Phase Modulated Radar Waveform Classification Using Quantile One-Class SVMs

Anne M. Pavy  
Sensors Directorate  
Air Force Research Laboratory  
WPAFB, OH 45433  
Email: anne.pavy@us.af.mil

Brian D. Rigling  
Department of Electrical Engineering  
Wright State University  
Dayton, OH 45435  
Email: brian.rigling@wright.edu

Abstract—Radar waveform classification is a difficult problem due to several different varying parameters. The classifier must handle waveform alignment, different pulse widths, and should degrade gracefully with decreasing signal to noise ratios. Along with these tasks, a crowded spectrum makes it highly unlikely that every waveform encountered will be in the waveform library. In this paper, these challenges are addressed through a combination of feature design, training protocol, and classifier approach. The classifier used in this effort is the quantile one-class SVM (q-OCSVM) that has the desirable properties of out-of-class rejection and likelihood estimation. These design choices result in a high performance waveform classifier that addresses the aforementioned challenges as demonstrated with extensive experimentation.

I. INTRODUCTION

Radar waveform classification is an important task in various systems and applications especially for future operations of cognitive radios and radars. With the increasing number of software-defined systems, adapting to the electromagnetic spectrum is necessary for efficient utilization. Waveform classification has several key challenges. Due to the crowded spectrum, it is highly unlikely that every waveform encountered will be in the waveform library. Therefore, the ability to reject waveforms that are not in the library is critical to accurate waveform classification. To take action based on the classification decision, it is important for the classifier to provide a likelihood or confidence in its answer. In addition, the classifier must handle waveform alignment, different pulse widths, and should degrade gracefully with decreasing signal to noise ratios (SNRs). In this effort, these challenges are addressed through a combination of feature design, training protocol, and classifier approach.

To handle waveform alignment issues, features are based on the waveform autocorrelation function. The classifier is trained with waveforms that have various pulse widths and possible SNRs to contend with these variabilities. The classifier used in this effort is the quantile one-class SVM (q-OCSVM) [1] that has the desirable properties of out-of-class rejection and likelihood estimation. These design choices result in a high performance waveform classifier that addresses the aforementioned challenges as demonstrated with extensive experimentation in this paper.

Over the past several years, many publications have addressed the radar waveform classification problem with various types of features, classifiers, and waveforms. In [2], an atomic decomposition-based radar signal detector and modulation classifier was developed using a chirplet dictionary. The modulation types addressed were linear frequency modulation (LFM), frequency shift keying (FSK), and phase shift keying (PSK). In a following study [3], an unmodulated class was added along with features based on short-time Fourier transforms (STFT). Both papers used rule-based classifiers that are derived from the pulse parameters. In [4], various features were used including Wigner and Choi-Williams time-frequency distributions that feed into a multilayer perceptron (MLP) network for radar waveform recognition. The signals detected were put into eight classes: LFM, Costas codes, binary phase, Frank, P1, P2, P3, and P4. The neural network classification structure approximates the posterior probabilities of the output class given the input feature vectors. Calculating the posterior probabilities allows the ability to adapt to changing signal environments and characterizes the correctness of the decision. Although not demonstrated experimentally, they have the ability to reject the classification decision if all the calculated posteriors were below a pre-defined threshold.

In [5], four types of radar modulations are classified: unmodulated, LFM, non-linear frequency modulation (NLFM), and FSK. The authors used a non-conventional feature, Gini’s coefficient, which measures the power inequality in the frequency domain along with other conventional features including the mean and variance of the signal. Waveform recognition was performed using maximum likelihood estimation.

In [6], automatic recognition of radar signals was employed by using image processing methods to extract the time-frequency image shape character. The shape character was classified using support vector machines (SVM) for eight different classes: normal pulse, BPSK, even quadratic frequency modulation (EQFM), Costas, Frank, sinusoidal FM (SFM), triangular linear FM (TLFM) and LFM.

The papers mentioned above use a variety of different feature sets to classify radar modulation types including chirplet dictionaries, Wigner and Choi-Williams time-frequency distributions, Gini’s coefficient, and time-frequency shape character.
Waveform classification is a difficult problem due to nuisance parameters (waveform alignment, varying pulse widths, unknown amplitudes, SNR) so it is important to choose invariant features. In order to handle alignment issues, we use an invariant feature set given in [7]. The feature set used takes the autocorrelation of the pulse and then performs a Fourier transform of the log intensity. This feature set is similar to Mel frequency cepstral coefficients in the speech recognition community. Twenty-three waveforms (Table I) are classified using these invariant features: Barker, Combined Barker, Maximum Length Pseudo Random, Minimum Peak Sidelobe, Polytime (T1, T2, T3), Polyphase Barker, P-codes (P1, P2, P3, P4), and Minimum Shift Keying (some codes are repeated with different code lengths). A Fisher’s Linear Discriminant (FLD) classifier is trained over a range of pulse parameters (waveform alignment, varying pulse widths, pulse repetition ratio (SNR)).

Discrimant (FLD) classifier is trained over a range of pulse parameters (waveform alignment, varying pulse widths, pulse repetition ratio (SNR)). Various classifiers, briefly mentioned, and other clustering methods (fuzzy c-means and fuzzy k-means) have been used as viable classifier options for radar waveform recognition. This paper outlines an approach to classify phase-coded radar waveforms using quantile one-class support vector machines (q-OCSVM) [1]. The q-OCSVM framework provides performance guarantees that prevent overfitting with small amounts of training data, has the ability to recognize that new waveforms are not in the library, and gives a confidence value on the classification decision.

The remainder of this paper is organized as follows. In Section II, the received signal model and ACF-based feature extraction is described. Section III discusses the onen-class SVM and terminology to help introduce the quantile one-class SVM (q-OCSVM) in Section IV. Section V presents simulation results for several different phase modulation classes as a function of SNR and the number of pulses observed. This section also includes a leave-one-out method to test the classifier against unknown phase modulations. Finally, Section VI concludes with a summary and future work.

II. ACF-BASED FEATURES

As mentioned above, in [7], the authors developed a feature set that is invariant to unknown signal parameters, including complex-valued amplitude ($A_p$), frequency offset ($\omega_0$), timing offset ($t_p$), and time scaling ($\eta$) to determine the phase modulation of a pulse. This paper uses the same feature set. The received signal model and feature extraction are summarized in this section.

A radar system emits a continuous-time prototype pulse

$$l_c(t) = \begin{cases} \exp\{j\phi_c(t)\}, & t \in [-\frac{T}{2}, \frac{T}{2}] \\ 0, & \text{otherwise} \end{cases}$$

(1)

with a phase modulation $\phi_c(t)$ belonging to one of the $C$ classes enumerated in Table I and where $T$ is the pulse width. The emitter generates a chain of $P$ pulses with pulse repetition interval $T$ that are observed by the receiver where the pulse processor performs envelope detection and thresholding. This allows the pulses to be segmented from the chain into a stack of independent pulse observations,

$$h_p(t) = A_p l_c(t/\eta - t_p) e^{j\omega_0 t} + z_p(t), \quad p = 0 \ldots P - 1$$

(2)

where $z_p(t)$ is white Gaussian noise with variance $\sigma_{z_p}^2$ and $t_p = pT$.

To discuss the feature extraction process, the received signal model will be represented in discrete-time, $h_p[u] = h_p(uT_s)$ for $u = 0 \ldots U - 1$ (samples in pulse) where $T_s$ is the sampling interval. For each input pulse, the auto-correlation function (ACF) is computed by

$$r_p[b] = \sum_{u=0}^{U-1} h_p[u] h_p^*[u - b].$$

(3)

With this step, the time offset $t_p$ is suppressed due to ACF invariance. To help suppress the noise, the processor then computes the ensemble average ACF for $P$ pulses which is represented by $r[b]$. Following the pulse averaging, the discrete-time Fourier transform of the log-magnitude is taken. This can be quickly computed using a FFT given by

$$x[m] = \sum_{u=-U+1}^{U+1} \log |r[b]| e^{-j2\pi mb/(2U-1)}, \quad m = 1 \ldots d$$

(4)

where the resultant coefficients capture the principal side lobe structure of the log-ACF.

The sequence of features $x[m]$ is used to identify the phase modulation used in the prototype pulse in (1). The coefficients are collected in an $n \times d$ matrix

$$X = [x_1, x_2, \ldots, x_d]$$

(5)

where $d$ is the number of spectral coefficients and $x_m$ is a $n \times 1$ vector of the $m$th spectral coefficient where $n$ is the number of . This coefficient set may now be used for waveform identification. For more detailed information on the invariance of this feature set, refer to [7].

III. SUPPORT VECTOR MACHINES

The Support Vector machine (SVM) was developed by Vapnik and his group in 1965 with the first main paper in 1995 [8]. SVMs aim to find a surface that maximizes the margin between classes, defining an optimal margin classifier. From statistical learning theory perspective, the SVM is based on the principle of structural risk minimization which aims at minimizing a bound on the generalization error of a model [9]. This bound is a desired property as it provides guarantees to prevent overtraining, which can be an issue with small amounts of training data which are far more likely in complex environments.

The SVM has consistently been one of the highest performing classifiers and has, in fact, been used as the backend classifier for the “deep learning” architectures which are the current focus of universities and industry (e.g., Google, Microsoft, Facebook) [10]. In addition, there has been a continuing focus on SVMs to advance their flexibility and performance. Many of these recent SVM advances make the SVM framework an excellent structure to meet the goals stated above.
A. One-class SVM approach to classify anomalous behavior

Consider a complex environment where new waveforms or new modes of existing systems can appear for which no previous training data is available. In this scenario, it is too difficult to characterize features under the alternative hypothesis. In such an environment, we propose to use one-class SVMs to both classify known waveforms and to detect anomalies for those classes where no previous training data exists.

In the one-class SVM classification problem, training points \( \{ x_i \}_{i=1}^n \) where \( x_i \in \mathbb{R}^d \) are separated from the origin in feature space and the distance from this hyperplane to the origin is maximized. In the popular \( \nu \)-SVM formulation [11], the algorithm finds a function \( g \) that returns +1 in a region capturing “most” of the data points, and −1 elsewhere. Each \( x_i \) is transformed via a map \( \Phi : \mathbb{R}^d \to \mathcal{F} \) where \( \mathcal{F} \) is a high dimensional Hilbert space generated by a positive-definite kernel \( k(x, x') \). The kernel function represents an inner product in \( \mathcal{F} \) through \( k(x, x') = \langle \Phi(x), \Phi(x') \rangle \). If the data is non-separable, slack variables \( \xi_i \) allow for some points to be within the margin and the parameter \( \nu \in [0, 1] \) is the regularization parameter that sets an upper bound on the fraction of these margin errors. The \( \nu \)-SVM formulation is solved using the following optimization:

\[
\min_{w, \xi, \rho} \frac{1}{2} \| w \|^2 - \rho \sum_{i=1}^n \xi_i \quad \text{s.t.} \quad \langle w, \Phi(x_i) \rangle \geq \rho - \xi_i, \; \xi_i \geq 0, \; i \in [n],
\]  

(6)

where \( \rho \in [0, 1] \) controls the width of the margin, which is equal to \( \frac{2\rho}{\|w\|^2} \). The function \( g(x) = \text{sgn}(\langle w, \Phi(x') \rangle - \rho) \) determines whether a point is in the positive or negative set.

IV. QUANTILE ONE-CLASS SVM

In [1], the authors formulated a one-class SVM extension where \( q \) quantiles (approximated by minimum-value sets) of a high-dimensional distribution are estimated. Their formulation, they introduced additional notation for minimum-value sets in a hypothesis space.

Equation (6) is extended by introducing a hypothesis space \( \mathcal{H} \) of half-space decision functions \( f_C(x) = \text{sgn}(\langle w, \Phi(x') \rangle - \rho) \) such that \( f_C(x) = 1 \) if \( x \in C \), and −1 otherwise, where \( C \) corresponds to the estimated minimum value set. The one-class SVM returns a function \( f_C \in \mathcal{H} \) that maximizes the margin between the half-space decision boundary and the origin in \( \mathcal{F} \) and bounds a portion (given by \( \nu \)) of examples satisfying \( f_C(x) = -1 \). By solving (6) with the extended hypothesis space \( \mathcal{H} \) described for \( \nu = 1 - \alpha \), the one-class SVM can be used to approximate \( C(\alpha) \).

In [1], a sequence of quantiles is introduced, \( 0 < \alpha_1 < \alpha_2, \cdots < \alpha_q < 1 \), where \( q \) is the number of minimum-value sets to approximate. The q-OCSVM algorithm generalizes Equation (6) by approximating a set of minimum-value sets \( \{ C_1, \ldots, C_q \} \) so that the hierarchy constraint \( C_i \subseteq C_j \) is satisfied for \( i < j \). The q-OCSVM algorithm [1] solves the following primal problem:

\[
\min_{w, \xi, \rho} \frac{q}{2} \| w \|^2 - \sum_{j=1}^{q} \rho_j + \frac{1}{\nu_j n} \sum_{i} \xi_{j,i}
\]

(7)

\[
\text{s.t.} \quad \langle w, \Phi(x_i) \rangle \geq \rho_j - \xi_{j,i}, \; \xi_{j,i} \geq 0, \; j \in [q], \; i \in [n]
\]

where \( \nu_j = 1 - \alpha_j \). The program finds multiple, parallel half-space decision functions by searching for a global minimum over the sum of \( q \) objective functions while the programs share the same \( w \). The \( q \) half-spaces in the solution are only different by their bias terms which makes them parallel to each other.

The more efficient q-OCSVM dual program is

\[
\min_{\lambda} \frac{1}{2q} \sum_{j=1}^{q} \sum_{i,s} \lambda_{j,i} \lambda_{p,s} k(x_i, x_s)
\]

(8)

\[
\text{s.t.} \quad 0 \leq \lambda_{j,i} \leq \frac{1}{\nu_j n} \sum_{i} \lambda_{j,i} = 1, \; i \in [n], \; s \in [q]
\]

where the dot product \( \langle \Phi(x_i), \Phi(x_s) \rangle \) is replaced with the Gaussian kernel

\[
k(x_i, x_s) = e^{-\gamma \|x_i-x_s\|^2}
\]

(9)

and \( \lambda_{j,i} \) are Lagrangian multipliers. The resulting decision function for the \( j \)th estimate is

\[
f_{C_j}(x') = \text{sgn} \left( g_{C_j}(x') \right)
\]

(10)

where

\[
g_{C_j}(x') = \frac{1}{q} \sum_{i} \sum_{j=1}^{q} \lambda_{j,i} k(x_i, x_s) - \rho_j.
\]

(11)

The values for \( \rho_j \), using the condition \( 1 < \lambda_{j,i} < \frac{1}{\nu_j} \), are found from a point \( x_{sv} \) by

\[
\rho_j = \langle w, \Phi(x_{sv}) \rangle = \frac{1}{q} \sum_{i} \sum_{j=1}^{q} \lambda_{j,i} k(x_i, x_{sv}).
\]

(12)

This formulation sets the stage for a probabilistic classifier that provides confidence and detects outliers. This approach is in contrast to the standard SVM which is non-probabilistic and discriminative. Hence, this classifier has three of the desirable properties mentioned – prevention of overtraining, detection of outliers, and prediction of probability of classification or confidence.

V. SIMULATION RESULTS

The training data, \( X \), in (5) was generated for every combination of class (c), number of pulses averaged (\( P \)), and SNR range (\( \beta \)). In order to clarify which training set is being examined, specific training data will be referred to as \( X^{c,P,\beta} \) a \( n \times d \) matrix where \( n = 1000 \) samples and \( d = 10 \) spectral components.

In the following simulations, 11 different over-lapping SNR ranges are used to train the classifier: \( \beta \in \{-12, -8, -10, -6, -8, -4, -6, -2, 0, 4, 2, 6, 4, 8, 6, 10, 8, 12\} \) dB (\( \beta = 1 \) refers to \( [-12, -8] \), \( \beta = 2 \) refers to \( [-10, -6] \), and so on). For each SNR range and each
pulse averaging combination, 1000 signals are simulated for each of the 23 modulation classes in Table I. Each SNR range is sampled uniformly within its range and the pulse widths are sampled discretely (pulse width ranges for each class are listed in Table I) from the set \( P \in \{ 1, 5, 10, 20 \} \). Hence, 44 \((11 \times 4)\) q-OCSVM models are trained for each class.

Each q-OCSVM classifier is trained in two ways. The difference was in the choice of the outer quantile. For the first method \( \alpha_j = [0.8, 0.6, 0.4, 0.2, 0.05] \) and for the second method \( \alpha_j = [0.8, 0.6, 0.4, 0.2, 0] \). Note that the difference between the two is the outer boundary where the first method’s outer boundary contains approximately 95% of the training data and the second method’s outer boundary contains approximately 100% of the training data. Hence, the innermost quantile contains roughly 20 percent of the training data, the next quantile contains 40 percent, then 60, 80, and (95 or 100). An example of a Barker 7 moduation (in the Barker 7 pulse width range), \([-12, -8]\), SNR range, and \( P=1 \) pulse averaged q-OCSVM model with its corresponding training points is shown in Figure 1. The first two spectral components are the shown features where the darkest grayscale corresponds to the highest probability. In Figures 1a and 1b, models are shown using \( \alpha_5 = 0.05 \) and \( \alpha_5 = 0 \), respectively, which clearly impacts inclusion of outliers.

<table>
<thead>
<tr>
<th>Type</th>
<th>Code Length</th>
<th>Training ( \tau (\mu\text{s}) )</th>
<th>Testing ( \tau (\mu\text{s}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barker</td>
<td>7 [0.875, 7]</td>
<td>1.75</td>
</tr>
<tr>
<td>2</td>
<td>Barker</td>
<td>11 [1.375, 11]</td>
<td>2.75</td>
</tr>
<tr>
<td>3</td>
<td>Barker</td>
<td>13 [1.625, 13]</td>
<td>3.25</td>
</tr>
<tr>
<td>4</td>
<td>Combined Barker</td>
<td>16 [1, 8]</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Combined Barker</td>
<td>49 [3.08, 21.1]</td>
<td>6.13</td>
</tr>
<tr>
<td>6</td>
<td>Combined Barker</td>
<td>169 [10.58, 84.6]</td>
<td>21.1</td>
</tr>
<tr>
<td>7</td>
<td>Max. Length Pseudo Random</td>
<td>15 [1, 3.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>8</td>
<td>Max. Length Pseudo Random</td>
<td>31 [0.235, 10.5]</td>
<td>1.5</td>
</tr>
<tr>
<td>9</td>
<td>Max. Length Pseudo Random</td>
<td>63 [4.221, 18.9]</td>
<td>6.3</td>
</tr>
<tr>
<td>10</td>
<td>Min. Peak sidelobe</td>
<td>7 [1.05, 4.2]</td>
<td>1.4</td>
</tr>
<tr>
<td>11</td>
<td>Min. Peak sidelobe</td>
<td>25 [1.25, 10]</td>
<td>2.5</td>
</tr>
<tr>
<td>12</td>
<td>Min. Peak sidelobe</td>
<td>48 [2.4, 19.2]</td>
<td>4.8</td>
</tr>
<tr>
<td>13</td>
<td>T1</td>
<td>NA [2, 16]</td>
<td>4</td>
</tr>
<tr>
<td>14</td>
<td>T2</td>
<td>NA [1.5, 12]</td>
<td>2</td>
</tr>
<tr>
<td>15</td>
<td>T3</td>
<td>NA [1, 8]</td>
<td>1</td>
</tr>
<tr>
<td>16</td>
<td>Polyphase Barker</td>
<td>7 [0.875, 7]</td>
<td>1.75</td>
</tr>
<tr>
<td>17</td>
<td>Polyphase Barker</td>
<td>20 [1, 8]</td>
<td>2</td>
</tr>
<tr>
<td>18</td>
<td>Polyphase Barker</td>
<td>40 [2, 16]</td>
<td>4</td>
</tr>
<tr>
<td>19</td>
<td>P1</td>
<td>NA [5, 20]</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>P2</td>
<td>NA [3.2, 25.6]</td>
<td>6.4</td>
</tr>
<tr>
<td>21</td>
<td>P3</td>
<td>NA [5.2, 25.6]</td>
<td>6.4</td>
</tr>
<tr>
<td>22</td>
<td>P4</td>
<td>NA [5, 20]</td>
<td>10</td>
</tr>
<tr>
<td>23</td>
<td>Minimum Shift Key</td>
<td>63 [2, 18.9]</td>
<td>4</td>
</tr>
</tbody>
</table>

The accuracy of the model for likelihood prediction is verified by generating new data consistent with the training data but with different noise and phase realizations. For each scenario trained (different SNR ranges and consistent of pulses averaged), an accompanying model accuracy figure is shown (Figures 3a and 4a). Each bin contained the predicted number of realizations averaged over all waveform models.

Testing data is generated using the testing pulse width for each class shown in Table I at specific SNRs and \( P \) values shown in Tables II and III. Each testing signal is classified by 253 \((23 \times 11)\) classifiers. Each classifier consists of \( q = 5 \) hierarchical quantiles that give a natural probability distribution. For example, if a test point falls within the first quantile (the innermost quantile), the likelihood that the test point belongs to that class is \( \alpha_1 = 0.8 \). The class chosen is either the unknown class which is outside the boundaries for all classifiers or the classifier that provides the highest likelihood score. The likelihood score is computed at the quantile level unless the signal lies in the same quantile for two or more classes. In such a case, a normalized distance metric is computed to resolve the ties at the quantile level. The normalized distance metric is computed as follows:

\[
D(x') = \frac{g_{C_j}(x_{svj}) - g_{C_j}(x')}{{g_{C_j}(x_{svj}) - g_{C_{j-1}}(x_{svj-1})}}
\]

where \( g_{C_j}(\cdot) \) is computed using Equation (11) and \( x_{svj} \) is
the point corresponding to the support vector on \( j \)th quantile boundary. This metric effectively interpolates with a quantile region to more accurately compare distance values. The class with the smallest normalized distance is chosen. This results in 11 class decisions from each SNR range bank. These 11 decisions are compared in the same way for a final class decision.

This comprehensive training structure develops a classifier that detects unknown pulse modulations as anomalies and accurately characterizes the waveform’s support in feature space. Each confusion matrix in Figures 3 and 4 consists of 23 different matrices combined by rotating the unknown class by leaving one out. The x-axis is the estimated class and the y-axis is the true class. Figures 3b and 3c show the 24 class problem (23 known and 1 unknown) for SNR at -10 dB, outer quantile boundary of \( \alpha_j = 0.05 \), and \( P = 1 \) and \( P = 20 \), respectively. Figures 4b and 4c show the 24 class problem for SNR at -10 dB, outer quantile boundary of \( \alpha_j = 0 \), and \( P = 1 \) and \( P = 20 \), respectively. As expected, the performance degrades for lower signal to noise ratios. However, the features provide sufficient separability such that excellent performance is achievable at low SNRs with \( P = 20 \). In Figure 3c, the P3 modulation (\( c = 21 \)) is confused with the unknown class. In Figure 2, data points for the first two spectral components for all SNR ranges are plotted for classes Barker 13 and P3 modulations for comparison. The P3 modulation data plot holds a unique shape, and it likely requires more samples to fully define its boundaries (especially in 10 dimensions).

In Tables II-V, the probability of correct classification for 24 different modulation types (23 known and 1 unknown class), denoted by \( P_{cc}^{A} \) for several \( \alpha_5 \) and \( P \) values is shown for \( \alpha_5 = 0.05 \) and \( \alpha_5 = 0 \), respectively. As expected, when the SNR and pulse averaging increased so did the performance. When the outer boundary of the models included roughly 95% of the data, the overall performance was worse compared to when the boundary was pushed out to include roughly all of the data. In Tables IV and V, the probability of correct classification for the unknown class, denoted by \( P_{cc}^{U} \) is shown for the same parameters. In this case, when the boundary included 95% of the data, the unknown class performance was better than the 100% boundary. The results show that the outer boundary is very important as it must balance overall classification performance versus unknown class sensitivity.

### Table II

<table>
<thead>
<tr>
<th>SNR (dB) ( P )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>0.3666</td>
<td>0.7901</td>
<td>0.8756</td>
<td>0.9082</td>
</tr>
<tr>
<td>-8</td>
<td>0.5409</td>
<td>0.8464</td>
<td>0.8801</td>
<td>0.9179</td>
</tr>
<tr>
<td>-6</td>
<td>0.5988</td>
<td>0.8575</td>
<td>0.8891</td>
<td>0.9117</td>
</tr>
<tr>
<td>-4</td>
<td>0.6813</td>
<td>0.8668</td>
<td>0.8970</td>
<td>0.9190</td>
</tr>
<tr>
<td>-2</td>
<td>0.7325</td>
<td>0.8675</td>
<td>0.8994</td>
<td>0.9154</td>
</tr>
<tr>
<td>0</td>
<td>0.7640</td>
<td>0.8808</td>
<td>0.9035</td>
<td>0.9164</td>
</tr>
<tr>
<td>2</td>
<td>0.7872</td>
<td>0.8943</td>
<td>0.9160</td>
<td>0.9198</td>
</tr>
<tr>
<td>4</td>
<td>0.8152</td>
<td>0.9092</td>
<td>0.9245</td>
<td>0.9295</td>
</tr>
<tr>
<td>6</td>
<td>0.8398</td>
<td>0.9249</td>
<td>0.9343</td>
<td>0.9393</td>
</tr>
<tr>
<td>8</td>
<td>0.8670</td>
<td>0.9432</td>
<td>0.9480</td>
<td>0.9440</td>
</tr>
<tr>
<td>10</td>
<td>0.8834</td>
<td>0.9575</td>
<td>0.9498</td>
<td>0.9440</td>
</tr>
</tbody>
</table>

### Table III

<table>
<thead>
<tr>
<th>SNR (dB) ( P )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>0.3595</td>
<td>0.7899</td>
<td>0.9038</td>
<td>0.9403</td>
</tr>
<tr>
<td>-8</td>
<td>0.4742</td>
<td>0.8495</td>
<td>0.9193</td>
<td>0.9509</td>
</tr>
<tr>
<td>-6</td>
<td>0.6092</td>
<td>0.8923</td>
<td>0.9274</td>
<td>0.9612</td>
</tr>
<tr>
<td>-4</td>
<td>0.9074</td>
<td>0.9604</td>
<td>0.9847</td>
<td>0.9683</td>
</tr>
<tr>
<td>-2</td>
<td>0.7598</td>
<td>0.9111</td>
<td>0.9457</td>
<td>0.9677</td>
</tr>
<tr>
<td>0</td>
<td>0.7981</td>
<td>0.9243</td>
<td>0.9478</td>
<td>0.9685</td>
</tr>
<tr>
<td>2</td>
<td>0.8239</td>
<td>0.9345</td>
<td>0.9583</td>
<td>0.9733</td>
</tr>
<tr>
<td>4</td>
<td>0.8443</td>
<td>0.9446</td>
<td>0.9655</td>
<td>0.9740</td>
</tr>
<tr>
<td>6</td>
<td>0.8630</td>
<td>0.9549</td>
<td>0.9720</td>
<td>0.9759</td>
</tr>
<tr>
<td>8</td>
<td>0.8804</td>
<td>0.9665</td>
<td>0.9771</td>
<td>0.9770</td>
</tr>
<tr>
<td>10</td>
<td>0.8903</td>
<td>0.9747</td>
<td>0.9787</td>
<td>0.9779</td>
</tr>
</tbody>
</table>

### Table IV

<table>
<thead>
<tr>
<th>SNR (dB) ( P )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>0.0124</td>
<td>0.3436</td>
<td>0.3526</td>
<td>0.4496</td>
</tr>
<tr>
<td>-8</td>
<td>0.0532</td>
<td>0.5047</td>
<td>0.6808</td>
<td>0.8254</td>
</tr>
<tr>
<td>-6</td>
<td>0.1420</td>
<td>0.5900</td>
<td>0.7592</td>
<td>0.8485</td>
</tr>
<tr>
<td>-4</td>
<td>0.3218</td>
<td>0.6686</td>
<td>0.7330</td>
<td>0.8787</td>
</tr>
<tr>
<td>-2</td>
<td>0.4377</td>
<td>0.7127</td>
<td>0.8010</td>
<td>0.9171</td>
</tr>
<tr>
<td>0</td>
<td>0.5081</td>
<td>0.7277</td>
<td>0.8318</td>
<td>0.9164</td>
</tr>
<tr>
<td>2</td>
<td>0.5547</td>
<td>0.7878</td>
<td>0.8469</td>
<td>0.9106</td>
</tr>
<tr>
<td>4</td>
<td>0.5826</td>
<td>0.8044</td>
<td>0.8559</td>
<td>0.9248</td>
</tr>
<tr>
<td>6</td>
<td>0.5959</td>
<td>0.8102</td>
<td>0.8641</td>
<td>0.9559</td>
</tr>
<tr>
<td>8</td>
<td>0.6063</td>
<td>0.8142</td>
<td>0.8803</td>
<td>0.9658</td>
</tr>
<tr>
<td>10</td>
<td>0.6118</td>
<td>0.8181</td>
<td>0.8940</td>
<td>0.9746</td>
</tr>
</tbody>
</table>

### Table V

<table>
<thead>
<tr>
<th>SNR (dB) ( P )</th>
<th>1</th>
<th>5</th>
<th>10</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>-10</td>
<td>0.0005</td>
<td>0.1028</td>
<td>0.3881</td>
<td>0.5688</td>
</tr>
<tr>
<td>-8</td>
<td>0.0111</td>
<td>0.3248</td>
<td>0.4763</td>
<td>0.6795</td>
</tr>
<tr>
<td>-6</td>
<td>0.0491</td>
<td>0.4390</td>
<td>0.5615</td>
<td>0.7448</td>
</tr>
<tr>
<td>-4</td>
<td>0.1499</td>
<td>0.5003</td>
<td>0.6325</td>
<td>0.7496</td>
</tr>
<tr>
<td>-2</td>
<td>0.3006</td>
<td>0.5877</td>
<td>0.6850</td>
<td>0.7736</td>
</tr>
<tr>
<td>0</td>
<td>0.3896</td>
<td>0.6664</td>
<td>0.7272</td>
<td>0.7994</td>
</tr>
<tr>
<td>2</td>
<td>0.4773</td>
<td>0.7313</td>
<td>0.7612</td>
<td>0.8139</td>
</tr>
<tr>
<td>4</td>
<td>0.6679</td>
<td>0.8244</td>
<td>0.8173</td>
<td>0.8327</td>
</tr>
<tr>
<td>6</td>
<td>0.8403</td>
<td>0.8777</td>
<td>0.8660</td>
<td>0.8554</td>
</tr>
<tr>
<td>8</td>
<td>0.8871</td>
<td>0.8811</td>
<td>0.9235</td>
<td>0.8733</td>
</tr>
<tr>
<td>10</td>
<td>0.9480</td>
<td>0.8887</td>
<td>0.8109</td>
<td>0.8734</td>
</tr>
</tbody>
</table>

Fig. 2. Class comparison
VI. CONCLUSION

Waveform recognition is a difficult problem with many variables including time alignment, varying pulse widths and amplitudes, and numerous types of waveforms. In this paper, we chose features computed from the autocorrelation function to be invariant to alignment issues and trained over multiple pulse widths and SNRs to handle the additional variability. We were able to reject waveforms that were not in the library by using the q-OCSVM classification framework and were able to control the out of library rejection rate by selecting the desired value for the outer classification boundary. This framework also provided a confidence or likelihood for each classification decision conditioned on each waveform model. In addition, since each waveform is trained separately, the addition of new waveforms to the library does not require the retraining of the original waveforms. These properties make this classification framework an attractive framework for building an adaptive waveform classification approach. Future work will pursue online learning of the out-of-library classes in order to adapt to and classify new waveforms by building models and learning these new classes on-the-fly.

REFERENCES