Deep Spiking Delayed Feedback Reservoirs and Its Application in Spectrum Sensing of MIMO-OFDM Dynamic Spectrum Sharing

Kian Hamedani,[1] **Lingjia Liu,**[1] **Shiya Liu,**[1] **Haibo He,**[2] **Yang Yi**[1]
[1]Department of Electrical and Computer Engineering, Virginia Tech, USA
{hkian, ljliu, shiyal, yangyi8}@vt.edu
[2]Department of Electrical, Computer, and Biomedical Engineering, University of Rhode Island, USA
{haibohe}@uri.edu

Abstract

In this paper, we introduce a deep spiking delayed feedback reservoir (DFR) model to combine DFR with spiking neuros: DFRs are a new type of recurrent neural networks (RNNs) that are able to capture the temporal correlations in time series while spiking neurons are energy-efficient and biologically plausible neurons models. The introduced deep spiking DFR model is energy-efficient and has the capability of analyzing time series signals. The corresponding field programmable gate arrays (FPGA)-based hardware implementation of such deep spiking DFR model is introduced and the underlying energy-efficiency and recourse utilization are evaluated. Various spike encoding schemes are explored and the optimal spike encoding scheme to analyze the time series has been identified. To be specific, we evaluate the performance of the introduced model using the spectrum occupancy time series data in MIMO-OFDM based cognitive radio (CR) in dynamic spectrum sharing (DSS) networks. In a MIMO-OFDM DSS system, available spectrum is very scarce and efficient utilization of spectrum is very essential. To improve the spectrum efficiency, the first step is to identify the frequency bands that are not utilized by the existing users so that a secondary user (SU) can use them for transmission. Due to the channel correlation as well as users' activities, there is a significant temporal correlation in the spectrum occupancy behavior of the frequency bands in different time slots. The introduced deep spiking DFR model is used to capture the temporal correlation of the spectrum occupancy time series and predict the idle/busy subcarriers in future time slots for potential spectrum access. Evaluation results suggest that our introduced model achieves higher area under curve (AUC) in the receiver operating characteristic (ROC) curve compared with the traditional energy detection-based strategies and the learning-based support vector machines (SVMs).

1 Introduction

Recurrent neural network (RNNs) have shown to be very powerful tools to analyze time series data. However, Due to the vanishing gradients, the traditional RNNs are very challenging or even impossible to train. Reservoir computing (RC) is a new generation of RNNs that is much easier to train, and in many cases have shown equivalent or even improved performances compared to the traditional RNNs (Palumbo et al. 2016; Antonelo, Schrauwen, and Stroobandt 2008). There are three different types of RC systems, echo state networks (ESN), delayed feedback reservoirs (DFR), and liquid state machines (LSM) (Lukoševičius and Jaeger 2009). The RC paradigms avoid the gradient vanishing problem through generating a recurrent reservoir layer where the weights of the connections between neurons are randomly created and remain fixed during the training (Schrauwen, Verstraeten, and Van Campenhout 2007). DFRs are the most recent generation of RC models and are constructed through a single neural model and a delay loop which performs as the reservoir layer (Antonik et al. 2017). It is shown to be powerful for attack detection in smart grids (Hamedani et al. 2018) and video based face recognition in cyber physical systems (CPS) (Hamedani et al. 2019; Bai et al. 2018).

Various mathematical models have been introduced to represent artificial neurons. The spiking neural networks (SNNs) are the most biologically plausible model with extremely high energy-efficiency since the spikes are the main signal format that the neurons of our brains adopt to communicate with each other(Uzzell and Chichilnisky 2004). Truenorth is a SNN chip which was developed by IBM in 2014 (Esser et al. 2016), and it forms a network composed of 1 million artificial neurons with 256 million synapses. Despite this large number of neurons and synapses, the Truenorth consumes only 70 milli-Watts (mW) in power.

It is known that neurons use spikes to communicate in our brains, however, the exact encoding approach the neurons adopt to encode the information is still unclear. Rate encoding and temporal encoding are the two major neural encoding schemes that have been introduced where temporal encoding schemes have drawn more attention recently due to their superior performance over rate encoding (Panzeri et al. 2010). Therefore, in this paper, we adopt temporal encoding for neural encoding.

To evaluate the introduced deep spiking DFR (DSDFR) model we focus on the application of spectrum occupancy prediction of a dynamic spectrum sharing (DSS) system using multiple-input-multiple-output (MIMO) orthogonal-frequency-division multiplexing (OFDM) technologies. The

mobile data traffic is predicted to grow 46 percent every year between 2017 to 2022 due to the vast popularity of smart phones and upcoming fifth generation (5G) mobile broadband networks (Cisco 2019). Due to the spectrum scarcity several solutions including network densification, DSS, and massive MIMO have been introduced in 5G networks (Atat et al. 2017).

In this paper, we focus on DSS which resolves the spectrum scarcity in 5G systems through sharing the spectrum among the users. The combination of MIMO and OFDM technologies improves the spectral efficiency, as the MIMO utilizes the spatial multiplexing gain and the OFDM avoids frequency selective fading. MIMO-OFDM has already been adopted in fourth generation (4G) LTE/LTE-Advanced (Liu et al. 2012) and will remain as the dominant technology in 5G as well. However, in a MIMO-OFDM system not all subcarriers are utilized simultaneously by the primary user (PU) leading to low spectrum utilization efficiency. The MIMO-OFDM based cognitive radios (CRs) in a DSS environment introduces a solution to resolve this problem where the under-utilized subcarriers can be used by the secondary users (SUs)/CRs. In DSS environments the SUs monitor the spectrum utilization of PUs through spectrum sensing and transmit data on the idle subcarriers (Chen et al. 2016). The main contributions of this paper are :

- Develop an energy efficient DSDFR model that can be used to analyze the time series in a DSS system.
- Explore different temporal encoding schemes and identify the optimal one.
- Develop a FPGA-based hardware implementation of DFRs and evaluate its efficiency and performance.

The organization of this paper is as follows. In Section II, our introduced model and the MIMO-OFDM DSS environment are presented. Section III shows the performance evaluation and corresponding FPGA implementation. Section IV concludes the paper and discusses the future work.

2 Deep Spiking Delayed Feedback Reservoirs

Figure 1 shows the model of the introduced DSDFR. There are three major layers in this model: 1) the input layer where the spectrum occupancy time series are received and prepossessed; 2) the reservoir layer is where the memory is formed and the temporal correlation of the time series is extracted; 3) the temporal information that is extracted by the reservoir layer is used for training the output layer.

2.1 MIMO-OFDM Input Time Series

In OFDM multi-carrier transmissions, $L-$point inverse discrete Fourier transform (IDFT) is applied to the modulated PU symbols. In this paper, we assume Quadrature Phase Shift Keying (QPSK) is used to modulate the transmit symbols. Subsequently, the cyclic prefix (CP) is added to the symbols and the symbols are transmitted over a wireless channel. Eq. 1 represents the k^{th} QPSK symbol generated by the PU after passing through the IDFT

$$s\left(t-kT_i\right)=\sum_{l=0}^{L-1} S_{k,l} e^{\frac{j2\pi l(t-kT_i)}{T_i}} e^{j2\pi f_i(t-kT_i)}, \quad (1)$$

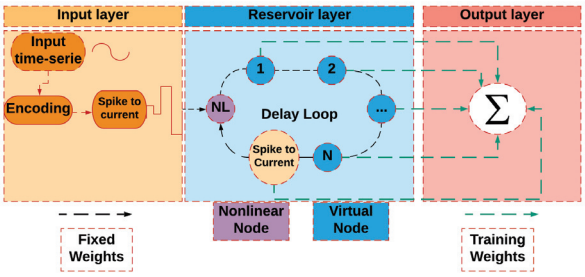

Figure 1: Architecture of DSDFR model.

where $S_{k,l}$ is the PU symbol that is modulated on the l^{th} subcarrier, f_i is the QPSK carrier frequency, and T_i is the QPSK symbol duration. This symbol is transmitted over a fading Rayleigh channel and the received signal is down converted to the baseband and then passes through the $P-$point discrete Fourier transform (DFT). The n^{th} OFDM symbol can be written as,

$$y\left(t-nT_s\right)=e^{-j2\pi f_s(t-nT_s)}\sum_{m=0}^{M-1} h_m s(t-kT_i-mT_s),$$

where h_m is the coefficient of the Rayleigh channel, T_s is the OFDM symbol period, and f_s is the OFDM carrier frequency. The signal transmitted on the p^{th} subcarrier is (Tse and Viswanath 2005):

$$\begin{aligned} Y_p(n)=&\sum_{l=0}^{L-1} X_{k,l} H_l e^{j2\pi(-kf_iT_i+nf_sT_s)} e^{j\pi\beta_{l,p}(P-1)} \\ &\times \frac{\sin(\pi\beta_{l,p}P)}{\sin(\pi\beta_{l,p})} \qquad 1\le p\le P, \end{aligned} \quad (2)$$

where $H_l = \sum_{m=0}^{M-1} h_m e^{-j2\pi m(f_iT_s+lT_s/T_i)}$, $\beta_{l,p} = (k/T_i+f_i-f_s)T_s/M - m/M$, and M is the number of subcarriers. Therefore, the received signal at the CR which is transmitted on the p^{th} subcarrier can be written as,

$$R_p(n)=Y_p(n)+N_p(n), \quad (3)$$

where $N_p(n)$ is the DFT of complex additive white Gaussian noise (AWGN) with zero mean and unit variance. The main objective of this paper is to determine the presence or absence of the signal on the p^{th} subcarrier. The presence and absence of the received signal are denoted as two hypothesis, H_1 and H_0 respectively. Therefore, the received signal can be expressed as

$$R_p(n)=\begin{cases} N_p(n) & H_0 \\ Y_p(n)+N_p(n) & H_1 \end{cases} \quad n=1,\ldots,N, \quad (4)$$

where N is the number of OFDM received symbols. In the energy detection based spectrum sensing, the decision statistics of each subcarrier is formed based on the average received energy of N symbols and is expressed as follows,

$$E_p=\begin{cases} \frac{1}{N}\sum_{n=1}^{N}\left|N_p(n)\right|^2 & H_0 \\ \frac{1}{N}\sum_{n=1}^{N}\left|Y_p(n)+N_p(n)\right|^2 & H_1. \end{cases} \quad (5)$$

The decision statistics, E_p, is compared with a threshold value, if the E_p is larger than the threshold then the subcarrier is considered as busy, otherwise the subcarrier is denoted as idle. The threshold is calculated based on the given probability of false alarm, P_f. It is ideal to have a high probability of detection for each subcarrier, P_d^p, while maintaining a low P_f^p. P_d^p and P_f^p are defined as (Liang et al. 2008),

$$P_f^p = \Pr\left(E_p > \epsilon^p | H_0\right) = Q\left(\left(\frac{\epsilon}{\sigma_n^2} - 1\right)\sqrt{N}\right),$$

$$P_d^p = \Pr\left(E_p > \epsilon^p | H_1\right) = Q\left(\left(\frac{\epsilon}{\sigma_n^2} - \gamma^p - 1\right)\frac{\sqrt{N}}{\gamma^p + 1}\right),$$

where ϵ^p is the energy detection threshold for subcarrier p^{th}, γ^p is the signal-to-noise ratio (not in dB) of the p^{th} subcarrier, $Q(\cdot)$ is the complementary function of a standard Gaussian distribution, and σ_n^2 is the noise variance which in this paper is assumed to be 1. The ϵ^p is calculated based on the given P_f^p.

For the case where there are multiple antennas at the CR receiver, the received signal of the p^{th} subcarrier at the j^{th} antenna can be expressed,

$$R_p^j(n) = Y_p^j(n) + N_p^j(n) \qquad j = 1, \ldots, N_R, \tag{6}$$

where N_R is the number of CR antennas. The corresponding decision statistics, E_p, can be expressed as

$$E_p = \begin{cases} \frac{1}{NN_R}\sum_{j=1}^{N_R}\sum_{n=1}^{N}\left|N_p^j(n)\right|^2 & H_0 \\ \frac{1}{NN_R}\sum_{j=1}^{N_R}\sum_{n=1}^{N}\left|Y_p^j(n) + N_p^j(n)\right|^2 & H_1. \end{cases}$$

The corresponding P_f^p and P_d^p can be written as

$$\begin{aligned} P_f^p &= Q\left(\left(\frac{\epsilon}{\sigma_n^2} - 1\right)\sqrt{NN_R}\right) \\ P_d^p &= Q\left(\left(\frac{\epsilon}{\sigma_n^2} - \gamma^p - 1\right)\frac{\sqrt{NN_R}}{\gamma^p + 1}\right). \end{aligned} \tag{7}$$

In most of the literature, it is assumed that the PU activity follows a Markov chain which is not very practical. To incorporate the temporal correlation of the PU activity, in this paper, instead of using Markov chains we use real world spectrum occupancy data to represent the PU activity. Static spectrum occupancy measurement campaign (Wellens, de Baynast, and Mahonen 2008) has been conducted by RWTH Aachen University to measure the PU activity in different frequency bands and time slots. Their results show that in each frequency band, a significant temporal correlation exists in the PU activity in different time slots. In this paper, the occupancy of each subcarrier is modeled based on its corresponding frequency occupancy model that is extracted from RWTH Aachen University's spectrum occupancy database. The input time series are generated using the energy of the received signals by the SUs.

2.2 Encoding

In order to process the time series using the artificial spiking neurons, the input signals have to be first encoded. As mentioned in Section 1, we focus on temporal encoding due to their superior performance. There are two major temporal encoding schemes: 1) latency or time to first spike (TTFS) encoding and; 2) interspike interval (ISI) encoding. In this section, we will study how to apply them on time series MIMO-OFDM data.

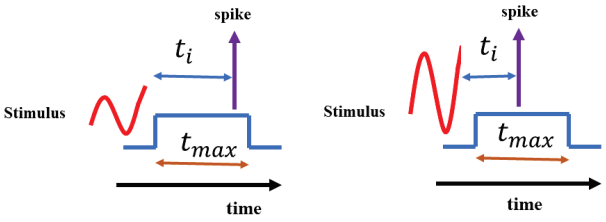

Figure 2: TTFS encoding.

TTFS Encoding In latency or TTFS encoding approach, the stimulus is encoded as the latency between the stimulus onset and the observation time of the first spike in an encoding time window.

Accordingly, TTFS can be expressed as (Hu et al. 2013):

$$TTFS = t_i = t_{\max} - \ln\left(\alpha \times r_i + 1\right), \tag{8}$$

where $t_{\max}$ is the length of the encoding window; α is the scaling factor; and r_i is the value of the stimulus (in our case, r_i is the received energy at the SUs/CRs). An example of TTFS encoding is demonstrated in Figure 2. As it can be seen, for the stimulus with higher intensity the latency is shorter. In this paper, t_{max} is a hyperparemeter that requires tuning. For each received signal in the spectrum occupancy time series, its corresponding TTFS is calculated using Equation 8 and is fed into the next block.

ISI Encoing ISI encoding is another category of temporal encoding where the relative distance between multiple spikes is used for encoding the stimulus (Zhao et al. 2015). In Figure 3, an example of ISI encoding is demonstrated. We leverage a nonlinear neuron model for ISI encoding. Equation 9 shows the ISI encoding strategy,

$$D_i = g\left(r_i, C_i, V_i\right) - g\left(r_i, C_{i-1}, V_{i-1}\right), \tag{9}$$

where D_i is the time distance between two consecutive spikes; g is the nonlinear ISI encoding neuron; C_i and V_i are the capacitance and threshold voltage of the i_{th} encoding neuron, respectively. The number of the ISI encoding neurons is a hyperparameter that requires tuning. In this paper, for the sake of computational simplicity, the number of ISI encoding neurons are considered to be 3. The relationship between the number of ISI encoding neurons and the number of spikes can be expressed as $N_S = 2^{N-1}$ where N_S is the number of spikes. Setting $N = 3$, we have $N_S = 4$.

2.3 Spike to Current

After the input time series are encoded, they need to be converted to an analog current to be processed by other neurons. This is inspired by how the information is processed in our brains. The sensory neurons receive the stimulus and encode

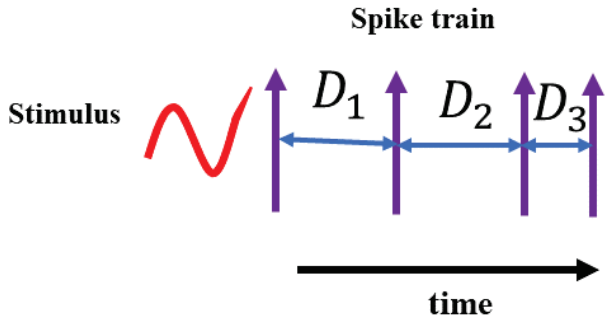

Figure 3: ISI encoding.

the information. Consequently, the encoded information is converted to analog current and then fed into the information processing neurons (Follmann, Goldsmith, and Stein 2018; Yu et al. 2013). The following equation can be leveraged to convert the encoded information to an analog current,

$$I_{PSAC}(t) = \sum_{t^j} \kappa(t - t^j) H(t - t^j), \tag{10}$$

where $I_{PSAC}(t)$ is the postsynaptic analog current; H is the Heaviside function; t^j is the time of spiking; and $\kappa(t - t_j)$ is a kernel function and is defined in Equation 11.

$$\kappa\left(t - t^j\right) = V_0 \times \left(\exp\left(-\frac{t - t^j}{\tau_s}\right) - \exp\left(-\frac{t - t^j}{\tau_f}\right)\right), \tag{11}$$

where V_0 is the normalization factor which helps us make sure that the magnitude of the kernel always remains less than 1; τ_s and τ_f are the slow and fast decay constants and are set to 10 ms and 4 ms in this paper, respectively. For each received signal in the time series, its corresponding analog current is produced and then is fed to the delay reservoir layer to be explained in more details in Section 2.4.

2.4 Delay Reservoir Layer

Delay exists in almost every system.

The delay can affect the performance of the neurons and it is essential to study the effects of the delay on the information processing neurons. In delay systems, the state of the system does not only depend on the current time but also on previous times. Delayed differential equations (DDEs) can be used to express the underlying relationship (Appeltant et al. 2011),

$$\frac{dx(t)}{dt} = f\left[x(t), x(t - \tau)\right], \tag{12}$$

where $f(\cdot)$ is a nonlinear function, and τ is delay value. It is shown that the delayed feedback loops can serve as RC models are capable of capturing the temporal correlations in the time series (Appeltant et al. 2011). The reservoir layer acts as the recurrent layer of the system, and extracts the temporal correlation of the input time series. The temporal features that are extracted from the reservoir layer, are used for training the weights of output layer.

The delay loop is composed of a nonlinear node (NL) and a set of virtual nodes. The nonlinear node in our case is leaky-integrate and fire (LIF) neuron, and the virtual nodes are separated by a fixed delay, θ. The LIF neuron is one of the major mathematical models that represents the behavior of spiking neurons (Wu et al. 2019). Equation 13 shows the mathematical representation of LIF neurons:

$$\tau_m \frac{dV_m}{dt} = -\left(V_m - E\right) + \left(I_{noise} + I_s\right) R_m, \tag{13}$$

where V_m is the membrane voltage of the LIF neuron; E is the rest potential; τ_m is the time constant of the neuron and $\tau_m = R_m C_m$; R_m and C_m are the resistance and capacitance of the neuron which are 1 mega ohms ($M\Omega$) and 10 nano Farads (nF), respectively; I_{noise} is the background noise; I_s is the stimulus current (in our case, it is the analog currents of each received signal in the time series).

The total delay that the delay loop goes through equals to τ, and $\theta = \tau/V$ is the corresponding delay of each virtual node where V represents the number of virtual nodes. The delayed version of each virtual node is stored in the next virtual node, and the output of the last virtual node is multiplied by a feedback gain (g), and then is converted to an analog current so that it can be added to the corresponding current of the next temporal sample in the time series. It is shown that the DFRs can show high dimensional behavior and short-term memory if θ, τ, and g are tuned such that the DFR performs at the edge of chaotic region (Ikeda and Matsumoto 1987; Bertram et al. 2003). Therefore, the virtual nodes are the reservoirs of the DFRs and their job is to form a short-term memory and map the data from a low-dimensional space to a high-dimensional space.

2.5 Output layer

The third layer of the DFR is the output or readout layer. In this layer, the states extracted by the reservoir (virtual) nodes undergo a training algorithm such that the desired output can be estimated by the DFR. After making sure that the training is completed successfully, the data that has not been seen by the DFR is used for testing. The estimated output is the weighted sum of the tapped reservoir states as the readout layer and is expressed as

$$\hat{o}(t) = \sum_{i=1}^{V} w_i x\left(t - \frac{\tau}{V}\left(V - i\right)\right), \tag{14}$$

where $\hat{o}(t)$ is the estimated output, $x(t - \tau/V(V - i))$ is the reservoir state, w_i is the set of training weights. The role of the training algorithm is to find a set of weights such that the mean square error (MSE) between the estimated outputs and the desired outputs is minimized. The MSE is defined as $||WX - O||^2$ where W is the training weight matrix, X is the reservoir state matrix , and O is the desired output matrix. In (Appeltant et al. 2011), a simple Moore-Penrose pseudo-inverse approach has been introduced to estimate W. To further improve the performance, we introduce a hybrid learning algorithm where we use a multi-layer perceptron (MLP) instead of the Moore-Penrose pseudo-inverse to find the optimal set of output weights. We will use stochastic gradient descent (SGD) to train the readout MLP and the cross-entropy will be the loss function. The spectrum sensing is a

binary classification task, and for each class its corresponding label is assigned. In this paper, the label of the busy subcarriers is assigned to be 1, and the label of the idle subcarriers is 0. The temporal features that are extracted by the reservoir layer are used for training a MLP with one hidden layer of 20 neurons.

The number of the input nodes of the MLP equals to the number of the reservoir nodes in the delay loop of the DFR. To avoid over-fitting the cross-validation and regularization are also utilized during the training process.

The feature that we use for training our hybrid spiking DFR and MLP structure is the energy of each received signal in the time series. The DFRs can be superior over other machine learning algorithms in scenarios where the PU activity at each time slot is temporally correlated with its activity in previous and future time slots.

3 Performance Evaluation & Hardware Implementation

3.1 TTFS vs ISI

We use receiver operating characteristics (ROC) curve to evaluate the prediction performance of the introduced DSDFR model. The ROC curve plots the P_d^p with respect to the P_f^p and the detection algorithm that achieves the highest area under the curve (AUC) is the best spectrum sensing algorithm. We perform our simulations for different signal-to-noise ratios (SNR(dB)), and number of transmit (Tx) and receive (Rx) antennas in a MIMO-OFDM DSS based CR.

In Section 2 TTFS and ISI are introduced as two temporal neural encoding schemes. In this section, we compare the performance of these encoding strategies. Table 1 contains the AUC of DSDFR for TTFS and ISI. As shown in the table, the AUC of ISI achieves significantly higher values. This observation implies the superiority of ISI over TTFS in neural spike encoding. In TTFS, the information is encoded with respect to the stimulus onset. Therefore, an external temporal reference is required in TTFS. In ISI, the stimulus is encoded with respect to the relative temporal distance between consecutive spikes in a spike train. Any imprecision in determining the external temporal reference of TTFS can result in lower performances. On the other hand, ISI has shown to convey more information than TTFS (Brasselet et al. 2012). Therefore, it is generally known that ISI is superior to TTFS where our evaluation confirms this understanding.

Table 1: TTFS vs ISI, Tx = 2 , Rx = 2

Method	SNR(db)	AUC
DSDFR-TTFS	-10db	0.92
DSDFR-TTFS	-20db	0.7
DSDFR-TTFS	-30db	0.51
DSDFR-ISI	-10db	**0.97**
DSDFR-ISI	-20db	**0.76**
DSDFR-ISI	-30db	**0.54**

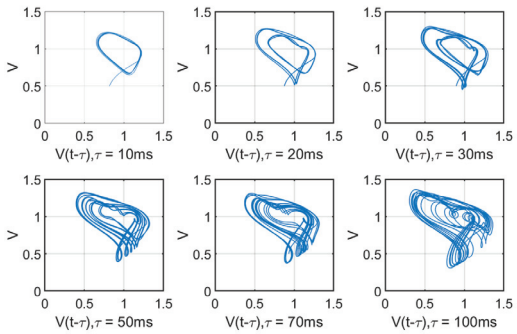

Figure 4: Phase Portrait.

3.2 Effect of Delay on Performance

The delay value of the reservoir layer is an important hyperparameter that requires tuning. It has been shown in (Bertram et al. 2003) that the delay feedback loop can form short-term memory and show high-dimensional behavior only if their delay value is tuned somehow that they can operate at the edge of chaos. The dynamic behavior of the delay loop shifts from periodic to edge of chaos and to completely chaotic. There are several evidences that the neurons of our brain also operate at the edge of chaos (Chua, Sbitnev, and Kim 2012). Therefore, it is essential to tune the delay value of our introduced DSDFR model so that it can operate at the edge of chaos. Plotting the phase portraits is one common method to analyze the dynamic behavior of delay feedback loop systems.

Solving the DDE equation of a delay system helps us to further investigate the dynamic behavior. In fact, by using the phase portraits we can visualize the dynamic behavior of a delay feedback system in long term run.Figure 4 demonstrates an example of a phase portrait corresponding to a delay feedback system. As it can be seen in the figure by increasing the delay value the dynamic behavior of a delay feedback system can shift from ordered to edge of chaos, and completely chaotic.

We investigate the effect of the delay value on DSDFR and identify the optimal delay value in this section. Figure 5 shows the AUC of DSDFR for different values of the delay. AUC is very low for small values of the delay, however, it improves as the delay increases. From the figure We can see that our introduced model achieves the highest AUC for $\tau = 30ms$. Furthermore, when $\tau > 30ms$, the AUC starts to reduce. This observation along with the phase portrait imply that $\tau \approx 30ms$ is the optimal delay value of our introduced model to analyze the MIMO-OFDM DSS CR spectrum occupancy time series. Therefore, we can conclude that for $\tau \approx 30ms$ DSDFR operates at the edge of chaos and can form a short-term memory and show high-dimensional behavior. Accordingly, we set $\tau = 30ms$ in this paper.

3.3 Comparison with other methods

We compare our results with square law combining (SLC) (Kuppusamy and Mahapatra 2008; Chen et al. 2015)

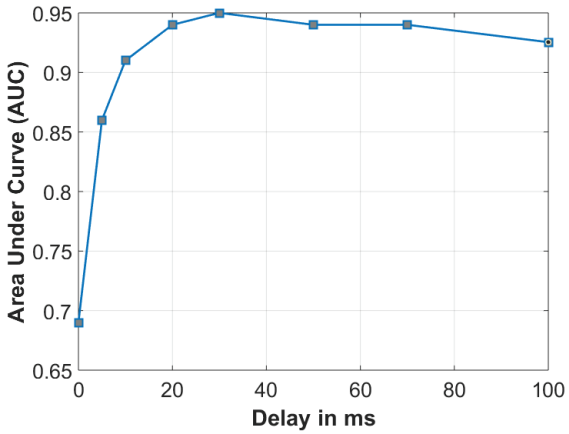

Figure 5: Delay effects: SNR(dB)=$-20dB$, Tx=4, Rx=4.

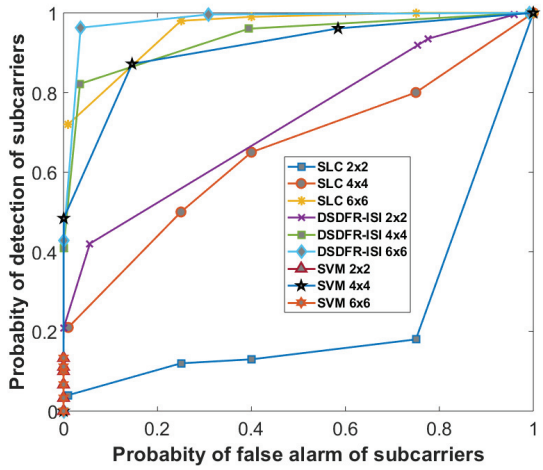

Figure 6: ROC curves for different sensing approaches and different number of antennas at SNR(dB) = $-20dB$.

which is an energy detection based spectrum sensing approach and SVM with radial basis function (RBF) kernel. Note that in the literature, SVMs have shown the highest detection performance in machine learning-based spectrum sensing (Davaslioglu and Sagduyu 2018; Thilina et al. 2013). Therefore, we will compare our results with SVM-based spectrum sensing strategies. The ROC curve of our simulation results at SNR(dB) = $-20dB$ for different numbers of Tx and Rx antennas are presented in Figure 6 for Rayleigh fading channels. As it can be seen in the figure that the DSDFR with ISI encoding outperforms the SLC and SVM for all scenarios because DSDFR covers more AUC compared with the other two approaches.

It can also be observed that as the number of antennas increases, the detection performance improves. A larger number of antennas will increase the spatial multiplexing gain and this helps to achieve higher detection performance as well. Furthermore, we can see in the figure that the SLC and DSDFR achieve the same performances when the number of Tx and Rx antennas equal to 6. However, for lower numbers of Tx and Rx antennas the DSDFR outperforms the other two methods. Table 2 presents the AUC for all these methods.

Table 2: AUC of Different MIMO-OFDM Spectrum Sensing Methods at SNR(dB)=-20 dB

Method	Tx antenna	Rx antenna	AUC
SLC	2	2	0.11
SLC	4	4	0.7
SLC	6	6	0.97
SVM	2	2	0.71
SVM	4	4	0.92
SVM	6	6	0.98
DSDFR-ISI	2	2	**0.76**
DSDFR-ISI	4	4	**0.95**
DSDFR-ISI	6	6	**0.99**

3.4 Architecture of FPGA Implementation

The DFR system is composed of three blocks: a mapping block, a delayed loop block and a readout block. As soon as, an input is received, it is firstly sent to the encoding block. In the encoding block, a one-dimensional input is mapped to 10 neurons. It is achieved by multiplying the input with a vector of synapse weights with size 10. These synapse weights are stored in block RAMs (BRAMs) to minimize the latency. Then, the output neurons are buffered in BRAMs and later will be used by DFR loop block. DFR loop block has two inputs. One input is the output neurons from encoding block and another input is the output of DFR loop block from previous time sample. To minimize the latency, weights of these fully-connected layers and intermediate data between these two layers are stored in BRAMs.

3.5 FPGA-based DFR Hardware Implementation

To show the energy-efficiency of our scheme, we implement the DFR-based spectrum sensing platform on FPGAs and evaluate its speed and energy consumption. The introduced DFR system is implemented on the Xilinx Zynq®-7000 FPGA with Dual ARM®Cortex®-A9 MPCore™ with CoreSight™. Vivado HLS is the tool we used to implement the DFR system. Vivado HLS is a high-level synthesis tool that convert C/C++ into RTL. We exported the DFR system implementation in Vivado HLS as an IP core which is imported and synthesized by Vivado. Finally, Vivado will generate bitstream and program the FPGA board.

Our hardware architecture is memory efficient: Only 3 out of 280 available BRAMs are used leading to 1% of the BRAM utilization rate. In terms of flip-flop, our DFR system consumes 6070 out of 106400 available flip-flops and the utilization rate is 5%. The LUT (Look-Up Table) utilization rate is 18% with 9976 out of 53200 LUTs are used in the DFR system. In terms of DSP, our DFR consumes 44 of 220 available DSP and the utilization rate is 20%. Meanwhile, the speed of our DFR system is fast: It is able to complete the inference of 5102 samples in 0.302 seconds(s). The

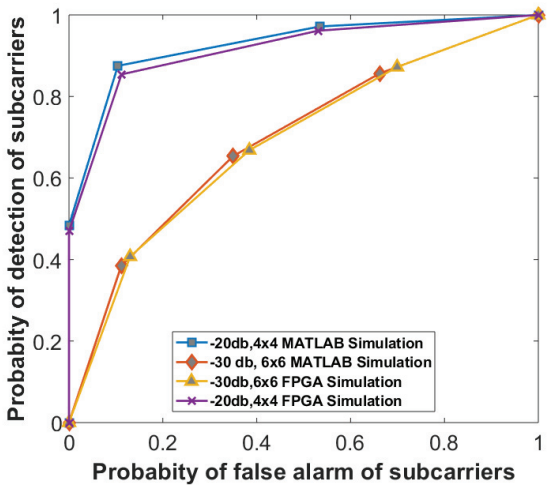

Figure 7: ROC curve of MATLAB simulations and FPGA Implementation platform.

power consumption of our DFR system block is 0.199 Watts (W). Figure 7 shows that the result of the FPGA implementation and that of software implementations are very similar: the FPGA and MATLAB implementations achieve the same AUC. Table 3 summarizes the speed and power consumption performance and Table 4 summarizes the resource utilization of our DFR system. We can see that the resource utilization and power consumption of this model is low, despite the large time series dataset that is being processed.

Our introduced DFR model can not only improve the AUC in spectrum sensing but reduce the computational complexity as well leading to an efficient design. The low computational complexity of DFR is due to the fact that the delay loop of DFR can extract the temporal correlation that exist in the data, and as a result of that the readout layer of DFR which in our case is a MLP can use this temporal information and identify the decisions boundaries more easily. However, the SVM cannot leverage this temporal information and spends more time to identify the decision boundaries. The computational complexity of the DFR based spectrum sensing comes only from the read out layer, where in our case is a MLP. The weights of the delay loop of the DFR are fixed and do not cause any computational complexity. In order to compute the computational complexity, the total number of floating point operations (FLOPs) have to be calculated(Hunger 2005). It has been shown in (Mosleh et al. 2017) that the computational complexity of calculating the output weights of a RC system is approximated by T_{trian}^3 where T_{trian} corresponds to the training time.

Table 3: Performance of DFR System

Blocks	Inference time	Power Consumption
DFR	0.302s	0.199W

Table 4: Resource Utilization of DFR System

Resource	Used	Available	Utilization(%)
DSP	44	220	20%
BRAMs	3	280	1%
Flip-Flop	6070	106400	5%
LUT	9976	53200	18%

4 Conclusion and Future Work

In this work, we introduced a spiking model of delayed feedback reservoirs that can be used for analyzing time series. RNNs have shown to be very useful for time series analysis. However, the traditional RNNs are not easy to be implemented by hardware. In this paper, we presented a deep spiking model of DFRs that can capture the temporal correlation, and at the same time is very efficient for hardware implementation. The spectrum occupancy time series data of MIMO-OFDM CRs in DSS environments was used to evaluate the performance of our introduced model. We explored two temporal neural spike encoding schemes to show that the ISI encoding outperforms the TTFS significantly in terms of AUC. Our introduced DSDFR using ISI encoding achieves better performances compared with SLC which is an energy detection based spectrum sensing approach, and SVM as well. The FPGA based hardware implementation of our introduced DFR model verifies the energy efficiency of this model, as a very small portion of the computational resources of FPGA are consumed.

In the future work, we will develop an architecture where multiple layers of DSDFRs are stacked together. Stacking more layers of DFRs could potentially increase the AUC.

5 Acknowledgement

This work was supported by the U.S. National Science Foundation under grants ECCS 1731672, ECCS-1802710, ECCS-1811497, CNS-1811720, and CCF-1937487.

References

Antonelo, E. A.; Schrauwen, B.; and Stroobandt, D. 2008. Event detection and localization for small mobile robots using reservoir computing. *Neural Netw.* 21(6):862–871.

Antonik, P.; Hermans, M.; Haelterman, M.; and Massar, S. 2017. Chaotic time series prediction using a photonic reservoir computer with output feedback. In *In Proc. of the AAAI Conf. on Artificial Intell.*

Appeltant, L.; Soriano, M. C.; Van der Sande, G.; Danckaert, J.; Massar, S.; Dambre, J.; Schrauwen, B.; Mirasso, C. R.; and Fischer, I. 2011. Information processing using a single dynamical node as complex system. *Nature commun.* 2:468.

Atat, R.; Liu, L.; Chen, H.; Wu, J.; Li, H.; and Yi, Y. 2017. Enabling cyber-physical communication in 5g cellular networks: challenges, spatial spectrum sensing, and cybersecurity. *IET Cyber-Phys. Syst.: Theory Appl.* 2(1):49–54.

Bai, K.; Li, J.; Hamedani, K.; and Yi, Y. 2018. Enabling an new era of brain-inspired computing: Energy-efficient

spiking neural network with ring topology. In *2018 55th ACM/ESDA/IEEE Design Autom. Conf. (DAC)*, 1–6.

Bertram, M.; Beta, C.; Pollmann, M.; Mikhailov, A. S.; Rotermund, H. H.; and Ertl, G. 2003. Pattern formation on the edge of chaos: Experiments with co oxidation on a pt (110) surface under global delayed feedback. *Physical Review E* 67(3):036208.

Brasselet, R.; Panzeri, S.; Logothetis, N. K.; and Kayser, C. 2012. Neurons with stereotyped and rapid responses provide a reference frame for relative temporal coding in primate auditory cortex. *J. of Neuroscience* 32(9):2998–3008.

Chen, H.; Liu, L.; Matyjas, J. D.; and Medley, M. J. 2015. Optimal resource allocation for sensing-based spectrum sharing d2d networks. *Comput. & Electr. Eng.* 44:107 – 121.

Chen, H.; Liu, L.; Novlan, T.; Matyjas, J. D.; Ng, B. L.; and Zhang, J. 2016. Spatial spectrum sensing-based device-to-device cellular networks. *IEEE Trans. Wireless Commun.* 15(11):7299–7313.

Chua, L.; Sbitnev, V.; and Kim, H. 2012. Neurons are poised near the edge of chaos. *Intl J. of Bifurcation and Chaos* 22(04):1250098.

Cisco. 2019. Visual networking index: Global mobile data traffic forecast update, 2017–2022. *Cisco White paper*.

Davaslioglu, K., and Sagduyu, Y. E. 2018. Generative adversarial learning for spectrum sensing. In *2018 IEEE Intl Conf. on Commun. (ICC)*, 1–6.

Esser, S. K.; Merolla, P. A.; Arthur, J. V.; Cassidy, A. S.; Appuswamy, R.; Andreopoulos, A.; Berg, D. J.; McKinstry, J. L.; Melano, T.; Barch, D. R.; di Nolfo, C.; Datta, P.; Amir, A.; Taba, B.; Flickner, M. D.; and Modha, D. S. 2016. Convolutional networks for fast, energy-efficient neuromorphic computing. *Proc. of the National Academy of Sciences (NAS)* 113(41):11441–11446.

Follmann, R.; Goldsmith, C. J.; and Stein, W. 2018. Multimodal sensory information is represented by a combinatorial code in a sensorimotor system. *PLoS biology* 16(10):e2004527.

Hamedani, K.; Liu, L.; Atat, R.; Wu, J.; and Yi, Y. 2018. Reservoir computing meets smart grids: Attack detection using delayed feedback networks. *IEEE Trans. Ind. Informat.* 14(2):734–743.

Hamedani, K.; Liu, L.; Hu, S.; Ashdown, J.; Wu, J.; and Yi, Y. 2019. Detecting dynamic attacks in smart grids using reservoir computing: A spiking delayed feedback reservoir based approach. *IEEE Trans. Emerging Topics. Computational Intell*.

Hu, J.; Tang, H.; Tan, K. C.; Li, H.; and Shi, L. 2013. A spike-timing-based integrated model for pattern recognition. *Neural comput.* 25(2):450–472.

Hunger, R. 2005. *Floating point operations in matrix-vector calculus*. Munich University of Technology, Inst. for Circuit Theory and Signal . . ..

Ikeda, K., and Matsumoto, K. 1987. High-dimensional chaotic behavior in systems with time-delayed feedback. *Physica D: Nonlinear Phenomena* 29(1-2):223–235.

Kuppusamy, V., and Mahapatra, R. 2008. Primary user detection in ofdm based mimo cognitive radio. In *2008 3rd Intl Conf on Cogn. Radio Oriented Wireless Netw. and Commun. (CrownCom 2008)*, 1–5.

Liang, Y.; Zeng, Y.; Peh, E. C. Y.; and Hoang, A. T. 2008. Sensing-throughput tradeoff for cognitive radio networks. *IEEE Trans. Wireless. Comm* 7(4):1326–1337.

Liu, L.; Chen, R.; Geirhofer, S.; Sayana, K.; Shi, Z.; and Zhou, Y. 2012. Downlink MIMO in LTE-Advanced: SU-MIMO vs. MU-MIMO. *IEEE Commun. Mag.* 50(2):140–147.

Lukoševičius, M., and Jaeger, H. 2009. Reservoir computing approaches to recurrent neural network training. *Computer Science Review* 3(3):127–149.

Mosleh, S.; Liu, L.; Sahin, C.; Zheng, Y. R.; and Yi, Y. 2017. Brain-inspired wireless communications: Where reservoir computing meets mimo-ofdm. *IEEE Trans. Neural Nets. & Learning Syst.* 29(99):1–15.

Palumbo, F.; Gallicchio, C.; Pucci, R.; and Micheli, A. 2016. Human activity recognition using multisensor data fusion based on reservoir computing. *J. of Ambient Intell. and Smart Environments (JAISE)* 8(2):87–107.

Panzeri, S.; Brunel, N.; Logothetis, N. K.; and Kayser, C. 2010. Sensory neural codes using multiplexed temporal scales. *Trends in neurosciences* 33(3):111–120.

Schrauwen, B.; Verstraeten, D.; and Van Campenhout, J. 2007. An overview of reservoir computing: theory, applications and implementations. In *Proc. of the 15th Eur. Symp. on Artificial Neural Netw.*, 471–482.

Thilina, K. M.; Choi, K. W.; Saquib, N.; and Hossain, E. 2013. Machine learning techniques for cooperative spectrum sensing in cognitive radio networks. *IEEE J. Sel. Areas Commun* 31(11):2209–2221.

Tse, D., and Viswanath, P. 2005. *Fundamentals of wireless communication*. Cambridge university press.

Uzzell, V., and Chichilnisky, E. 2004. Precision of spike trains in primate retinal ganglion cells. *J. of Neurophysiology* 92(2):780–789.

Wellens, M.; de Baynast, A.; and Mahonen, P. 2008. Exploiting historical spectrum occupancy information for adaptive spectrum sensing. In *2008 IEEE Wireless Commun. and Netw. Conf.*, 717–722.

Wu, Y.; Deng, L.; Li, G.; Zhu, J.; Xie, Y.; and Shi, L. 2019. Direct training for spiking neural networks: Faster, larger, better. In *Proc. of the AAAI Conf. on Artificial Intell.*, volume 33, 1311–1318.

Yu, Q.; Tang, H.; Tan, K. C.; and Li, H. 2013. Precise-spike-driven synaptic plasticity: Learning hetero-association of spatiotemporal spike patterns. *Plos one* 8(11):e78318.

Zhao, C.; Wysocki, B. T.; Liu, Y.; Thiem, C. D.; McDonald, N. R.; and Yi, Y. 2015. Spike-time-dependent encoding for neuromorphic processors. *ACM J. Emerg. Technol. in Comput. Syst. (JETC)* 12(3):23.