We suggest the use of additional features for the task of AMR in the 868 MHz band. The type of features we describe are commonly used in AMR problems, cf. e.g. [1], [3], [17], [21], [22].

TABLE I: Summary of feature variables.

Description	Features
Features from [5]	m_1, m_2, …, m_5
Transformation-based	$\gamma_{\max}$, Γ_2, Γ_4
Higher-order cumulant	C_{II}, C_{IQ}, C_{QQ}, C_{III}, C_{IIQ}, C_{IQQ}, C_{QQQ}, C_{IIII}, C_{IIIQ}, C_{IIQQ}, C_{IQQQ}, C_{QQQQ}
Magnitude spectrum	$\{\|R[k]\| \mid k = 45, \ldots, 160\}$

1) Transformation Based Features: The first set of proposed features is related to the spectral representation of the analytical signal. More specifically, we compute the maximal squared magnitude frequency component

$$\gamma_{\max} = \max_{k=0,\ldots,N-1} \frac{1}{N}|A[k]|^2. \tag{7}$$

Since $\gamma_{\max}$ corresponds to amplitude variation of signals [17], this feature is useful in discriminating between OOK and the PSK/FSK modulations. Furthermore, the maximum value of the DFT of the 2^{nd} and 4^{th} power of the analytical form is computed, i.e.,

$$\Gamma_n = \max_{k=0,\ldots,N-1} \frac{1}{N}\left|\sum_{i=0}^{N-1}\left(\frac{|a[i]|}{\mu_a}\right)^n \mathrm{e}^{-\mathrm{j}2\pi\frac{ik}{N}}\right|^2, \tag{8}$$

for $n = 2$ and $n = 4$. The features, Γ_2 and Γ_4 are useful in classification of PSK signals, specifically between BPSK and OQPSK modulations [17].

2) Higher-Order Cumulant Features: Higher-order cumulant (HOC) features capture information about amplitude and phase distributions of modulations and are not influenced by additive white Gaussian noise for cumulants with order greater than two (see [1] for an in-depth discussion of the advantages of cumulant features for AMR). In this work, time-averaging approximations of the second, third, and fourth order cumulants are computed:

$$\begin{aligned} C_{x_1x_2} &= \frac{1}{N}\sum_{k=0}^{N-1} x_1[k]x_2[k] \\ C_{x_1x_2x_3} &= \frac{1}{N}\sum_{k=0}^{N-1} x_1[k]x_2[k]x_3[k] \\ C_{x_1x_2x_3x_4} &= \frac{1}{N}\sum_{k=0}^{N-1} x_1[k]x_2[k]x_3[k]x_4[k] - C_{x_1x_2}C_{x_3x_4} \\ &\quad - C_{x_1x_3}C_{x_2x_4} - C_{x_1x_4}C_{x_2x_3} \end{aligned} \tag{9}$$

where $x_i[k]$, $i = 1, \ldots, 4$, is either the in-phase or quadrature component of the normalized analytical signal $a[k]/\mu_a$, which will be denoted by $x_i = I$ and $x_i = Q$, respectively, for a total of 12 HOC features.

3) Magnitude Spectrum: The last set of proposed features consists of the normalized magnitude spectrum, $|R[k]|$ for frequency indices of $k = 45$ to $k = 160$ for a total of 116 magnitude spectrum features. These features are particularly useful in detecting the BFSK-B modulation, as it is modulated at a higher carrier frequency than the remaining transmission formats. Table I summarizes a total of 136 features used in the following.

TABLE II: The training data where the first column consists of the six transmission formats, and the columns to the right corresponds to features m_1 to $R[160]$ correspond to the feature variables.

Transmission format	m_1	m_2	m_3	...	$R[160]$
OOK	190	0.05	0.3	...	2.5
OOK	186	0.12	0.21	...	1.8
OOK	172	0.08	0.25	...	3.1
..	..	..	..	..	..
BFSK-R2	15.3	0.64	0.03	...	5.4
BFSK-R2	14.4	0.54	0.08	...	3.3
BFSK-R2	89.7	0.35	0.12	...	2.48

IV. Classification

We now turn our attention to the construction of reliable classifiers for the AMR problem. A classifier is a function that maps feature variables onto a set of classes, in our case, the different signal types. We briefly revisit the FBT of [5], and then introduce the classification tree and random forest classifiers. A training set is required to select the parameters of each classifier. The training data could be generated experimentally or, as in our work, from a Monte-Carlo simulation (see Section V for more information). An example of the training data is shown in Table II where the columns $m_1, m_2, \ldots, R[160]$ correspond to the features.

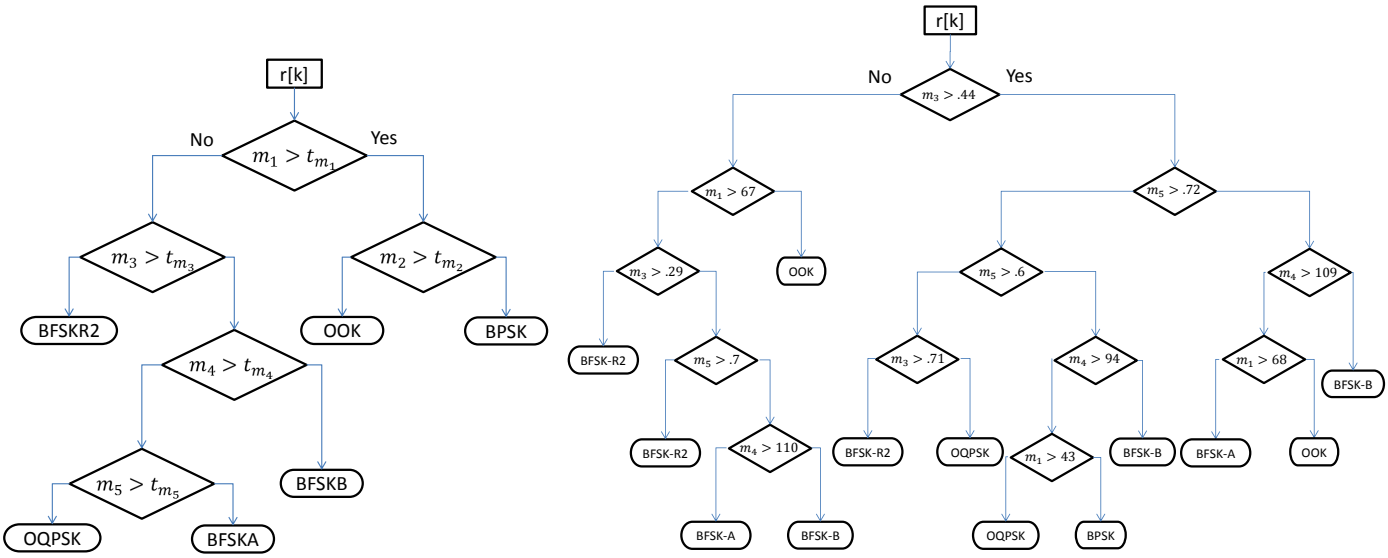
A. Feature-Based Tree (FBT)

The FBT algorithm in [5] utilizes 5 feature variables m_i, $i = 1, \ldots, 5$ and selects thresholds t_{m_i} to discriminate among different signal types. The tree is presented on the left panel of Figure 2 where the first split is based on feature m_1. The derivation of the decision thresholds is explained in [5]. The transmission format of a new signal is predicted by extracting the corresponding features m_i and traversing down the decision tree until an end node is reached.

B. Classification Tree (CT)

Similar to the FBT above, a CT is constructed with a series of binary splits of the available features. Furthermore, CTs use an algorithm to automatically select which features and threshold values to use at each node in order to obtain highly homogeneous (in terms of signal types) partitions of the training data (see [20] for more details). It is easy to see that increasing homogeneity at leaf nodes improves predictive accuracy. Consider Figure 3 where the parent node depicts red circles and blue crosses as two possible outcomes. If, for example, we were to predict that observations falling in the parent node are circles, we can expect to have an error rate of 50%, since half the training set in the node are crosses. However, if we split the node along the dashed line, we obtain child nodes with increased homogeneity. If we classify points in the left node as circles, and those in the right node as crosses, our error rate is now 20%.

To construct a CT we begin with all observations in the root node of the tree. In its simplest form, the algorithm then recursively splits the data into binary partitions until



(a) Feature-Based Tree

(b) Classification Tree

Fig. 2: Flowchart of trees depicting features and decision thresholds at interior nodes, and signal types at the leaf nodes.

all the observations at the leaf nodes are of the same class. Early stopping rules and pruning (discussed below) are also used to prevent overfitting. Fix a node, and let Y denote the transmission format observed by randomly drawing a signal from this node. If $p_k = P(Y = k)$ then the entropy (H) of Y is

$$H(Y) = -\sum_k p_k \log(p_k) \ . \tag{10}$$

The decision thresholds are determined by choosing the feature and threshold minimizing the entropy from the resulting splits. Typically only a subset of features are selected in the output of the tree, and these features can appear more than once in the tree.

To avoid constructing a classifier that over-fits the training set, a pruning step is recommended. The strategy is to grow a large tree and then use cross-validation to estimate the optimal trade-off between the competing goals of maximal homogeneity and a small number of terminal nodes. We apply the CT using features from Section III-B with and without the additional features introduced in Section III-C. The implementation of pruning a fully grown tree is carried out in rpart, which uses 10-fold cross-validation to determine the optimal size tree. In our experiments, we trained CTs using the `rpart` package in R [23].

The right panel of Figure 2 illustrates an example of a CT based on the features from [5]. Note, for example, that feature m_2 is not used in the CT while m_1 is used more than once, which emphasizes the difference between the CT and FBT.

C. Random Forests

To reduce prediction variability, RFs combine the predictions of a relatively large number of CTs obtained by bootstrapping the original data [19], [20]. Furthermore, at each split of each bootstrapped CT, RF only considers a randomly

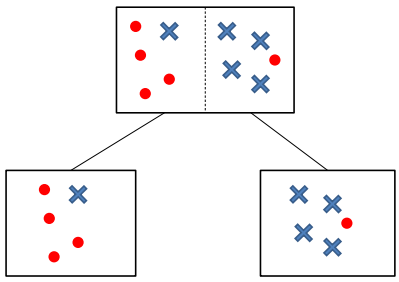


Fig. 3: Example of splitting a node to increase homogeneity.

chosen subset of the available features, which reduces the correlation among the bootstrapped trees and results in a more stable classifier. In addition, the most useful features in terms of prediction are selected by initially fitting the RF to all the features, and then keeping the features with the highest variable importance as described by the tuning procedure in Section IV-C1. Model fitting is carried out using the `randomForest` package in R [24].

1) Tuning the parameters of the RF: RFs require selecting three parameters: the number of trees, n_{tree}, the number of randomly selected features at each split, n_{try}, and the number of features in an initial feature selection step, p_{sel}. The inventors of RF suggest n_{try} to be approximately equal to the square-root of the number of features [20]. Hence, if we consider only $m_1, \ldots, m_5$, then $n_{\text{try}} = 2$, while using all of the features from Table I leads to $n_{\text{try}} = 12$. The number of trees should be large enough so that predictions are made multiple times for each observation [19]. The number of trees (n_{tree}) we considered for tuning were 400, 600, 800, and 1000.

Also, to select an initial subset of features, we use a variable importance measure based on the average (over trees

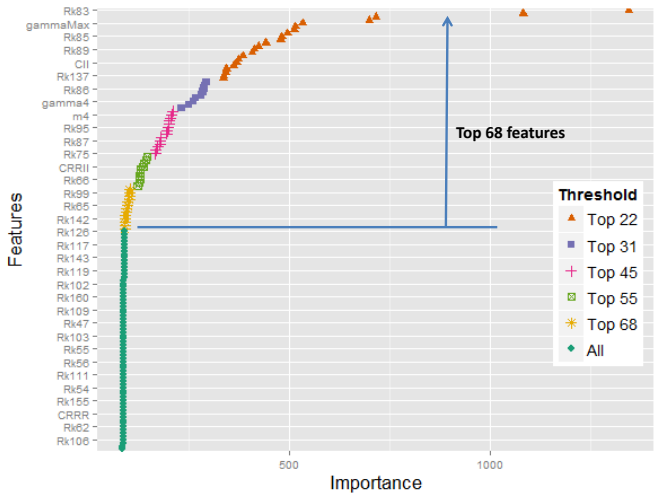


Fig. 4: Variable importance based on the RF using all the features. The number of retained features are indicated in the legend. For example, the blue line and arrow indicate the top 68 features.

in the forest) improvement in homogeneity obtained whenever each feature is used to split a node [20]. This selection step filters out irrelevant features that could lead to overfitting. However, not too many features should be filtered as interactions between features are important. Figure 4 shows the variable importance measure of the available features in our experiment. We considered selecting the top $p_{\text{sel}} = 22$, 31, 55, 68, and 136 (all) features, by finding a reasonable threshold in variable importance. Other reasonable cut-offs could be considered but the difference in performance would likely be minor.

We applied 10-fold cross-validation to each of the number of trees and retained top features using the training set described in Section V. The success rates are averaged over all SNR. Figure 5 summarizes the results of using each combination of n_{tree} and p_{sel}. We found that using 800 trees and keeping the top 68 features yielded the best prediction performance.

2) Prediction: Prediction of a new input signal is made by traversing down each tree in the RF, and the transmission format obtained most frequently from these trees is the prediction made by the RF. Ties are broken uniformly at random.

V. Results and Discussion

In this section, we compare the AMR performance between the FBT classifier from [5] and the proposed CT and RF classifiers, using different feature sets. To this end, we measure the success rate (SR) of each classifier by applying it on a new testing data set that was not used for training, and then recording the proportion of correct predictions obtained by the different classifiers.

We generate a training data set based on $P = 200$ values for each feature per transmission format between SNR from -32 dB to 16 dB in steps of 2 dB. The testing data set is generated similarly to training, except $P = 1100$ is used.

Number of features for initial selection

Number of trees	Top 22	Top 31	Top 45	Top 55	Top 68	All
400	0.6853	0.6797	0.6846	0.6840	0.6829	0.6854
600	0.6853	0.6822	0.6838	0.6848	0.6854	0.6846
800	0.6840	0.6819	0.6848	0.6868	0.6874	0.6856
1000	0.6851	0.6809	0.6865	0.6854	0.6857	0.6858

Cross-validation results

Fig. 5: Cross-validation results from tuning the number of trees (left most column) and features for selection (top row).

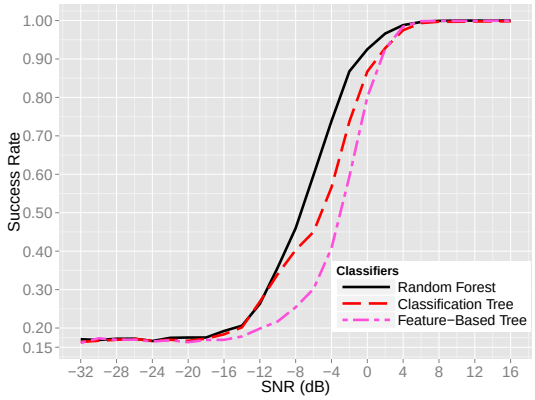


Fig. 6: SR of each classifier using five features from [5] as a function of SNR.

To further highlight the robustness of the proposed classifiers to variations of transmission formats due to frequency errors, we also show results for experiments that include frequency offsets taken uniformly at random from the range of $[-12.5, +12.5]$ kHz applied to the intermediate frequency as described in Section II in both the training and testing data.

A. Experiment One: Applying Features from [5]

In the first experiment, we only use the features from [5] to train each of the classifiers. Figure 6 shows the SR of each classifier. We observe notable gains for both RF and CT over FBT for SNRs ranging from about -15 dB to 1 dB. RF shows the overall best performance with improvements in SR over FBT of, for example, 32% and 13% (in absolute difference) at -5 dB and 0 dB, respectively.

The results for transmission with random frequency offsets are shown in Figure 7. We observe that they are fairly similar to those without offsets in Figure 6 and thus they suggest that the classifiers are robust to such practical impairments.

B. Experiment Two: Applying Additional Features

In the second experiment, we apply RF and CT based on all the features described in Section III and summarized in Table I. The corresponding SR results are shown in Figure 8 along with those of the 5-feature FBT of [5]. Comparing the

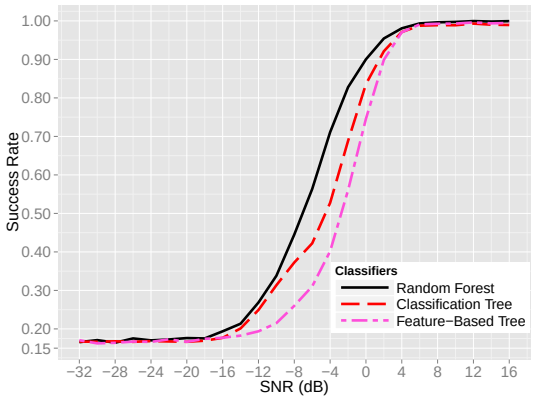


Fig. 7: SR of each classifier using five features from [5] as a function of SNR. Transmissions with random frequency offsets.

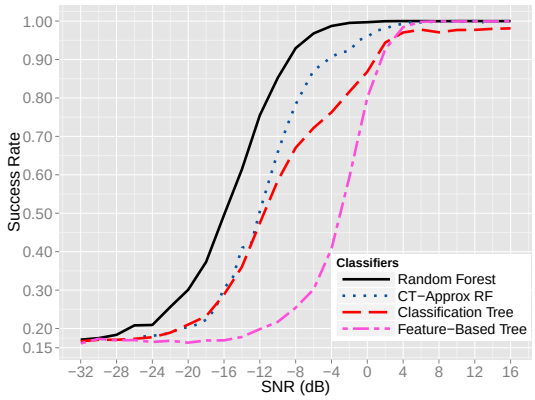


Fig. 8: SR for RF and CT with all features from Table I and for FBT with the five features from [5], as a function of SNR.

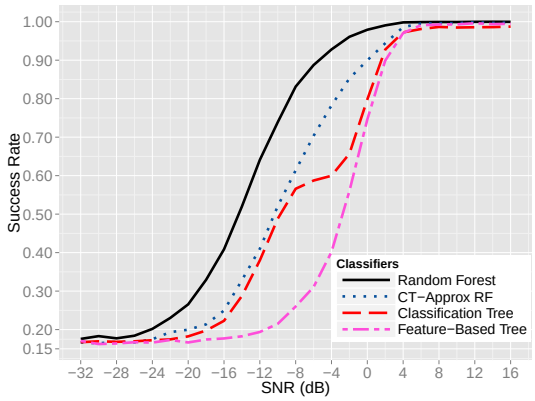


Fig. 9: SR for RF and CT with all features from Table I and for FBT with the five features from [5], as a function of SNR. Transmissions with random frequency offsets.

SR curves for RF and CT from Figures 6 and 8, we note a significant gain due to the inclusion of the additional features. The RF and CT classifiers now outperform the FBT classifier by a large margin on a wide SNR range (from about -25 dB to 2 dB). Furthermore, the RF classifier achieves best SRs, and its advantage over the CT is larger (compare with Figure 6).

As expected, all three classifiers obtain SRs of 1/6 when the SNR is very low (less than -15 dB in Figure 6, less than -27 dB in Figure 8). Also, note that the CT classifier does not reach an SR of one even for very high SNRs. One possible explanation is that its thresholds have been influenced by the more noisier (low SNR) part of the training data. This observation is supported by the notable improvements that CT achieves over FBT at low SNRs ($\lesssim -2$ dB). Finally, note that, also as expected, all classifiers perform almost perfectly well for large SNR levels (SNR $\gtrsim$ 3 dB). The advantage of our proposal is most evident for the more challenging situations of moderate to low SNRs.

The results for the case of random frequency offset are shown in Figure 9. Here we observe some degradation when using the extended set of features compared to the results in Figure 8. The gains compared to FBT are still significant though, which suggests a favorable robustness of the presented classifiers to variations from the nominal transmission frequency.

The SR for detecting each transmission format using RF and all the features from Table I is shown in Figure 10. Note that BFSKB has a higher SR of detection than the other modulations, which is expected considering how BFSKB is modulated at a different intermediate frequency and that the features with the highest importance consist of mostly the magnitude spectrum features as shown in Figure 4. It is also not surprising that BPSK and OQPSK have a similar probability of detection, since both use phase-shift keying modulation. The same applies to BFSKA and BFSKR2.

C. Similarity Between CT Fitted with high SNR Data and FBT

Although the structure of the CT is quite different from the FBT as shown in Figure 2(a), it turns out that fitting the CT with only high SNR (> 1 dB) and the features from [5] results in a tree that is almost equivalent to the FBT. The CT fit with SNR > 1 dB is shown in Figure 11. The structure of the tree at depth two is identical with only minor differences in threshold values. The nodes highlighted in orange emphasize the distinctions where m_5 is chosen before m_4 for the CT. Based on the left subtree after splitting by m_3, small values of m_5 predict OQPSK for both trees. Similarly, small values of m_4 predict BFSK-A and large values of m_5 predict BFSK-B for both trees.

The reason for this similarity can be explained from the way the FBT is constructed as shown in Figure 3 in [5]. The figure shows that each of the thresholds chosen are obtained at SNR≥ 4 except for t_{m_4} obtained at a SNR of approximately -2.5 dB. Indeed, one of the differences in the CT trained with high SNR corresponds to the feature m_4, while the rest of the nodes in the tree are almost equivalent.

D. Relationship Between SNR and Variable Importance

The level of SNR can also affect the variable importance of features obtained from the RF algorithm as shown by Figure 12 which displays the top 15 features in terms of

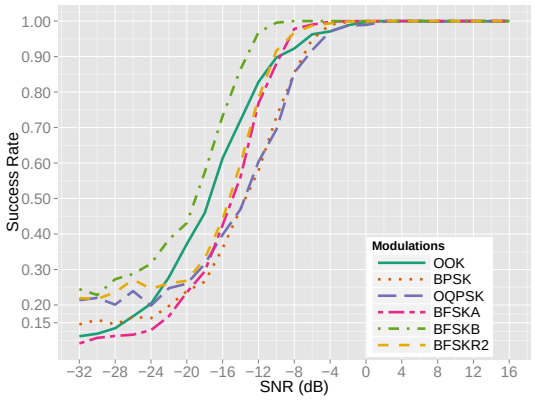


Fig. 10: SR of detection for each transmission format using RF and all the features from Table I.

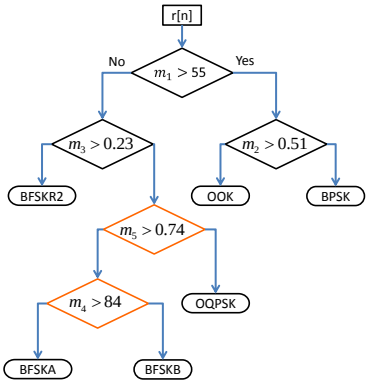


Fig. 11: Flow chart of CT fit at data with high SNR. Distinctions from the FBT in Figure 2(a) are highlighted in orange.

variable importance obtained from three different SNR ranges: full range (-32 to 16 dB), low (< -20 dB), and high (> 1 dB). The top features for the full range SNR are consistent with the plot in Figure 4 although some features are not displayed in the plot to reduce visual clutter. Interestingly, the top 15 features obtained from the full range SNR share many of the same features from both the high (R[83], m_3, R[136], R[84], γ_{max}, m_1, Γ_2, R[73], and m_5) and low SNR ranges (R[83], [R136], [R81], [R85], [R82], and [R77]) which suggests that the RF algorithm is able pick out features that lead to high predictive power under different scenarios such as SNR levels. Furthermore, R[83] has the highest importance regardless of the SNR range used.

E. Complexity Analysis

The price to be paid for the performance gain of a RF is an increase in computation time and storage requirements. Predictions using an RF are typically 2 to 3 orders of magnitude costlier than those obtained from a single tree.

The computational complexity of training a basic implementation of a CT classifier for T training samples and M features is $O(MT\log(T))$ as described on page 199 of [25]. The computational cost of training a RF classifier is

	Full range of SNR	SNR < -20 dB	SNR > 1 dB
1	R[83]	R[83]	R[83]
2	m3	R[136]	m1
3	R[136]	R[107]	gammaMax
4	R[84]	R[81]	m3
5	gammaMax	R[85]	gamma2
6	m1	R[147]	R[136]
7	gamma2	R[110]	CII
8	R[139]	R[77]	m5
9	R[85]	R[125]	R[84]
10	R[81]	R[59]	R[73]
11	R[82]	R[154]	R[132]
12	R[73]	R[82]	CRR
13	R[89]	R[115]	R[93]
14	m5	R[135]	R[132]
15	R[77]	R[150]	CRR

Fig. 12: Top 15 features in terms of variable importance obtained by training the RF at different SNR ranges: full range (-32 to 16 dB), low (< -20 dB), and high (> 1 dB). The variables gammaMax and gamma2 correspond to γ_{max} and Γ_2, respectively.

$O(n_{tree}n_{try}T\log(T))$ since the algorithm involves constructing n_{tree} CTs, and each split in the CTs randomly selects only n_{try} variables instead of all M variables. It is shown in Section IV-C that we use $n_{try} = \sqrt{M}$. Training an FBT classifier only costs $O(T\log(T))$, because the features are selected prior to threshold determination (and sorting a single feature variable takes $O(T\log(T))$ on average). Therefore, the CT and RF classifiers require $O(M)$ and $O(n_{tree}n_{try})$ times more operations to train than an FBT, respectively.

What is more important is the computational cost of prediction, since it is performed every time a signal detection is required. The computation cost for the FBT is approximately $O(1)$ evaluations, since the size of the tree is fixed, and requires at most four comparisons to reach a leaf node (see Figure 2(a)). The CT classifier with pruning is similar to the FBT and requires $O(1)$ operations since the tree is also fixed in terms of depth (see Figure 2(b)). Therefore, both methods are comparable in cost of prediction. However, the CT without pruning requires $O(\log(T))$ operations since the tree is $O(\log(T))$ on average in depth. Thus, a prediction for RF requires $O(n_{tree}\log(T))$ operations.

Since the number of thresholds depend on the number of features used to build the FBT, the storage cost is $O(1)$. Likewise, the CT with pruning often results in a near constant depth tree regardless of the size of the data set. So, the storage cost of CT with pruning is $O(1)$, and therefore the storage requirement of CT is relatively modest. However, the RF requires $O(n_{tree}T)$ in storage requirement as the trees are not pruned.

Since, especially when the modulation recognition classifier is run on an integrated device, a good trade-off between

complexity and performance is paramount, we suggest that after finding the optimal RF with the training set as described above, a CT be computed to approximate the RF predictions on the training set. The predictive performance of such a CT in our experiment is included in Figure 8 (labeled "CT-Approx RF"). We observe that its SR is notably better than that of the pruned CT for a wide range of SNRs, resulting in a much simpler classifier than the RF with competitive results.

VI. Conclusion

In this paper, we apply feature-based classification to AMR in the European 868 MHz band. We studied the application of CT and RF classifiers in some detail and our results show a significant improvement in transmission-format prediction over a previously presented FBT. Furthermore, including additional features in the classifiers notably improves their performance. Including computational and memory considerations, we suggest that a good trade-off between complexity and performance is obtained by approximating the optimal RF with a single CT.

Acknowledgment

The authors would like to thank M. Kuba for helpful comments and making the results from [5] available.

References

[1] M. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," *Signal Processing*, vol. 84, no. 2, pp. 351–365, 2004.

[2] O. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, Apr. 2007.

[3] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic digital modulation classification using genetic programming with K-nearest neighbor," in *IEEE Military Communications Conference (MILCOM)*, Oct. 2010, pp. 1731–1736.

[4] M. Kuba, M. Klatt, K. Ronge, and R. Weigel, "Automatic communication standard recognition in wireless smart home networks," in *IEEE Consumer Communications and Networking Conference (CCNC)*, Jan. 2012, pp. 270–274.

[5] M. Kuba, K. Ronge, and R. Weigel, "Development and implementation of a feature-based automatic classification algorithm for communication standards in the 868 MHz band," in *IEEE Global Communications Conference (GLOBECOM)*, Dec. 2012, pp. 3104–3109.

[6] W. Ben Chikha, I. Dayoub, W. Hamouda, and R. Attia, "Modulation recognition for mimo relaying broadcast channels with direct link," *IEEE Wirel. Commun. Lett.*, vol. 3, no. 1, pp. 50–53, 2014.

[7] K. Hassan, I. Dayoub, W. Hamouda, C. N. Nzéza, and M. Berbineau, "Blind digital modulation identification for spatially-correlated mimo systems," *IEEE Trans. Wireless Commun.*, vol. 11, no. 2, pp. 683–693, 2012.

[8] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, 1998.

[9] M. Valipour, M. Homayounpour, and M. Mehralian, "Automatic digital modulation recognition in presence of noise using SVM and PSO," in *International Symposium on Telecommunications (IST)*, Nov. 2012, pp. 378–382.

[10] N. Ahmadi and R. Berangi, "Modulation classification of QAM and PSK from their constellation using Genetic Algorithm and hierarchical clustering," in *Intl. Conf. on Information and Communication Technologies: From Theory to Applications*, 2008, pp. 1–5.

[11] E. Avci, "Selecting of the optimal feature subset and kernel parameters in digital modulation classification by using hybrid genetic algorithmsupport vector machines: HGASVM," *Expert Systems with Applications*, vol. 36, no. 2, Part 1, pp. 1391 – 1402, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417407005635

[12] C. Dubuc, D. Boudreau, F. Patenaude, and R. Inkol, "An automatic modulation recognition algorithm for spectrum monitoring applications," in *IEEE International Conference on Communications (ICC)*, vol. 1, 1999, pp. 570–574.

[13] M. L. D. Wong, S. K. Ting, and A. K. Nandi, "Naïve bayes classification of adaptive broadband wireless modulation schemes with higher order cumulants," in *International Conference on Signal Processing and Communication Systems (ICSPCS*, Dec. 2008, pp. 1–5.

[14] W. B. Chikha and R. Attia, "On the performance evaluation of bayesian network classifiers in modulation identification for cooperative MIMO systems," in *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, Sep. 2015, pp. 138–142.

[15] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. Cambridge, Mass: MIT Press, 2009.

[16] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA: MIT Press, 2012.

[17] A. Hazza, M. Shoaib, S. Alshebeili, and A. Fahad, "An overview of feature-based methods for digital modulation classification," in *International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Feb. 2013, pp. 1–6.

[18] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. Chapman & Hall / CRC, 1984.

[19] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009, vol. 2, no. 1.

[21] Z. Wu, X. Wang, Z. Gao, and G. Ren, "Automatic digital modulation recognition based on support vector machines," in *International Conference on Neural Networks and Brain (ICNNB)*, vol. 2, Oct. 2005, pp. 1025–1028.

[22] F. Xie, C. Li, and G. Wan, "An efficient and simple method of MPSK modulation classification," in *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, Oct. 2008, pp. 1–3.

[23] T. Therneau, B. Atkinson, and B. Riply, *rpart: Recursive Partitioning and Regression Trees*, 2014, r Package version 4.1-8. [Online]. Available: http://CRAN.R-project.org/package=rpart

[24] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[25] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.