

On the Application of Neural Networks to the Classification of Phase Modulated Waveforms

Anthony Buchenroth^a, Joong Gon Yim^b, Michael Nowak^a, and Vasu Chakravarthy^a

^aAir Force Research Laboratory, Wright-Patterson AFB, Ohio

^bBooz Allen Hamilton, Dayton, Ohio

ABSTRACT

Accurate classification of phase modulated radar waveforms is a well-known problem in spectrum sensing. Identification of such waveforms aids situational awareness enabling radar and communications spectrum sharing. While various feature extraction and engineering approaches have sought to address this problem, the use of a machine learning algorithm that best utilizes these features is becomes foremost. In this effort, a comparison of a standard shallow and a deep learning approach are explored. Experiments provide insights into classifier architecture, training procedure, and performance.

Keywords: neural network, waveform classification, spectrum sensing

1. INTRODUCTION

Dynamic spectrum access (DSA), spectrum sharing, and radio frequency (RF) convergence are concepts that have been receiving ever-increasing attention. As technology has advanced, more and more devices are requiring access to the RF spectrum. The proliferation of various spectrum users and the increasing bandwidth requirements of those spectrum users has driven many to consider the RF spectrum as a commodity. As scarcity for this commodity increases in some frequency bands, the need to more effectively and efficiently use the RF spectrum has impacted both policy and technology.

Recent research in waveform design has focused on facilitating methods for radar and communications spectrum sharing and optimizing performance in multifunction radar. Specifically, radar waveform design has focused on optimizing phase coded waveforms to attain a tactical goal. Furthermore, radar systems are increasingly equipped with waveform management algorithms that employ many dissimilar waveforms to be emitted.

This broad diversity of waveforms in the RF environment poses a challenge for spectrum sensing devices that provide situational awareness to RF spectrum users. Previous works have focused on feature engineering and machine learning for waveform recognition.¹²³⁴ While some works¹²³ focus mainly on feature engineering, other work⁴ places more emphasis on developing a machine learning approach to both classify within-library modulation and detect out-of-library modulations. Similarly, many communities including speech recognition, image classification, and computer vision have begun to place more emphasis on the application of more sophisticated machine learning. To that end, many classification applications have focused on applying neural network algorithms.⁵⁶³

This paper leverages existing feature engineering work¹ to investigate the applicability of sophisticated neural network approaches to the recognition of phase modulated waveforms. We employ the use of features extracted from the ambiguity function (AF) of an intercepted waveform that have shown to aid in accurate classification. This effort will provide a performance comparison between the classifiers in literature¹ and a single hidden layer neural network (SHLNN) classifier, identify fundamental differences in training procedures, and lead to the robust application of a convolutional neural network (CNN) classifier.

Further author information: (Send correspondence to Anthony Buchenroth)
E-mail: anthony.buchenroth@us.af.mil, Telephone: 937-713-4022

The remainder of this paper is organized as follows. Section 2 will briefly discuss the waveform feature extraction process. Section 3 describes the application and results of SHLNN classification methods. Section 4 presents the application and results of CNN classification methods. Finally, Section 5 concludes with a summary of findings and future efforts.

2. FEATURE EXTRACTION

Given that a radar emits a continuous-time prototype pulse

$$x_c(t) = \begin{cases} \exp\{j\phi_c(t)\}, & t \in [-\tau/2, \tau/2] \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\phi_c(t)$ is a phase modulation belonging to one of C classes in Table 1. As described,¹ the intercept receiver observes the signal

$$y_p(t) = A_p x_c(t/\alpha - t_p) e^{j\omega_0 t} + v_p(t), \quad p = 0 \dots P - 1, \quad (2)$$

corrupted by white Gaussian noise $v_p(t)$ with variance σ_v^2 and potentially degraded by model mismatch in the receiver. Further, the observed pulses are altered by jitter in complex amplitude A_p , time scaled by α to achieve the designed pulse width, and modulated to carrier frequency, ω_0 ; all of which are unknown to the intercept receiver. Additionally, each pulse from the radar is contained within a window slightly larger than the pulse width to allow for errors in edge detection. The offset within the window, t_p , is assumed to be unknown. Radar pulse observations are then input into the phase modulated waveform classifier in digitized form, $y_p[n] = y_p(nT_s)$, $n = 0 \dots N - 1$, where T_s is the sampling interval assumed to satisfy the Nyquist requirement.

Table 1. Phase modulation types.¹

c	Modulation Type	Code Length	Training τ (μsec)	Testing τ (μsec)
1	Barker	7	1.75	7.0
2	Barker	11	2.75	11.0
3	Barker	13	3.25	13.0
4	Combined Barker	16	2.0	8.0
5	Combined Barker	49	6.13	2.11
6	Combined Barker	169	22.1	84.6
7	Maximum Length Pseudo Random	16	1.5	4.5
8	Maximum Length Pseudo Random	64	3.5	10.5
9	Maximum Length Pseudo Random	256	6.3	18.9
10	Minimum Peak Sidelobe	10	1.4	4.2
11	Minimum Peak Sidelobe	25	2.5	10.0
12	Minimum Peak Sidelobe	48	4.8	19.2
13	T1	NA	4.0	16.0
14	T2	NA	3.0	12.0
15	T3	NA	8.0	2.0
16	Polyphase Barker	7	1.75	7.0
17	Polyphase Barker	20	2.0	8.0
18	Polyphase Barker	40	4.0	16.0
19	P1	NA	10.0	20.0
20	P2	NA	6.4	25.6
21	P3	NA	6.4	25.6
22	P4	NA	10.0	29.0
23	Minimum Shift Key	64	18.9	8.0

Because the focus of this paper is on the application and analysis of neural network classifiers to the waveform classification problem, for brevity, we will not overly emphasize the feature extraction process. As such, we compute the two-dimensional discrete Fourier transform of the log-scaled ambiguity function given by

$$L_{\hat{\Phi}}[u, v] = \frac{1}{MK} \sum_{m=0}^{2M-1} \sum_{k=0}^{2K-1} \left[\log \left| \hat{\Phi}_c[m\Delta t, k\Delta f] \right| + \log \frac{|A|^2}{\hat{\Phi}_c[0, 0]} + z[m, k] \right] e^{-j2\pi \left(\frac{u}{2M-1} m + \frac{v}{2K-1} k \right)}, \quad (3)$$

$$u \in \left[\frac{-1}{2\Delta t}, \frac{1}{2\Delta t} \right], \quad v \in \left[\frac{-1}{2\Delta f}, \frac{1}{2\Delta f} \right].$$

For a more detailed and complete description of the feature extraction process, refer to the ambiguity-based classification algorithm.¹

3. SINGLE HIDDEN LAYER NEURAL NETWORK CLASSIFICATION

In keeping with the classification strategy in the ambiguity-based classification algorithm,¹ we leverage the classifier training methodology and apply the use of a single hidden layer neural network. Specifically, we train our classifier with the 23 phase modulations displayed in Table 1. For each class, we perform a 1000 iteration Monte Carlo simulation at 10 dB SNR with the pulse widths denoted in Table 1. In each iteration, pseudo-random noise and phase realizations are imparted on the waveform. Additionally, the time offset within the pulse capture window and sampling rate is randomized. This training procedure allows for an intact, yet well-generalized waveform data set to be input into our feature extractor in an effort to safeguard against biases and overfitting in our classification system.

This training procedure is especially well-suited to training Fisher's linear discriminant (FLD) and support vector machine (SVM) classifiers based on how each forms the separating hyperplane. Because both seek to maximize the margin between two classes, training these classifiers with feature sets that best maintain the structure of the waveform will allow for maximum separation between classes in feature space. For this reason, we train at high SNR. Neural networks behave differently, however. At a fundamental level, neural networks are linear; performing a weighted sum of inputs to form its net activation. The network emits an output that is a nonlinear function of its activation given by

$$g_k(x) = \sigma \left(\sum_{j=1}^n \omega_{kj} \left(\sigma \sum_{i=1}^d (\omega_{ji} x_i + b_j) \right) + b_k \right), \quad (4)$$

where i and j index the input and hidden layer, respectively, ω and b denote the weights and biases, and σ denotes the nonlinear activation function. Because Equation 4 is a nonlinear function, data input into the SHLNN are mapped non-linearly in feature space altering class separability. To properly utilize this non-linearity, it is best to train using a more diverse data set where the structure of each class is widely varied. These widely varied classes allow for the SHLNN to account for the structure of the features in diverse backgrounds. To create a diverse training set, we generate data at multiple SNRs as noted in Section 3.2.

3.1 Single Hidden Layer Neural Network Performance

To test our classification algorithm, we apply a SHLNN using Tensorflow⁷ with different pulse widths than in our training phase, as shown in Table 1. We postulate that a SHLNN should trend similarly to the FLD classifier⁸ and the SVM classifier⁹ given that, at the most fundamental level, a SHLNN behaves linearly.

Figure 1, displays a comparison of the overall accuracy of the FLD, SVM, and SHLNN classifiers for a range of SNRs. Note, that accuracy curves in both plots show a comparison of SHLNN performance under two different training methodologies. This will be explained in further detail in Section 3.2. The left-hand plot in Figure 1 shows that at high SNR, the SHLNN performs similarly to both the FLD and SVM classifiers. As SNR declines below 2 dB, performance degrades quickly. Because *good* performance is defined as accuracy greater than 90%, it is interesting that the SHLNN performs 2 dB worse than the FLD classifier and 4 dB worse than the SVM classifier.

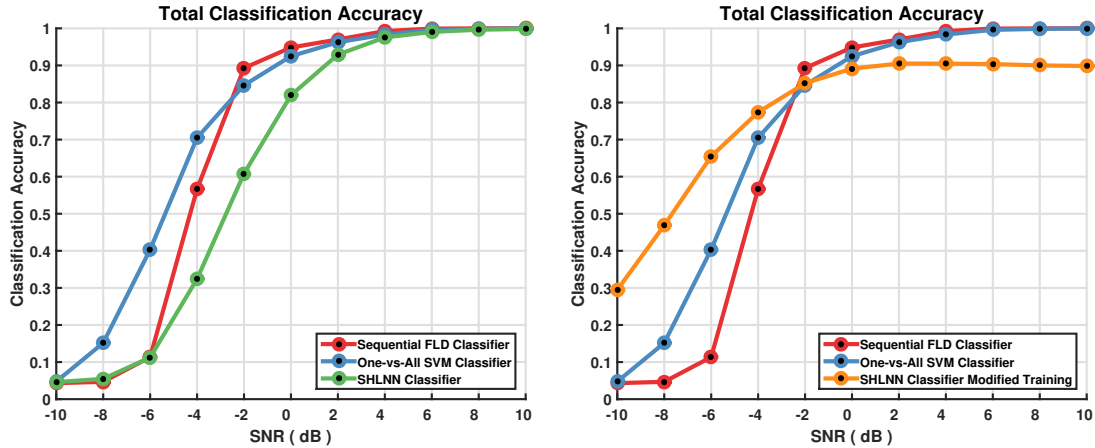


Figure 1. (left) Overall classifier accuracy using training method in the ambiguity-based classification algorithm.¹ Note, that while performance is comparable at high SNR, a sharp decline in performance occurs at 0 dB. (right) Overall classifier accuracy using a modified training method. Observe, overall performance increases but reaches as asymptotic limit at about 90% accuracy.

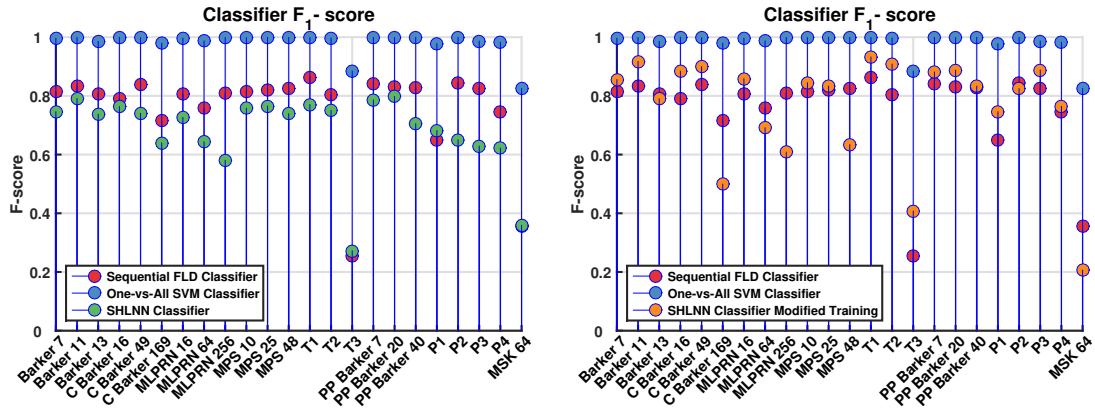


Figure 2. (left) Comparison of classifier F_1 -score when trained in a identically. Interestingly, the FLD and SHLNN classifiers show similar trends in their abilities to classify each modulation. (right) Comparison of classifier F_1 -score when the SHLNN incorporates a modified training method. Notice, per-class performance is widely varied.

As noted in the ambiguity-based classification algorithm,¹ accuracy alone does not provide a sufficient indication of classifier performance. Instead, the combination of accuracy and the F_1 -score given by

$$F_{\beta} = (1 + \beta^2) \frac{\text{precision} \cdot \text{sensitivity}}{(\beta^2 \cdot \text{precision}) + \text{sensitivity}} \Big|_{\beta=1} = 2 \cdot \frac{\text{precision} \cdot \text{sensitivity}}{\text{precision} + \text{sensitivity}}, \quad (5)$$

where precision is a measure of a classifier's exactness and sensitivity is a measure of a classifier's completeness. Figure 2 displays the F_1 -score of each class for the FLD, SVM, and SHLNN classifiers. Interestingly, while both the FLD and SVM classifiers outperform the SHLNN, the F_1 -scores of the FLD and SHLNN trend similarly for the majority of classes.

At this point, it is appropriate to mention the right-hand plots in Figures 1 and 2. Because neural networks, in general, differ greatly from other linear classifiers in the way features are processed, we institute a methodology to improve our training phase. Using this modified training methodology, we are able to drastically improve

classifier accuracy as shown in the right-hand plot of Figures 1 and 2. Notice, however that our overall accuracy reaches an asymptotic limit at around 90%. While finding the optimal training set is a difficult problem in neural network research, we do not overly dwell on this for SHLNNs.

3.2 Modified Training Methodology

Based on other works,⁵⁶³ neural networks seem to naturally work well with highly varied training data. As such, we remove the restraint of training our classifiers strictly at 10 dB SNR. Instead, based on our performance displayed in Figures 1 and 2, we see that for all of our classifiers, accuracy sharply declines after 0 dB SNR and asymptotically approaches a minimum accuracy of approximately 5%. Thus, we choose to retrain our SHLNN with 1000 Monte Carlo iterations of 0 dB and -10 dB each. Note that all other random variables in our original training methodology remain.

As stated in the previous section, optimizing this methodology for the SHLNN was not the focus of this work. As such, the asymptotic performance limit of approximately 90% could be improved by incorporating additional training data at a higher SNR.

4. CONVOLUTIONAL NEURAL NETWORK CLASSIFICATION

Recently, convolutional neural networks have received increasing notoriety. From applications in facial, handwriting, and image recognition, CNNs have shown to provide outstanding performance.⁵⁶ Much of this increase in performance is due to its ability to extract meaningful features from large and complex sets of data. Because CNN classifiers have worked well with classifying images, we alter our feature set to maximally utilize the CNN algorithm.

Specifically, we choose to train and test the CNN classifier with features computed directly from the ambiguity function of an intercepted pulse as shown in Figure 3. For each waveform class, one half of the discrete AF is computed and the resulting 101-by-101 image is input into the CNN. Note, that for training the CNN, we follow the both our original¹ and modified training methodology as discussed in Section 3.2.

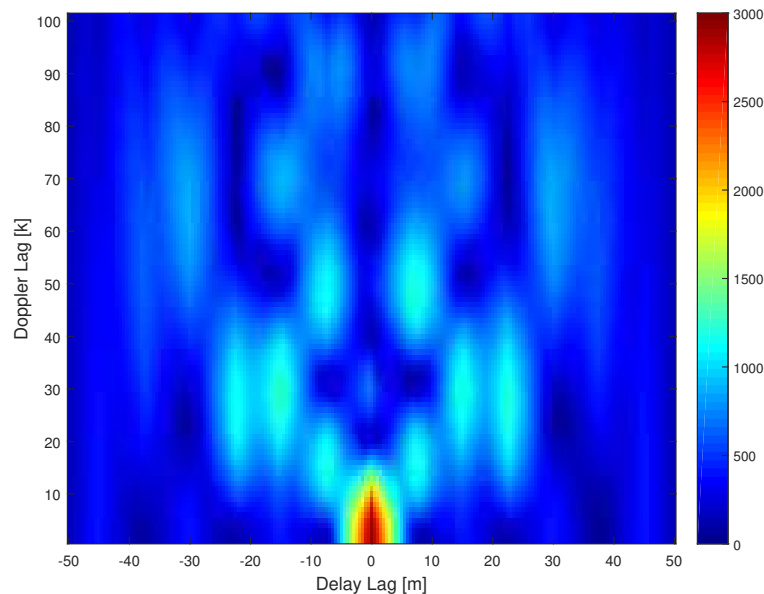


Figure 3. Discrete ambiguity function of a Barker 7 waveform.

4.1 Convolutional Neural Network Performance

Figure 4 displays the classification accuracy and F_1 -score of the CNN classifier under both training methodologies. Using the original training methodology,¹ the accuracy of the CNN trends similarly to the FLD, SVM, and SHLNN. Interestingly, *good* performance ($\geq 90\%$) is achieved at 0 dB but, both the FLD and SVM have higher accuracy. Furthermore, the F_1 -score of the CNN classifier appears to closely match the FLD classifier.

As shown, performance is drastically increased under the modified training methodology. Training via the modified methodology, overall accuracy of the CNN far exceeds that of any other classifier. Near-perfect accuracy is achieved by the CNN for all SNRs above 0 dB, before gracefully declining at -2 dB. Furthermore, the CNN achieves approximately 60% accuracy at -10 dB; a difference of nearly 32% from the next nearest classifier.

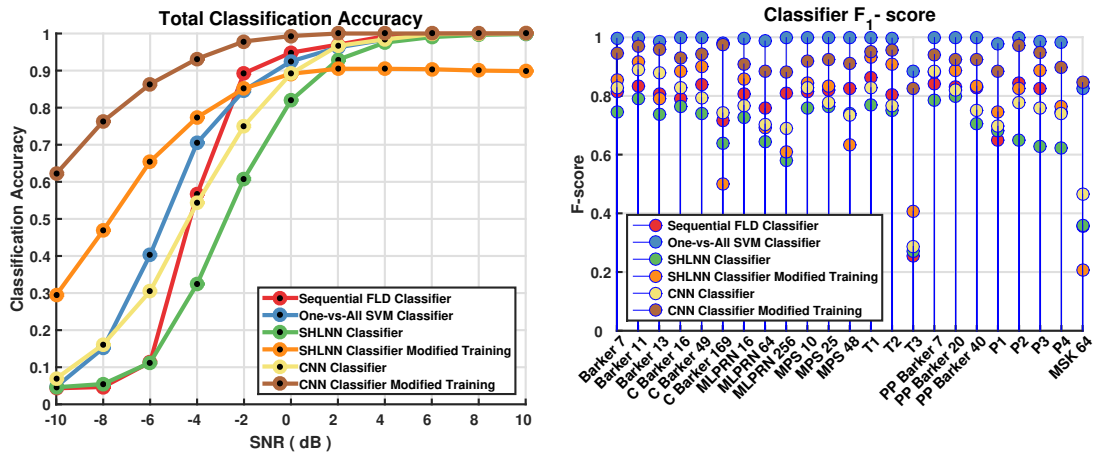


Figure 4. (left) Comparison of overall classifier accuracy. Note, that when all classifiers are trained using the methodology in the ambiguity-based classification algorithm,¹ accuracy trends are similar. Using the modified training approach, performance at low SNR is greatly increased. (right) Comparison of classifier F_1 -scores. Note, that while the overall accuracy of the CNN classifier using the modified training is superior to all others tested, the one-vs-all SVM maintains the highest F_1 -score.

But, as previously stated, overall accuracy is not a complete indicator of the performance of a classifier. The right-hand plot displays the F_1 -score of all classifiers. As shown, the CNN classifier using the modified training methodology performs well on all waveform classes. Note, that while the CNN trained using the modified methodology has superior overall accuracy, the SVM classifier has a superior F_1 -score on nearly all waveform classes. This is due to the one-vs-all structure of the SVM classifier. Because each class is compared to all remaining classes, false positives are minimized by allowing for the *all* classification decision, thus, the SVM classifier has superior precision. Conversely, the CNN classifier forces a classification into one of the 23 trained classes resulting in an increased amount of false positives.

Table 2. Aggregate F_1 -score for each classifier.

Classifier	Aggregate F_1 -score
Sequential FLD	.7597
One-vs-All SVM	.9818
SHLNN	.6804
SHLNN Modified Training	.7648
CNN	.7491
CNN Modified Training	.9223

Because we seek a classification system that achieves the highest overall performance without respect to any one particular class, assessing our classifiers in terms of an average, or aggregate, F_1 -score is appropriate. This

measure, like our overall accuracy measure, serves as a composite metric to easily compare all classifiers. Table 2 displays the aggregate F_1 -scores for each classifier. For each classifier, the F_1 -scores of each waveform class are averaged, providing a measure of the entire classification system. As previously mentioned, the one-vs-all SVM attains the highest aggregate F_1 -score followed closely by the CNN using the modified training. The SHLNN using the training methodology in the ambiguity-based classification algorithm¹ provides the worst F_1 -score. As previously noted, the relatively low performance of the SHLNN is due to the training methodology. When training using the modified methodology, performance is increased, resulting in an aggregate F_1 -score slightly higher than that of the FLD and CNN classifiers under the original training methodology.¹ This modification in our training was especially beneficial to the CNN performance, where the F_1 -score experienced a 17% increase.

5. CONCLUSION

To enable spectrum sharing, modern RF systems must employ situational awareness algorithms to detect, characterize, and identify other spectrum users. Due to the advances in communications and radar waveform design, incorporating an algorithm to recognize a wide range of modulations provides crucial information to empower spectrum knowledge in RF devices. In this paper, we present a comparison of performance for a single hidden layer neural network and a convolutional neural network applied to the phase modulated waveform recognition problem. We assess our performance against sequential FLD and one-vs-all SVM classifiers and document key findings. Future work will explore incorporating a broader range of waveforms and resolving waveform ambiguities resulting from complex environments.

REFERENCES

- [1] Buchenroth, A., Rigling, B., and Chakravarthy, V., "Ambiguity-based classification of phase modulated radar waveforms," in [*Radar Conference (RadarConf), 2016 IEEE*], 1–6, IEEE (2016).
- [2] Rigling, B. D. and Roush, C., "Acf-based classification of phase modulated waveforms," in [*Radar Conference, 2010 IEEE*], 287–291, IEEE (2010).
- [3] Lundén, J. and Koivunen, V., "Automatic radar waveform recognition," *IEEE Journal of Selected Topics in Signal Processing* **1**(1), 124–136 (2007).
- [4] Pavy, A. M. and Rigling, B. D., "Phase modulated radar waveform classification using quantile one-class svms," in [*Radar Conference (RadarCon), 2015 IEEE*], 0745–0750, IEEE (2015).
- [5] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* **86**(11), 2278–2324 (1998).
- [6] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "Imagenet classification with deep convolutional neural networks," in [*Advances in neural information processing systems*], 1097–1105 (2012).
- [7] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X., "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015).
- [8] Duda, R. O., Hart, P. E., and Stork, D. G., [*Pattern classification*], John Wiley & Sons (2012).
- [9] Chang, C.-C. and Lin, C.-J., "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**(3), 27 (2011).