A Fast Soft Decision Algorithm for Cooperative Spectrum Sensing

Mehran Golvaei and Mohammad Fakharzadeh, *Senior Member, IEEE*

***Abstract*—Hidden Primary User problem caused by fading and shadowing severely affects the detection rate of the cognitive radio systems with a single spectrum sensor. Cooperative Spectrum Sensing has been introduced to tackle this problem by using the spatial diversity of spectrum sensors. It is shown that the use of soft decision algorithms in fusion center has a better performance than hard decision algorithms. The problem of soft decision based on sensor measurements perfectly matches the Machine Learning paradigm. In this brief, a novel fast soft decision algorithm is proposed based on Machine Learning theory for wideband Cooperative Spectrum Sensing, which finds a decision boundary to classify the Power Spectral Density (PSD) measurement vectors of sensors into the empty and occupied channel classes. The statistical characteristics of sensors PSD samples are employed to derive a fast solution, which outperforms the SVM-linear algorithm. By solving an optimization problem a Constant False Alarm Rate algorithm is introduced and then it is improved to the Adaptive False Alarm Rate algorithm. The proposed algorithm reduces the average training time to one tenth of the SVM-linear training time, while the detection probability is the same. The hardware implementation of the proposed algorithm is described in Verilog HDL and the corresponding simulation results are presented.**

***Index Terms*—Cognitive radio, cooperative spectrum sensing, machine learning, support vector machine, wideband spectrum sensing.**

I. Introduction

THE EVER increasing demands for higher data rates in wireless communication, while the available spectrum is limited has motivated the use of Cognitive Radio (CR) for the efficient spectrum management. Spectrum sensing is a critical part of CR systems. Nevertheless, the Hidden Primary User (PU) problem can severely defect the detection rate of the conventional spectrum sensing methods. To tackle this problem, Cooperative Spectrum Sensing (CSS) is introduced, which takes advantage of the spatial diversity of the available spectrum sensors. In CSS, the Fusion Center (FC) determines the channel state based on spectrum sensors reports.

There are two categories of decision making approaches in CSS systems. In the *hard* decision approach, each sensor makes an initial decision about the channel state individually using a conventional spectrum sensing algorithm. Based on these reported decisions, FC decides about the final channel state [1]. In the *soft* decision systems, all sensors report the estimated energy of the received signal over the sensing period to FC, without any decision. Simultaneous measurements from the different sensors form vectors, which must be classified into the empty or occupied channel classes by FC. It is shown that soft decision has better performance compared to hard decision [2]. Interpreting soft decision as a classification problem is a significant motivation to use Machine Learning (ML) algorithms [3]. The measurement vectors in FC can be treated as Feature Vectors (FVs). Generally, ML consists of two stages. First, in the *training* stage a decision boundary is derived by applying ML algorithm to training data, i.e., FVs. Then, in the *test* stage, the decision boundary is used to classify new FVs. By using ML the effective parameters of sensing environment, are learned during the training process. Thus, the system can adapt to the sensing environmental change [4]. For example, a major challenge in spectrum sensing is noise calibration, because any error in estimating noise power can lead to SNR wall problem [5]. ML can resolve such issues. Nevertheless, the training process is usually time-consuming and needs to be repeated due to the environmental changes, which causes long pauses in data transmission. In [6], a Deep Convolutional Neural Network is applied to the CSS system, but the problem of long training time is not solved. Linear Support Vector Machine (SVM-Linear) is known to have the best performance in terms of Receiver Operation Characteristics (ROC) [3], but it is computationally expensive.

In this brief a novel cooperative wideband spectrum sensing method is introduced based on the classification of Power Spectral Density (PSD) vectors, which uses the statistical characteristics of the Secondary User (SU) signals to reduce the computational complexity. Consequently, the training time is decreased up to one tenth of the SVM-based methods, and the long pause during data transmission is avoided, while accurate channel detection is achieved. Furthermore, a lower computational complexity decreases the CR power consumption. Furthermore, the Verilog HDL implementation of the proposed algorithm is presented in this brief.

This brief is organized as follows. Section II describes the system model, and the statistical analysis. Section III introduces Constant False Alarm Rate (CFAR) and Adaptive False Alarm Rate (AFAR) fast soft decision algorithms, and compares their performance to SVM-linear algorithm through various simulations. The hardware implementation of

Manuscript received June 29, 2020; accepted July 12, 2020. Date of publication July 20, 2020; date of current version December 21, 2020. This brief was recommended by Associate Editor C. W. Sham. *(Corresponding author: Mohammad Fakharzadeh.)*

The authors are with the Electrical Engineering Department, Sharif University of Technology, Tehran 14699993451, Iran (e-mail: golvaei.mehran@ee.sharif.edu; fakharzadeh@sharif.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2020.3010587

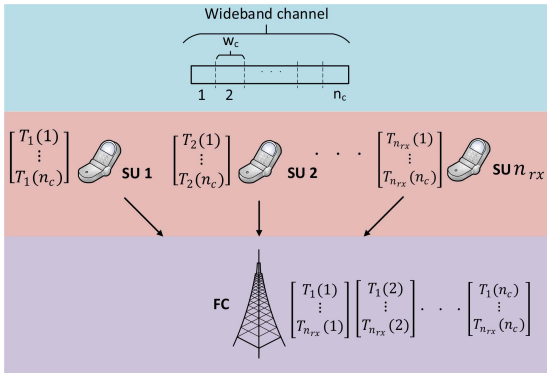

Fig. 1. System model for wideband cooperative spectrum sensing.

the proposed AFAR algorithm is explained in Section IV. Section V concludes this brief.

II. Cooperative Spectrum Sensing Model

A. System Model

Fig. 1 illustrates the wideband spectrum sensing problem, where the allowed spectrum consists of n_c non-overlapping narrowband channels. Each channel has a bandwidth of w_c, in which at most one PU transmits signal. There are n_{rx} spectrum sensors, i.e., SU, which apply Fast Fourier Transform (FFT) on the received signal to derive its PSD. The FFT frame length the and number of averaging frames are denoted by L and M, respectively. These PSD vectors are reported to the Fusion Center (FC), where FVs are formed for each channel. Then, an ML algorithm is applied to each channel to derive a decision rule to determine the channel state. Wideband sensing is achieved by sensing the narrowband channels sequentially. In other words, ML algorithm is applied to each channels to derive the decision rules, which signifies the importance of the fast training stage.

B. The Statistical Model

The discrete time domain samples $x_r[n] = s_r[n] + v_r[n]$ are received at r^{th} sensor, where $s_r[n]$ denotes the PU signal and $v_r[n]$ represents the Additive White Gaussian Noise (AWGN) with zero mean and variance $\sigma^2_{v,r}$. Using Parseval's theorem in frequency domain [7], the power of channel c at sensor r, denoted by $t_r(c)$, is calculated as

$$t_r(c) = \frac{1}{ML} \sum_{m=0}^{M-1} \sum_{k=k_c^l}^{k_c^h} |X_r^m[k]|^2, \tag{1}$$

where $X_r^m[k]$ is k^{th} bin from FFT of $x_r[n]$ in m^{th} frame. Moreover, k_c^l and k_c^h are the lower and higher bin numbers of channel c, respectively. Using the central limit theorem [8], the statistical model for t_r is derived as

$$\begin{aligned} H_r^0 &\sim \mathcal{N}(\sigma^2_{vf,r}, \frac{\sigma^4_{vf,r}}{MK}), \\ H_r^1 &\sim \mathcal{N}(\sigma^2_{vf,r} + \sigma^2_{s,r}, \frac{(\sigma^2_{vf,r} + \sigma^2_{s,r})^2}{MK}), \end{aligned} \tag{2}$$

where $\sigma^2_{vf,r}$ and $\sigma^2_{s,r}$ denote the noise and signal power in channel c at r^{th} sensor, respectively and K is the width of channel c. The distribution of t_r in the absence of PU signal, i.e., when SU only receives noise, is represented by H_r^0. Furthermore, H_r^1 denotes the distribution of t_r in the presence of PU signal. The SNR in channel c at sensor r is given by

$$\text{SNR}_r = \frac{\sigma^2_{s,r}}{\sigma^2_{vf,r}}. \tag{3}$$

The fading effect is modeled as a random loss coefficient, because it is not deterministic and varies for different sensors. Thus, (2) is rewritten as

$$\begin{aligned} H_r^0 &\sim \mathcal{N}(\sigma^2_{vf,r}, \frac{\sigma^4_{vf,r}}{MK}), \\ H_r^1 &\sim \mathcal{N}(\sigma^2_{vf,r} + l_r\sigma^2_{s,r}, \frac{(\sigma^2_{vf,r} + l_r\sigma^2_{s,r})^2}{MK}), \end{aligned} \tag{4}$$

where l_r is the loss coefficient due to the fading effect in channel c and sensor r, which has a uniform distribution over [0, 1]. It varies over sensing time because of the environmental changes. The reported FV for channel c is

$$\mathbf{t} = \begin{bmatrix} t_1 \\ \vdots \\ t_{n_{rx}} \end{bmatrix}. \tag{5}$$

Since t_r always follows the Gaussian distribution, $\mathbf{t}$ has a Multivariate Gaussian distribution.

C. Decision Rule

The previous studies have shown that the SVM-linear algorithm, referred to as SVM, has the best performance with regard to ROC [3]. SVM proposes a hyperplane as the decision boundary. So, the classification rule for $\mathbf{t}$ is written as

$$(\mathbf{t} - \mathbf{p})^T \mathbf{u} \overset{?}{>} 0, \tag{6}$$

where $\mathbf{p}$ is a point in the hyperplane decision boundary and $\mathbf{u}$ is its normal vector [9]. Thus, the decision boundary is determined by finding $\mathbf{p}$ and $\mathbf{u}$, denoted as $b_{\mathbf{u},\mathbf{p}}$. We propose a method to find these values in a fast and simple way.

III. The Fast Soft Decision Algorithm for CSS

SVM is a general solution for finding the decision boundary, while FVs characteristics are partially known, but its complexity is high. In this Section, first a Constant False Alarm Rate (CFAR) algorithm is proposed. Then it is improved to an Adaptive False Alarm Rate (AFAR) algorithm to achieve the SVM detection rate with less computation and training time.

A. Constant False Alarm Rate Algorithm

According to (4), in the absence of PU, every element of $\mathbf{t}$ follows H_r^0. So, the distribution of $\mathbf{t}$ is normal, or $\mathbf{t}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \Sigma_0)$, where $\boldsymbol{\mu}_0$ is

$$\boldsymbol{\mu}_0 = \begin{bmatrix} \sigma^2_{vf,1} \\ \vdots \\ \sigma^2_{vf,n_{rx}} \end{bmatrix}. \tag{7}$$

The received noise signals in different sensors are uncorrelated, so Σ_0 is a diagonal matrix with the following diagonal elements

$$[\frac{\sigma_{vf,1}^4}{MK}, \ldots, \frac{\sigma_{vf,n_{rx}}^4}{MK}]. \quad (8)$$

As noise variance is identical for all sensors, $\mathbf{t}_0$ has Symmetric Multivariate Gaussian distribution. The variance and mean of each element of $\mathbf{t}_0$ is denoted by σ_0 and μ_0, respectively. For an arbitrary boundary of $b_{\mathbf{u},\mathbf{p}}$, the false alarm probability, P_{fa}, is derived using the tail distribution of $\mathbf{t}_0$,

$$P_{fa}(b_{\mathbf{u},\mathbf{p}}) = 1 - F_{\mathbf{t}_0}(b_{\mathbf{u},\mathbf{p}}), \quad (9)$$

where $F_{\mathbf{t}_0}$ is the cumulative distribution function (CDF) of $\mathbf{t}_0$. As $\mathbf{t}_0$ is symmetric, the CDF value only depends on the hyperplane distance from $\boldsymbol{\mu}_0$. Fig. 2 depicts the probability density function (PDF) for $\mathbf{t}_0$ in 2D space, which is constant over the dotted circle. Because of symmetry, CDF values for any line such as l_1, l_2 or l_3 are the same. In general, by using a boundary such as l_1, P_{fa} is derived

$$P_{fa}(b_{\mathbf{u},\mathbf{p}}) = 1 - \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \int_{t_1<\mu_0+c}^{+\infty} \prod_{i=1}^{n_{rx}} f(t_i)\, dt_1 \ldots dt_{n_{rx}}, \quad (10)$$

where $f(t_i)$ is the PDF of t_i and t_i's are independent. One can further simplify (10) to

$$P_{fa}(b_{\mathbf{u},\mathbf{p}}) = 1 - F_{t_1}(\mu_0 + c). \quad (11)$$

Let $\mathbf{p}$ be the nearest point of the hyperplane to $\boldsymbol{\mu}_0$. If c in (11) is replaced by $\rho\sigma_0$, then

$$\mathbf{p} = \boldsymbol{\mu}_0 + \rho\sigma_0\hat{\mathbf{u}}, \quad (12)$$

where $\hat{\mathbf{u}} = \frac{\mathbf{u}}{\|\mathbf{u}\|}$ for any desired $\mathbf{u}$ [10]. Thus,

$$P_{fa}(b_{\mathbf{u},\mathbf{p}}) = \frac{1}{2}\Big[1 - erf(\rho/\sqrt{2})\Big]. \quad (13)$$

By choosing a proper value for ρ, P_{fa} can be fixed at a desired constant value. It is worth noting that (13) is independent of n_{rx}. The next step is to find the proper $\mathbf{u}$, which maximizes the detection probability, P_d. First, we investigate the system with zero fading condition. In the presence of PU, the distribution of $\mathbf{t}$ is normal or $\mathbf{t}_1 \sim \mathcal{N}(\boldsymbol{\mu}_1, \Sigma_1)$, where $\boldsymbol{\mu}_1$ is

$$\boldsymbol{\mu}_1 = \begin{bmatrix} \sigma_{vf,1}^2 + \sigma_{s,1}^2 \\ \vdots \\ \sigma_{vf,n_{rx}}^2 + \sigma_{s,n_{rx}}^2 \end{bmatrix}. \quad (14)$$

The detection probability, P_d, is given by

$$P_d(b_{\mathbf{u},\mathbf{p}}) = 1 - F_{\mathbf{t}_1}(b_{\mathbf{u},\mathbf{p}}), \quad (15)$$

where $F_{\mathbf{t}_1}$ is the CDF of $\mathbf{t}_1$. In this case, all received signals by sensors are correlated due to the presence of the PU signal. However, since SNR is usually low, $\mathbf{t}_1$ is assumed to follow the Symmetric Multivariate Gaussian Distribution. Similar to P_{fa}, the distance of $\boldsymbol{\mu}_1$ from the hyperplane determines the P_d value. Assuming P_{fa} is fixed by choosing a proper ρ, we want to find $\mathbf{u}$, which maximizes P_d. Thus, an optimization problem is stated as

$$\begin{aligned} &\underset{\mathbf{u}}{\text{argmax}} \quad d = Min(\|\boldsymbol{\mu}_1 - \mathbf{x}\|), \\ &\text{subject to} \quad (\mathbf{x} - \mathbf{p})^T\mathbf{u} = 0, \end{aligned} \quad (16)$$

where $\mathbf{u} = \boldsymbol{\mu}_1 - \boldsymbol{\mu}_0$ is the solution. Although this solution is derived for zero fading probability, it can be applied to the deep fading cases.

CFAR algorithm is simulated for various SNR and fading conditions, which are called simulation scenarios. The obtained values for P_d and P_{fa} are sorted with respect to SVM P_d and plotted.

First, CFAR algorithm is simulated for $P_{fa} = 5\%$. Fig. 3 shows that P_{fa} of CFAR is properly fixed at 5%. As illustrated in Fig. 3 inset, when P_{fa} of CFAR is lower than P_{fa} of SVM, the corresponding P_d is severely decreased.

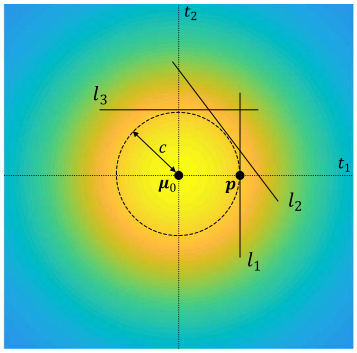

Fig. 2. Demonstration of a 2D Symmetric Multivariate Gaussian Distribution.

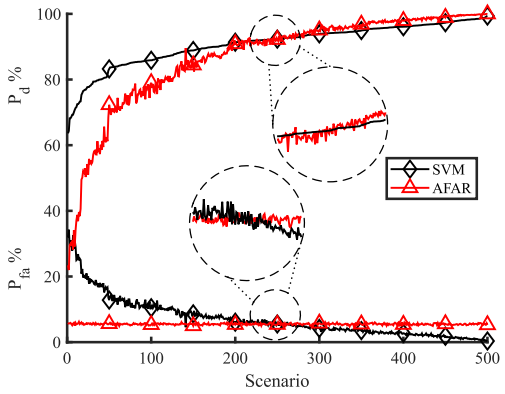

Fig. 3. CFAR algorithm results for $P_{fa} = 5\%$ for a 10-sensor CSS system.

B. Adaptive False Alarm Rate Algorithm

Fig. 4 shows CFAR results for $P_{fa} = 25\%$. Compared to Fig. 3, for $P_{fa} = 25\%$, CFAR algorithm shows a better performance in following SVM results. Thus, if P_{fa} is varied adaptively, a better performance in following SVM results will be achieved. For high values of P_{fa}, the proposed algorithm must set its false alarm rate to a large value. On the other hand, as P_{fa} reduces, the proposed algorithm must avoid unnecessary large P_{fa} values and reduce its threshold. This explains the core idea of AFAR algorithm. Simulation results

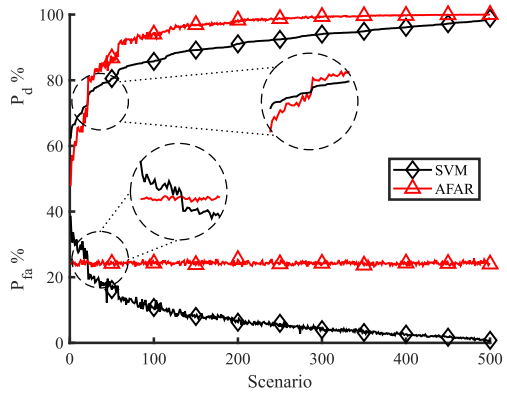

Fig. 4. CFAR method result for $P_{fa} = 25\%$ for the 10-sensor system.

TABLE I
$\|\mathbf{u}\|$, P_{fa} AND ρ VALUES FOR $M = 20, L = 2048$

$\|\mathbf{u}\| * 10^{-16}$	P_{fa} %	ρ
>100	3	1.96
100-80	8	1.42
80-55	14	1.10
55-44	18	0.94
44-35	25	0.69
35-28	29	0.56
28-22	34	0.41
22-18	38	0.28
18-14	43	0.15
14-11	47	0.06
<11	50	0.00

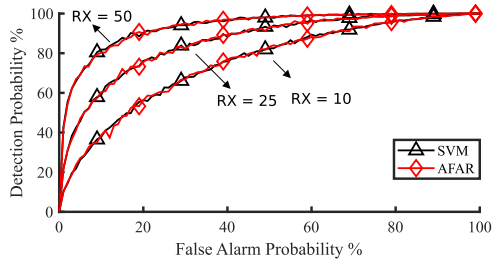

Fig. 5. SVM and AFAR ROC for 10, 25 and 50 sensors with SNR $= -15$ dB.

TABLE II
COMPARISON OF AFAR AND SVM TEST AND TRAINING TIMES IN MS

n_{rx}	SVM train	AFAR train	SVM test	AFAR test
10	83	2	2	1
25	100	5	12	1
35	110	8	16	1
55	120	13	23	1

show that as empty and occupied channel clusters get close to each other, P_{fa} increases. The distance between the cluster means ($\|\mathbf{u}\|$) can be used to measure the interference between clusters. By excessive simulation of SVM in different scenarios a table of corresponding values of $\|\mathbf{u}\|$ and P_{fa} is formed (Table I), which is independent of n_{rx}, noise or signal power and only depends on M and L. Also, P_{fa} of algorithm is set by ρ.

Fig. 5 depicts ROC of SVM and AFAR algorithm for $\text{SNR}_r = -15dB$ for different number of sensors. AFAR successfully follows ROC of SVM, while it is much faster.

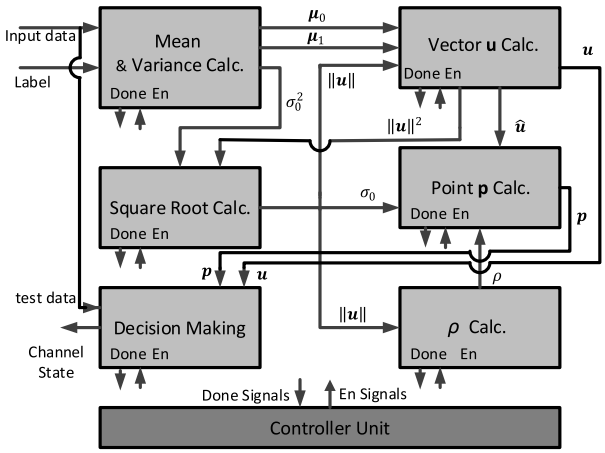

Fig. 6. Hardware structure diagram.

Table II compares the execution time of AFAR and SVM algorithms on a Windows 8, Core*i*7, 8G RAM device, in terms of training and test times. It is seen that the proposed AFAR algorithm has a constant test time and the training time is one order of magnitude shorter than that of the SVM.

IV. HARDWARE IMPLEMENTATION AND RESULTS

A. Hardware Overview

Fig. 6 shows an overview of the hardware structure and computation blocks. To distinguish the occupied channel and empty channel classes in the training stage, a label signal is used, where 0 indicates the empty channel and 1 indicates the occupied channel. The hardware consists of computation blocks to perform the operations noted in Section III. Finally, the Decision Making Unit indicates the channel state. To control the sequence of computation, a controller unit is designed. Using the fixed-point simulation and error analysis, the input data is a 10-bit length signal. The number of training samples is a power of 2, where the excessive simulation shows 2^{10} is sufficient.

B. Mean and Variance Calculation Unit

As described in Section III, the mean and the standard deviation values of the occupied and empty channel clusters are required. The mean values are calculated by dividing the sum of each class samples to the total number. The sum is calculated by an accumulated shift register. Since the total number of samples is 1024, the division is done by ignoring the 10 least significant bits (LSB) of the accumulation result. The variance is calculated as

$$\sigma_0^2 = \left(\frac{1}{n}\sum_{i=1}^{n} x_i^2\right) - \bar{x}^2. \tag{17}$$

To find the standard deviation (σ_0), the square root of the variance (σ_0^2) is calculated using a look-up table. The sorted values of σ_0^2 and the corresponding σ are stored in the same addresses of memory A and memory B, respectively. The root value address (in memory B) is determined by comparing the input value with the square values sequentially (in memory A).

TABLE III
COMPARISON OF AFAR AND SVM HARDWARE COMPLEXITY

Implementation	AFAR - 55 RX	SVM -19 RX [11]
Clock Frequency (MHz)	50	50
Logic Elements	747	6395
Memory Bits	2620	444928
9x9 Multipliers	16	240
Training Clock Cycles	119165	253656348
Training Time (ms)	2.383	5070

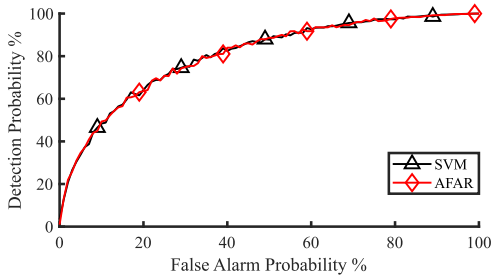

Fig. 7. ROC of AFAR and SVM for 25-sensor system with $\mathrm{SNR}_r = -15dB$.

The input samples range and SNR values indicate square and root values range.

C. Normal Vector Calculation Unit

To calculate $||\mathbf{u}||$, its squared value is fed to Square Root Calculation Unit. Then $\hat{\mathbf{u}}$ is calculated by using the sequential subtraction as division. To perform division operation, each $\mathbf{u}$ member is concatenated with 10 zero bits as LSBs, so all precision bits are considered in the subtraction result.

D. Other Units

To calculate the coefficient ρ, $||\mathbf{u}||$ values are compared with the maximum limits of the predefined ranges, and the result is mapped to the corresponding ρ value memory address using a priority encoder. The parameter $\mathbf{p}$ is the weighted sum of $\boldsymbol{\mu}_0$ and $\hat{\mathbf{u}}$. The weighted summation is calculated sequentially for each dimension using a shift register.

According to (6), the channel state is determined by sign of the inner product of $\mathbf{u}$ and the difference between the input sample and $\mathbf{p}$, which is performed sequentially using an accumulator. Because the sign is important, signals are sign extended to 11 bits. After n_{rx} clock, the accumulator sign bit represents the channel state. Moreover, the Controller Unit is implemented using a Finite State Machine (FSM), whose states correspond to the activation sequence of the functional units. The *done* signals are the inputs and *enable* signals are the outputs of this FSM.

E. Simulation Results

The hardware is described by Verilog HDL. AFAR algorithm hardware description code is synthesized on Cyclone IV FPGA using Quartus Prime. Table III compares the clock frequency, hardware complexity and the training stage clock cycles of AFAR and SVM hardware implementation of [11].

The detection performance of AFAR algorithm hardware is simulated for a 25-sensor system. Fig. 7 shows that for

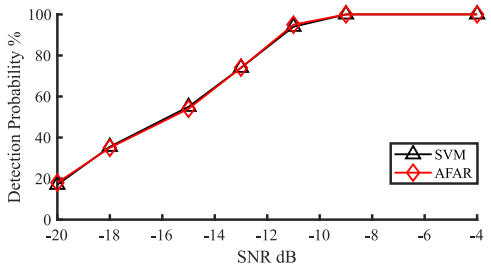

Fig. 8. SVM and AFAR P_d for 25-sensor system.

$\mathrm{SNR}_r = -15$ dB, ROC of AFAR follows SVM closely. Fig. 8 depicts the detection probability P_d of AFAR and SVM for this system for different SNR values.

V. CONCLUSION

In this brief a fast soft decision algorithm for CSS systems using ML theory was proposed, which outperforms SVM-Linear by decreasing the training and test time to one tenth. First, we redefine the problem of finding decision boundary as finding a point ($\mathbf{p}$) of the hyperplane and its normal vector ($\mathbf{u}$). Based on the statistical analysis of FVs in different channel states in CFAR algorithm ($\mathbf{p}$) is chosen so that false alarm rate is fixed in a desired constant value and ($\mathbf{u}$) is chosen so that the detection probability is maximized for given false alarm rate. In AFAR algorithm by using cluster means distance ($\|\mathbf{u}\|$) as an indicator of data separability, the false alarm rate is changed adaptively to follow SVM-linear detection rates.

REFERENCES

[1] K. Cichoń, A. Kliks, and H. Bogucka, "Energy-efficient cooperative spectrum sensing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1861–1886, 3rd Quart., 2016.
[2] I. Akyildiz, B. Lo, and R. Balakrishnan, "Cooperative spectrum sensing in cognitive radio networks: A survey," *Phys. Commun.*, vol. 4, pp. 40–62, Mar. 2011.
[3] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2209–2221, Nov. 2013.
[4] C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
[5] R. Tandra and A. Sahai, "SNR walls for signal detection," *IEEE J. Sel. Topics Signal Process.*, vol. 2, no. 1, pp. 4–17, Feb. 2008.
[6] W. Lee, M. Kim, and D. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, Mar. 2019.
[7] T. Yu, S. Rodriguez-Parera, D. Marković, and D. Čabrić, "Cognitive radio wideband spectrum sensing using multitap windowing and power detection with threshold adaptation," in *Proc. IEEE Int. Conf. Commun.*, Cape Town, South Africa, May 2010, pp. 1–6.
[8] A. Vosoughi, J. R. Cavallaro, and A. Marshall, "A context-aware trust framework for resilient distributed cooperative spectrum sensing in dynamic settings," *IEEE Trans. Veh. Technol.*, vol. 66, no. 10, pp. 9177–9191, Oct. 2017.
[9] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411
[10] J. K. Patel and C. B. Read, *Handbook of the Normal Distribution, Second Edition*. New York, NY, USA: Marcel Dekker,Inc, 1996.
[11] T. Kuan, J. Wang, J. Wang, P. Lin, and G. Gu, "VLSI design of an SVM learning core on sequential minimal optimization algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 4, pp. 673–683, Apr. 2012.