

© 2023 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Galib, Md Mehedi Hassan, Mohamed F. Younis, and Brian W. Stevens. "Spectral-Temporal Model for Opportunistic Spectrum Access in Cognitive Radio Networks." IEEE Transactions on Cognitive Communications and Networking, 2023, 1–1. <https://doi.org/10.1109/TCCN.2023.3342428>.

<https://doi.org/10.1109/TCCN.2023.3342428>

Access to this work was provided by the University of Maryland, Baltimore County (UMBC) ScholarWorks@UMBC digital repository on the Maryland Shared Open Access (MD-SOAR) platform.

### **Please provide feedback**

Please support the ScholarWorks@UMBC repository by emailing [scholarworks-group@umbc.edu](mailto:scholarworks-group@umbc.edu) and telling us what having access to this work means to you and why it's important to you. Thank you.

# Spectral-Temporal Model for Opportunistic Spectrum Access in Cognitive Radio Networks

Md Mehedi Hassan Galib, Mohamed F. Younis, *Fellow, IEEE*, Brian W. Stevens, *Member, IEEE*

**Abstract**—In the quest for overcoming spectrum scarcity, cognitive radios opt to dynamically exploit underutilized frequency bands in a primary wireless network to support communication among secondary users. However, the high fluctuation of channel usage requires continuous network monitoring through spectrum sensing, complicates medium access control, and forces secondary communications to be opportunistic with unbounded latency. To overcome these challenges, this paper advocates the use of *machine learning* (ML) to form a predictive model for when a channel becomes available. In particular, we propose a *spectral-temporal spike neural network* (ST-SNN). Unlike conventional ML-based approaches, our design tracks the variability of channel selection (spectral aspect) and infers access patterns of the channel over time (temporal aspect). The architecture fundamentally adopts both recurrent and feed-forward neural networks where channel usage tracking data is used for training and predicting the spectrum utilization for the next cycle; the error is fed back to improve the prediction accuracy. Our results using an *Long Term Evolution* (LTE) dataset show that the proposed predictive model achieves about 96.6% accuracy of channel availability assessment.

**Index Terms**—Cognitive radios, Spectrum access prediction, CRN, Spatio-temporal models, Spike neural network.

## I. INTRODUCTION

THE prevalence of wireless applications, fog computing, personal communication devices, and autonomous vehicles have increased demand for radio spectrum exponentially. Yet, all the usable radio frequency bands designated by the *Federal Communications Commission* (FCC) are fully allocated. Hence, supporting the growing demand should be through dynamic spectrum sharing. In [1], it has been noted that the utilization of the allocated spectrum varies temporally and spectrally. According to the FCC data, barely 30% of the US frequency spectrum is used under a fixed allocation strategy, resulting in a substantial temporal spectrum occupancy of milliseconds to hours [2]. Thus, *cognitive radios* (CRs) have grown in popularity as a means for dynamically distributing unused frequency spectrum to *secondary non-licensed users* (SNU) while *principal licensed users* (PLUs) are inactive or broadcasting at low power. Research suggests that SNUs may pursue underlay, overlay, and interweave strategies to access these unutilized regions [3].

**CR Strategies:** In the underlay strategy, SNUs can coexist with transmission interference below the PLU's tolerated threshold, making underlay a form of spectral sharing. The overlay strategy assumes that SNUs have previous knowledge of PLU's

codebooks and messages, and hence allows access to a frequency band only while PLUs are idle, i.e., temporal sharing. For the interweaving technique, also known as opportunistic spectrum access, SNUs must locate the spectral-temporal void in the licensed spectrum, also known as white space, while the PLUs are not broadcasting. SNUs have sporadic access to the spectrum and can begin sending data; however, they must halt their transmissions when PLUs resume. Such a process implies that more frequent and time-domain monitoring of PLU activities is needed when using the interweave strategy. In addition, since SNUs are uncertain when a PLU will transmit next, they must detect white space and wrap up data transfer as soon as possible in order to avoid interference. The interweave strategy is considered for this paper, since it requires little knowledge about the spectrum, and does not cause much inference.

Therefore, to support effective SNU's use of a radio band and avoid interference with PLUs, a *cognitive radio network* (CRN) architecture comprises three fundamental components: spectrum sensing, dynamic spectrum management, and adaptive communications. During the spectrum sensing phase, it is required to identify the empty slots in the frequency band as soon as they become available. Such an effort is made to ensure that SNUs may have completed transmitting their data within the allowable time slot. During the period of dynamic spectrum management, a CRN is responsible for selecting the communication channels that are the most suitable for SNUs. During adaptive communications, SNUs must specify carrier and bandwidth. Among the three CRN phases, the duration of the sensing phase is more dominant, with a chance of false detection of white spaces. Indeed, false detection is unavoidable due to the inherent feedback delays, estimate errors, and quantization flaws in real-world wireless systems, which cause unwarranted interference to the PLUs [4].

**Technical Gap:** To prevent frequent false detection, some researchers have proposed increasing the sensing time from a microsecond to several milliseconds [5]. However, from the SNU's perspective, the likelihood of opportunistic uses of unused spectrum significantly decreases with the increased sensing time. Moreover, a tradeoff exists between sensing time and data transmission [6]; hence, with the sensing time increase, more time is spent tracking PLU activities rather than supporting SNU transmissions, which ultimately causes SNU data transfer to be less viable. Several research studies have focused on spectrum sensing using a wide variety of white space detectors to tackle the tradeoff between sensing time and data transmission. Existing detectors can be categorized as interference-based, cooperative, energy, cyclo-stationary,

Md Mehedi Hassan Galib and Mohamed F. Younis are with the Dept of Comp. Science & Elec. Eng., Univ. of Maryland Baltimore County, Baltimore, MD 21250 USA (e-mail: mgalib1@umbc.edu, and younis@umbc.edu).

Brian W. Stevens is with Geon Technologies, LLC. Columbia, MD 21046, USA (e-mail: bsteven1@umbc.edu).

matching filter, and blind [7], [8]. In addition, hybrid detection techniques also exist, such as the delta detector [3], which combines energy and blind detection to enhance detection accuracy and reduce the time needed for sensing. Sensors are used further used to collect statistics on spectrum usage patterns; these statistics are then used to define a probabilistic model to schedule secondary transmissions [9], [10].

Meanwhile, quite a few ML methods, including K-means clustering, support vector machines, and deep neural networks, have been exploited to limit interference with PLUs and overcome the issue of hidden PLUs [11], [12], [13], [14], [15]. Moreover, ML-based feature template matching techniques have been developed to increase sensing accuracy and time; examples include grayscale-based, image correlation, edge-based, and naïve template matching [16], [17]. Other reward-based ML schemes, like reinforcement learning, have been proposed to improve spectrum usage [18]. However, the scope of these ML techniques is limited to spectrum sensing and none of them are geared for capturing the behavior of PLUs over time, which is necessary for effective resource allocation. Indeed, early prediction of white space before it becomes accessible helps in accurately allocating and distributing CRNs' resources, and SNU's can define carrier and bandwidth before availability. Indeed, early anticipation of white space maximizes the SNU's spectrum access time, and increases the likelihood of successful data transmission.

**Contribution:** To address the limitations of sensing-only based CRN, we propose a model for prediction of white space through anticipation of PLU activities using bio-inspired ST-SNN and show its advantage using LTE data collected in the field. The LTE network is pursued in the validation given its popularity and widespread adoption. ST-SNN is regarded as the third generation of neural networks by addressing the complexity of deep neural networks that deem them unsuited for edge devices, where the computational capacity is limited and energy is a prime concern. Additionally, event-driven weight calculation and discrete spikes sparsity allow ST-SNN to capture the inherent spatio-temporal dynamics of the frequency spectrum effectively. Our proposed model consists of two modules: data pre-processing and ST-SNN architecture. We utilize an energy detector to determine PLU spectrum uses. The spectrum information is then converted to event-driven spikes and fed to the ST-SNN module. The proposed ST-SNN architecture consists of a recurrent temporal network and feed-forward spectral network, capturing dynamics of spectrum usage. In summary, the contribution of the paper is follows:

- Propose an effective and efficient spectral and temporal-based prediction method to anticipate PLU access patterns and predict white space available in both domains.
- Develop a bio-inspired model to capture the underlying pattern of frequency spectrum usage.
- Demonstrate the efficiency of our proposed prediction method using field-collected LTE data.

This paper is organized as follows. Section II covers related work. Section III discusses the considered system model and provides an overview of the solution strategy. Section IV describes the proposed ST-SNN based prediction model in

detail. Section V discusses the pre-processing of LTE data, and the training of the ST-SNN model, and presents the performance results. Finally, Section VI concludes the paper and hints at future extensions.

## II. RELATED WORK

Given the design objective of the proposed ST-SNN, we cover related work that pursue only contemporary detectors or machine learning techniques for sensing and assessing white spaces. A comparative summary is provided in Table I. We also highlight the advantage of SNN over deep learning techniques.

### A. Conventional CRN Sensing

According to the IEEE 802.22 Wireless Regional Area Network standard [19], a medium access control (MAC) layer has a sensing slot and a data transmission slot, demanding periodic spectrum sensing. Hence, a tradeoff between sensing and throughput has been defined in [2], achieving maximum throughput through an energy detection scheme. Rather than using energy filters, Stotas and Nallanathan [5] have considered the average transmitted and interference power for maximizing the throughput. In [20], local spectrum sensing has been studied using several filters, such as matched filter, energy filter, and feature detector, to identify white spaces. Matched filters opt to maximize signal-to-noise ratio, where SNU's correlate the received and known PLU signals to detect the presence of PLUs. Meanwhile, in the energy detector, no prior knowledge of the PLUs is needed, where the average energy is compared with a threshold to detect PLU access. Moreover, cyclo-stationary features of the received signal are exploited to determine the PLU's presence.

However, the aforementioned local sensors faced several challenges, such as wide-band sensing, synchronization, noise uncertainty, multi-path fading, and shadowing; hence, centralized cooperative spectrum sensing was proposed. In [21], local spectrum sensing has been evaluated under shadowing, and collaborative spectrum sensing is proposed for the energy detector. Spectrum sensing error and mutual interference have been considered in [4], to support various services, namely real-time and non-real-time SNU's. Power distribution for co-operative and non-cooperative scenarios have been compared to determine the optimal sensing time by maximizing average transmission and minimizing interference [22]. In [23], the probability of white space utilization is determined under energy detector sensing flaws for diverse CRN topologies. Spectral and temporal white space detection models are devised in [24] for both cooperative and non-cooperative sensing. Again reliance on detectors limits the utilization of white space given the limited time left for SNU's to transfer their data. The proposed ST-SNN model strives to overcome such a limitation.

### B. ML-based Sensing & Prediction

Thilina et al. [13] studied the effectiveness of pattern recognition-based ML algorithms for spectrum sensing, namely, unsupervised K-means clustering, Gaussian mixture model, and supervised support vector machine, and K-nearest

TABLE I: A comparative summary of contemporary detectors and machine learning techniques for sensing and predicting white space in the spectrum.

Study	Methodology	Type of access	Approach	Prediction Basis	Spectral	Temporal
[2]	Sensing	Cooperative	Conventional detector	—	✗	✓
[4]	Sensing	Both	Conventional detector	—	✓	✗
[5]	Sensing	Opportunistic	Conventional detector	—	✓	✗
[12]	Sensing	—	ML(KNN, SVM)	—	✓	✗
[13]	Sensing	Cooperative	ML (K-mean, GMM, SVM, weighted KNN)	—	✓	✗
[14]	Predicting	Cooperative	ML(ANN, RF, SVM)	Channel	✓	✗
[15]	Sensing	Both	ML(CNN)	—	✓	✗
[18]	Predicting	Opportunistic	ML(Probabilistic)	Channel	✓	✗
[19]	Sensing	Opportunistic	Conventional detector	—	✗	✓
[20]	Sensing	Cooperative	Conventional detector	—	✗	✓
[21]	Sensing	Opportunistic	Conventional detector	—	✗	✓
[22]	Sensing	Both	Conventional detector	—	✓	✗
[23]	Sensing	Opportunistic	Conventional detector	—	✗	✓
[24]	Sensing	Both	Conventional detector	—	✓	✓
[25]	Sensing	Both	ML(Stackelberg)	—	✓	✗
[26]	Sensing	Opportunistic	ML(Markov decision process)	—	✗	✓
[27]	Sensing	Opportunistic	ML(Multi-agent reinforcement learning)	—	✗	✓
[28]	Sensing	—	ML(CNN, LSTM)	—	✗	✓
[29]	Sensing	Opportunistic	ML(Spiking deep ResNet)	—	✓	✗
[30]	Sensing	Cooperative	ML(Modified Whale optimizer with SNN)	—	✓	✗
[31]	Predicting	—	ML(Non-stationary HMM)	PLU	✗	✓
[32]	Predicting	—	ML(MLP)	Channel	✓	✗
[33]	Predicting	Both	ML(Delayed feedback reservoir)	—	✗	✓
[34]	Predicting	Cooperative	ML(LSTM)	Channel	✗	✓
[35]	Predicting	—	ML(ANN+Q-learning)	Channel	✓	✗
[36]	Predicting	Opportunistic	ML(GRU, CNN, DCG)	Channel	✓	✗
[37]	Predicting	Opportunistic	ML(2D LSTM)	Channel	✓	✓
[38]	Predicting	Both	ML(Temporal CNN)	Channel	✓	✓
<b>Our Work</b>	Predicting	Opportunistic	ML(ST-SNN)	PLU	✓	✓

neighbors. These classifiers use energy vectors estimated at SNUs to determine spectrum holes and categorize feature vectors into "channel available" and "channel unavailability" groups. Similarly, ML-based supervised, and unsupervised frameworks for CRN have been investigated in [12], where PLUs operate at several transmit power levels. After learning PLU's transmission patterns through an unsupervised K-means clustering, SNUs sense their surroundings by adopting a supervised supporting vector machine to differentiate the PLU state based on energy feature vectors. In [25], a survey has been conducted to visualize Stackelberg game applications for resource allocation for ad-hoc and heterogeneous CRNs.

A probabilistic resource allocation strategy is proposed in [18]. The probability of channel availability is derived from spectrum sensing to optimize channel and power allocation on multi-channel equipment. However, this method suffers a high rate of false detection, which ultimately causes interference with PLU signals. In [26], a Markov decision process is formulated based on delay and energy. The delay cost reflects staying idle in a time slot, the energy cost is based on spectrum sensing and data transmission. Throughput gain is defined as successful transmission. A reinforcement learning-based multi-agent system has been proposed in [27], where SNUs sense the medium and adjust the transmitted power locally based on availability and interference constraints. Moreover, Bhatti et al. [15] have conducted a comparative study among six well-known Convolutional neural networks, namely, Alexnet, VGG16, Inception V3, Squeezenet, Resnet18, and Resnet50, to classify radio technologies and their interference for spectrum monitoring. A spiking deep ResNet architecture is proposed in [29], where SNUs utilize sub-Nyquist compressive

sensing techniques to detect PLU access. A spiking neuron approach is also employed in [30] for sensing the cooperative relay spectrum. However, these spiking neurons only used the spectral or temporal aspect of the spectrum, but not both. A blind spectrum deep learning sensing method is developed in [28] using a combination of three different neural networks, namely, CNN, LSTM, and fully connected layers. None of the discussed ML-based methods so far predict the channel occupancy or PLU access patterns.

Existing predictive methods can be categorized based on what is being factored into (i) spectral, (ii) temporal, and (iii) combined. Examples in the first category include [14], where a vehicular data rate prediction method is proposed based on channel quality assessment and estimation of network load. A multi-layer perceptron-based channel prediction method is proposed in [32], where no prior knowledge about channel status is required. However, spectral-based methods fail to capture the temporal correlation aspect of channel access. Example methods in the second category include [33], where a variant of recurrent neural network (RNN), namely, a deep delayed feedback reservoir (DFR), is pursued to improve the spectrum utilization along the temporal domain. Alternatively, LSTM is used in [34]. It is to be noted that, unlike conventional non-linear neurons, we pursue bio-inspired spike neurons to model the DFR in order to leverage the underlying energy efficiency provided by spiking neurons. To support quality of service (QoS), a two-phase proactive ML-based prediction method is proposed in [35], where the relative traffic of the channels is first predicted and then user satisfaction is maximized using Q-learning. Chen et al. [31] employ non-stationary hidden Markov Model (HMM) to capture the time-

varying property of PLU access; however, unlike our work, the spectral aspect is not factored in.

Approaches in the third category consider features from both the spectral and temporal domains. For example, a combination of CNN and gated recurrent unit were employed in [36] to predict channel availability for local SNUs. A 2D LSTM is used in [37] for temporal and spectral correlation in spectrum occupancy prediction. Alternatively, a sequence-to-sequence temporal CNN (TCN) model is pursued in [38]. However, other than [31], none of the temporal models consider PLUs dynamic access pattern during prediction. Moreover, except [33], existing ML-based predictive medium access models employ conventional activation functions, which are resource intensive, unlike spiking neurons. Hence, our proposed ST-SNN model avoids these shortcomings through observing PLU access patterns along spectral and temporal domains and employing lightweight bio-inspired spike neurons for activation.

### C. Low Power ST-SNN

Although deep neural networks (DNNs) efficiently handle feature detection, image identification, and signal variation, training these models requires a lot of computational resources, time, and energy, making it impractical to use and implement them on edge devices [39]. Hence, spike neural networks have been proposed as a solution. The design of SNNs is inspired by event-driven biological information processing rather than conventional data-driven DNN [40]. Such an event-driven method biologically mimics the human brain, where the information is encoded into temporal pulse trains, which require less low-power consumption [41]. Moreover, the DNN's back-propagation learning method consumes much power. In contrast, bio-inspired spike time-dependent plasticity utilizes the Hebbian learning to update the synaptic weight among the neuron in SNN architecture, which is less computationally demanding and more power efficient. Anticipation of the PLU's behavior (access dynamics) is a two-dimensional sequential pattern, where the x-axis represents time (temporal pattern), and the y-axis demonstrates its activity (spectral information). As in the biological brain, the combination of basal ganglia, hippocampus, and prefrontal cortex forms a working memory which offers a sparse sequential pattern in a temporal sequence [42]. Hence, in [43], a spatio-temporal model has been proposed where two architectures, specifically, a recurrent spiking network to serve as a temporal backbone, and a feed-forward read-out network to capture the spatial aspects; together the two architectures formulate a working memory, analogous to a biological brain. The dynamics of the recurrent driver network were transferred to the read-out neurons to encode space by employing bio-inspired Hebbian learning to update synaptic weights. As explained in the following sections, ST-SNN is better suited for capturing PLU's dynamic access pattern. To the best of our knowledge, no prior work has considered ST-SNN in the context of CRNs.

## III. SYSTEM MODEL AND APPROACH OVERVIEW

### A. Spike Neural Network

Unlike traditional neuron models, in ST-SNN, the information is transmitted as an action potential, also known as spikes

between the layers. Several bio-inspired neuron models have been proposed in the literature [39], [44], [45], [46] to handle these spikes and emulate the biological brain. As illustrated in Fig. 1(a), when a postsynaptic neuron receives a sequence of spikes at time  $t$ , the dynamics of the membrane potential ( $V(t)$ ) changes depending on the nature of presynaptic neurons. The spikes from an excitatory neuron depolarize the membrane potential, where an inhibitory neuron's spike causes hyperpolarization, as illustrated in Fig. 1(b). Depolarization boosts  $V(t)$ , while hyperpolarization decreases  $V(t)$ . A bio-inspired neuron sums all the depolarization and hyperpolarization received from presynaptic neurons and generates a spike when the resultant summation exceeds threshold  $V_{th}$ , as shown in Fig. 1(b). It is to be noted that after generating a spike, the bio-inspired neuron enters into a temporary refractory period, where the membrane potential goes below the nominal resting potential  $V_R$ . During the refractory period, the neuron becomes irresponsible of external spikes.

To simulate the aforementioned biological process of spike generation, E. M. Izhikevich [45] proposed a simple *leaky integrate and fire* model (LIF), shown in Fig. 1(c). The physical model of LIF comprises membrane capacitance ( $C_m$ ), leaky resistance ( $R_L$ ), reversal potential ( $E$ ), presynaptic current ( $I_{presynaptic}$ ), and external current ( $I_{ex}$ ). The leaky current ( $I_L$ ) is defined as  $I_L = \frac{E-V}{R_L}$ . Moreover, the presynaptic current depends on the path resistance and membrane potential of presynaptic neurons. However, the presynaptic membrane potential depends on whether they provide excitatory or inhibitory input spikes. Now, all these currents charge the membrane's capacitance, ultimately increasing the membrane's potential. The circuit generates a spike if the membrane potential exceeds the threshold voltage ( $V_{th}$ ). Since this simple LIF model combines linear filtering of input currents with a strict voltage threshold, we adopt them to formulate the inhibitory and interneuron neuron model of ST-SNN. However, to replace strict threshold with a more realistic smooth spike initiation zone, Destexhe et al. [44] proposed intrinsic current through exponential LIF model, which is added as a  $\sum I^{int}$  branch in Fig. 1 (c). We adopt this sophisticated exponential model for read-out and supervisor neurons of ST-SNN. Moreover, adding the adaptation current  $I_A$  allows the inclusion of sub-threshold resonance, which dynamically normalizes the threshold to prevent over-fitting. Since over-fitting causes failure in the recurrent structure, we adopt this complex neuron model to formulate excitatory neurons of ST-SNN.

### B. System Model and Design Requirements

Fig. 2 captures the ST-SNN model deployment for opportunistic spectrum access in the context of CRN. We consider a wireless communication network that employs a time-based medium access arbitration, to share a single or multiple frequency channels. An example of such a network is the LTE cellular system, which is also used for evaluating the performance of our ST-SNN model. Our future research plan includes studying the ST-SNN model integration and performance with a primary 5G cellular network. The proposed ST-SNN model is confined to predicting the availability of

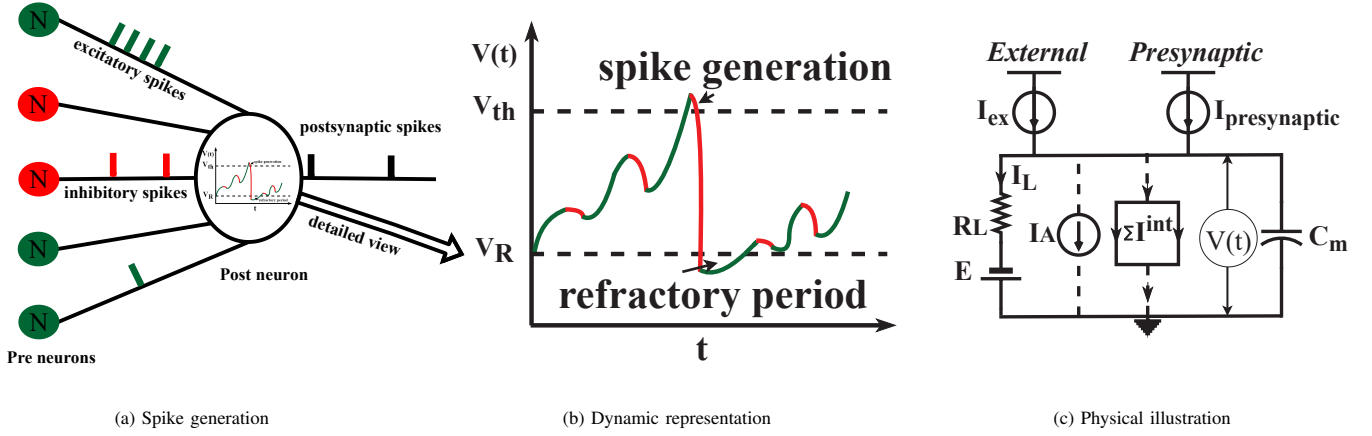


Fig. 1: Bio-inspired spiking neuron model considered for ST-SNN. (a) Representation of spike generation of a ST-SNN neuron after receiving presynaptic excitatory and inhibitory spike inputs. (b) Membrane potential variation of a bio-inspired neuron with time. Here, “green” represents depolarization caused by excitatory presynaptic inputs and “red” represents hyperpolarization caused by presynaptic inhibitory inputs. (c) Physical representation of dynamic membrane potential in terms of capacitance, resistance, and current injections.

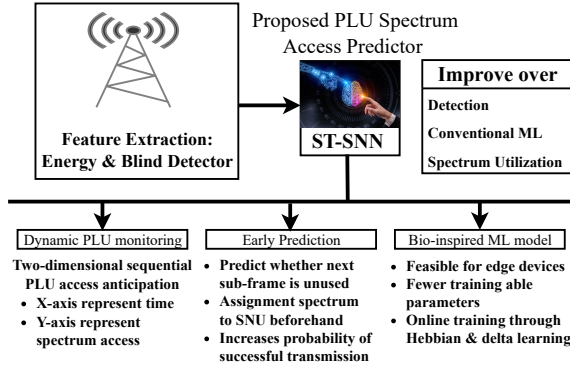


Fig. 2: ST-SNN system deployment for opportunistic spectrum access in CRN.

white space; detectors for spectrum access are still needed to confirm that an SNU can transmit without interfering with the PLU. In the validation, we consider energy and blind detectors. Nonetheless, other detectors can be used as well.

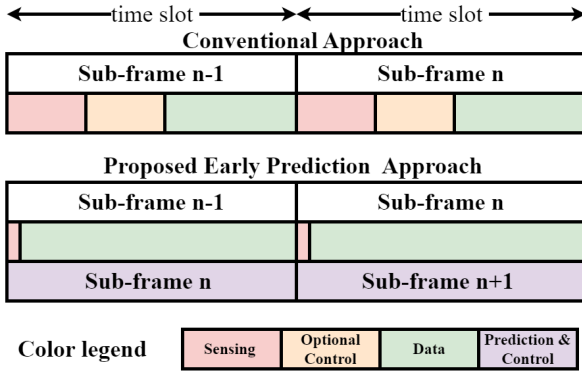


Fig. 3: Comparing the proposed approach to the conventional method for spectrum sensing, dynamic spectrum management, and data transmission.

For a cognitive wireless ecosystem, a minimum set of capabilities is required, including spectrum sensing, and adaptability through learning [47]. Yet, for modern communication in-

frastructure, prediction of white space is necessary for efficient frequency spectrum utilization. Hence, we employ a spatio-temporal window to predict white space. As illustrated in Fig. 3, conventional approaches rely only on spectrum sensing for SNU allocation. However, the sensing and configuration (reallocation) latency reduces the available time for a SNU to send data, which could lead to unsuccessful transmission. To mitigate such a shortcoming, we propose a ST-SNN model that can effectively learn and anticipate PLU dynamics.

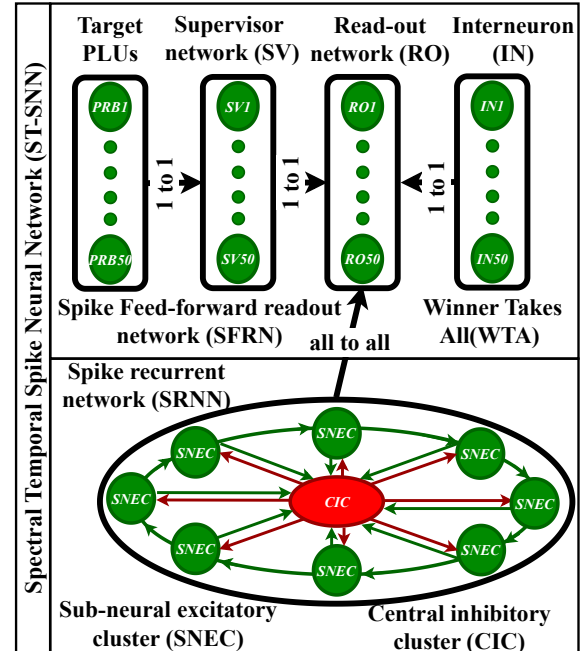


Fig. 4: Articulating the proposed ST-SNN architecture. The upper SFRN network captures PLUs dynamics while the lower SRNN anticipates their occurrence in a time scale.

Generally, an SNU can be any device. Yet, with the increased pervasiveness of wireless services and the prevalence of the internet of things, edge devices are expected to be the common SNUs. Since edge devices are typically constrained in

their onboard computational resources and energy supply, the CRN support needs to be lightweight. In that regard, ST-SNN is a prime candidate given its advantages over deep learning techniques, as noted in the previous section.

### C. Solution Strategy

Sequential behavior, which occurs due to several motor actions, is observed in the animal's daily behaviors. Learning these sequential behaviors in a neural network (NN) is very difficult due to the consideration of a single learning network that aggregates all dynamics and weights. Such difficulty is encountered because sequential learning is an array of precisely timed motor actions; these actions can be projected in the spectral domain, and the times of these actions must be mapped to the temporal domain. The dynamics and the weight update of the spectral domain differ from that of the temporal domain. However, existing NN models consider both domains in a single network and hence, fail to generate the sequential activity observed in animal behaviors. We argue that the access pattern of primary users reflects a sequential behavior, and hence needs an expressive spatio-temporal prediction model.

This paper advocates the micro-motif networks in ST-SNN as a suitable model that captures the dynamic spectrum access pattern of PLUs. These micro-motifs in ST-SNN are inspired by the biological working memory observed in the prefrontal cortex, basal ganglia, and hippocampus loop [42]. In the context of CRNs, these micro-motifs attract isolated motor actions (PLU's behavior) and map each sub-neural cluster (time scale) to a behavioral sequence to evolve across a precise time (anticipation of PLU's behavior with time). As PLU's access pattern changes with time, our proposed ST-SNN should be able to capture such change as well, which is done through adaptive neurons, time-varying synaptic dynamics, and online updating of synaptic weights. Online updating of synaptic weights is done through various plasticities, such as spike-time-dependent plasticity and homeostatic plasticity, which result in the capturing of distinct PLU's spectrum access sequence. Our ST-SNN model is illustrated in Fig. 4.

## IV. ST-SNN-BASED SPECTRUM PREDICTION MODEL

To form prediction model, we consider the following:

- *Spectral domain correlation*: In the underlay CRN strategy, the SNU uses the spectrum simultaneously with PLUs [48]. In this case, we need to monitor the energy level of each resource block to avoid inference, which could be done through spectral sensing. However, poor spectral sensing may lead to high inference and disrupt PLU communication.
- *Temporal domain correlation*: In the case of overlay CRN, SNU only transmits when PLUs are idle, which means temporal channel vacancy needs to be detected [48]. However, such a strategy leads to spectrum under-utilization, as the spectral channel access is not factored in.

Hence, to improve spectral efficiency, many researchers resort to interweaving or a mixed strategy that considers both overlay and underlay[24], [48]. ST-SNN is well-suited for such a mix strategy. Our proposed ST-SNN model combines two different

conventional neural networks, namely, recurrent neural networks and feed-forward neural networks, and employs bio-inspired features through SNN. Such a model efficiently maps temporal correlation through a spike recurrent neural network (SRNN) and spectral correlation through a spike feed-forward readout network (SFRN). Overall, our ST-SNN model consists of three parts: a temporal SRNN to provide a time scale for anticipating PLU access patterns, a spectral SFRN to anticipate PLU channel selection to avoid inference; and plasticities to define interactions within the ST-SNN architecture.

### A. Temporal SRNN Model

To formulate a time scale to capture PLU's access pattern, we employ the temporal SRNN model shown in Fig. 4. A SRNN generates a chronological sequence by activating *sub-neuronal excitatory clusters* (SNECs). Our model opts to mimic overlay functionality through sensing temporal vacancy, and is formulated as follows:

- To capture the temporal access pattern, establish recurrent dynamics inside SNEC and directional feed-forward dynamics between successive SNECs. To do so, we provide sequential external input, which forces these SNECs to form a loop centered on an *central inhibitory cluster* (CIC), as illustrated in Fig. 4. Such an inhibitory architecture prevents unstable dynamics via lateral inhibition and facilitates the formation of feed-forward connections between successive SNECs [49].
- Through online updating of homeostatic plasticity, CIC inhibits all remaining SNECs when a particular SNEC gets sequential external input. This method of providing external input helps to enhance the synaptic weight matrix of SNEC corresponding to a particular external input while nullifying the weight of other SNECs. Such a procedure is very essential to prevent desynchronization of the PLU time scale. For instance, a particular PLU could be inactive at 5<sup>th</sup> ms; however, due to desynchronization, it may be interpreted at 3<sup>rd</sup> ms, which will cause false detection at 3<sup>rd</sup> ms and interference at 5<sup>th</sup> ms.

Delivering external inputs sequentially potentiates off-diagonal synaptic weight matrix between successive SNECs following Hebbian rules [43]. Successive off-diagonal synaptic weights create a chronological structure providing a continuous time scale to monitor PLU's access pattern.

### B. Spectral SFRN Model

To mimic the underlay CRN strategy as well as to integrate the overall ST-SNN model, we adopt a SFRN model. This is the second part of our ST-SNN model, which has three different inputs, as illustrated in Fig. 4: (1) from the temporal SRNN model for sensing time vacancy through providing time scale, (2) from the supervisor network, which provides training data for spectrum vacancy or coexistence, and (3) the interneurons. The latter is to create winner-take-all (WTA) architecture, which is similar to the SoftMax layer of conventional DNN. It is to be noted that the number of read-outs (ROs), supervisors (SVs), and interneurons (INs) are dependent on the number

of primary resource blocks (PRBs). A resource block refers to a frequency channel and time-slot combination. PLUs are occupying these PRBs both spectrally and temporally. Hence, to capture the dynamics nature of PLU access, we monitor these PRBs in readout networks as follows:

- A PLU's access pattern needs to be converted to a Poisson spike, which corresponds to the sequence that the readout network should learn. Hence, in our design, the SVs of the SFRN are fed with rated spikes that are generated from the corresponding PLU state.
- A PLU state is non-Markovian sequence; for instance, from an inactive PLU's state, the next stage can be inactive/active. Hence, assessing a PLU state requires information storage over a long period, i.e., many epochs [32]. However, such storage does not constitute a challenge since it is resolved naturally through other inputs of the read-out network, such as the temporal network created by SRNN and the WTA created by interneurons. As discussed earlier, to learn data sequences, conventional ML architectures need to consider a huge weight matrix of various sequence patterns and lengths. This is a major advantage of our ST-SNN design.
- When considering the access pattern of the first PLU, the corresponding SNEC of SRNN and SV are activated simultaneously, ultimately triggering the RO neuron. At the same time, the WTA network ceases the firing of other RO neurons in the SFRN. Such an activation of SNEC, SV, RO, and IN helps store the PLU access information (sequence) in the corresponding SNEC-to-RO synaptic weight matrix. The same is applied to the remaining PLUs in the system.
- After training for several epochs, the whole model is rerun with no supervisor inputs to eliminate the noisy replica of the learned sequences through inhibitory WTA architecture and consequently increases the model's performance.

### C. ST-SNN Dynamics & Plasticities

As mentioned earlier, we need bio-inspired neuronal plasticities to handle discrete event-driven information that flows between intra-layer and inter-layer SRNN and SFRN networks. Indeed, these plasticities are essential for predicting PLU access patterns. As the dynamics of intra-layer and inter-layers differ for both SRNN and SFRN [43], [49], we need to choose a neuronal model and plasticities efficiently to preserve the information integrity. Here, we discuss the different neuronal dynamics and plasticities adopted in our design.

#### a) Neuronal model

Given the complexity of predicting a non-Markovian state (PLU access pattern), we adopt five different neuron models to capture PLU's dynamic behavior in the ST-SNN model.

(1) **SNEC Neuron:** For stimulating biological SNEC neurons in the temporal network of our ST-SNN model, we adopt an adaptive exponential LIF model as mentioned in Section III-A and shown in Fig. 1(c). Such LIF model expresses time-varying neuronal dynamics, which are essential for providing a time scale for the PLU's access pattern. The membrane dynamics  $\frac{dV}{dt}$  is calculated as follows:

$$C_m \frac{dV}{dt} = I_L + \sum I^{int} + I_{presynaptic} + I_A \quad (1)$$

where,  $C_m$  is the neuron membrane capacitance, and  $I_L$  is the leakage current, which is defined as

$$I_L = \frac{(E_L - V)}{R_L} \quad (2)$$

Here,  $E_L$  is the neuron leaky resting potential, which is different for different types of neurons and  $R_L$  is the neuron leakage resistance. The summation of intrinsic current,  $I^{int}$ , can be derived from the membrane ion's molar concentration with time [44]. To avoid the complexity of solving differential equations, in this work we use the following simplified equation for calculating  $I^{int}$ :

$$I^{int} = \frac{\delta_e \exp(\frac{V - V_{th}}{\delta_e})}{R_L} \quad (3)$$

where,  $\delta_e$  is the exponential activation window, and  $V_{th}$  is sub-threshold voltage. As mentioned in Section III-A, the  $I_{presynaptic}$  current depends on presynaptic membrane potential and path resistance, and can be defined by

$$I_{presynaptic} = \frac{E_E - V}{R_{EE}} + \frac{E_I - V}{R_{EI}} \quad (4)$$

Here,  $R_{EE}$  is the presynaptic excitatory neuron's path resistance,  $R_{EI}$  is the presynaptic inhibitory neuron's path resistance,  $E_E$  is the presynaptic excitatory neuron's reversal potential, and  $E_I$  is the presynaptic inhibitory neuron's reversal potential. Moreover, the adaptability dynamics  $\frac{dI_A}{dt}$  is to be calculated by

$$\frac{dI_A}{dt} = \frac{\alpha(V - E) - I_A}{\tau_{IA}} \quad (5)$$

where,  $\alpha$  is the sub-threshold adaptation, and  $\tau_{IA}$  is the adaptive time constant. A neuron experiences a sub-threshold increase by a factor  $A$  after each spike. Moreover, there is also increase in adaptation current as well, which is referred by  $\beta$ .

(2) **CIC Neuron:** To prevent interference and desynchronization among PLUs, we adopt inhibitory neurons to model the CIC neurons of the temporal network. As mentioned above, we chose a simple LIF neuron model with no  $I_A$  and  $\sum I^{int}$  current. The membrane dynamics ( $\frac{dV}{dt}$ ) is defined as follows.

$$C_m \frac{dV}{dt} = I_L + I_{presynaptic} \quad (6)$$

$I_L$  is defined in eq. (2) and  $I_{presynaptic}$  can be calculated by:

$$I_{presynaptic} = \frac{E_E - V}{R_{IE}} + \frac{E_I - V}{R_{II}} \quad (7)$$

Although the presynaptic excitatory and inhibitory neuron's potential is the same as in eq. (4), the path resistance for inhibitory neurons of CIC cluster are different from that of SNEC neurons. We define  $R_{IE}$  and  $R_{II}$  to represent presynaptic excitatory and inhibitory path resistances, respectively.

(3) **SV Neuron:** The SV neurons receive Poisson-encoded PLU's access sequences and propagate them to the RO network. Hence, we adopt an exponential LIF model. It is to be noted that we consider no  $I_A$  current as SV neurons only supervise RO neurons. Moreover, no loop is formulated in SV



networks, which in turn, reduces the chance of over-fitting. The membrane dynamics ( $\frac{dV}{dt}$ ) is defined as:

$$C_m \frac{dV}{dt} = I_L + \sum I^{int} + I_{presynaptic} \quad (8)$$

$I_L$  and  $\sum I^{int}$  are defined as eq. (2) and eq. (3), respectively. Since SV neurons simply relay inputs, they have no presynaptic inhibitory spikes. Hence,  $I_{presynaptic}$  is defined as

$$I_{presynaptic} = \frac{E_E - V}{R_S} \quad (9)$$

where  $R_S$  is the presynaptic excitatory input path resistance.

**(4) IN Neuron:** Like CIC, we chose inhibitory neurons to provide WTA architecture in order to suppress interference among PLUs in the spectral domain. Hence, for IN neurons we also adopt simple LIF, yet with no presynaptic inhibitory input since, unlike CIC neurons, IN neurons are discrete. The membrane dynamics ( $\frac{dV}{dt}$ ) is described as follows.

$$C_m \frac{dV}{dt} = I_L + I_{presynaptic} \quad (10)$$

$I_L$  is based on eq. (2) and  $I_{presynaptic}$  is defined as:

$$I_{presynaptic} = \frac{E_E - V}{R_H} \quad (11)$$

with  $R_H$  being the presynaptic excitatory input path resistance.

**(5) RO Neuron:** To anticipate PLU's access, RO neurons receive inputs from the SRNN, SNs, and INs. We consider an exponential LIF neuron to capture the dynamics of the read-out neuron. We note that no  $I_A$  current is considered, since RO neurons form a feed-forward architecture without forming a loop. The membrane dynamics ( $\frac{dV}{dt}$ ) is described as follows:

$$C_m \frac{dV}{dt} = I_L + \sum I^{int} + I_{presynaptic} \quad (12)$$

Like SV and SNEC neurons,  $I_L$  and  $\sum I^{int}$  are defined by eq. (2) and eq. (3), respectively. Since RO neurons receive presynaptic inputs from SNECs, SV, and IN,  $I_{presynaptic}$  is defined as

$$I_{presynaptic} = \frac{E_E - V}{R_{RT}} + \frac{E_E - V}{R_{RS}} + \frac{E_I - V}{R_{RI}} \quad (13)$$

where,  $R_{RT}$ ,  $R_{RS}$ , and  $R_{RI}$  are presynaptic input path resistances from the temporal, supervisor, and interneuron networks, respectively. As both SNECs and SVs are excitatory neurons, we consider  $E_E$  as the presynaptic excitatory membrane potential. On the contrary, since IN neurons provide WTA architecture for RO neurons, we consider presynaptic inhibitory potential  $E_I$  for IN neurons.

#### b) Synaptic Dynamics

Our ST-SNN model considers synaptic dynamics that modify the path's conductivity or resistivity over time to formulate reward-based Markov's chain policy. This means that SNUs will choose the PLUs which have more stable states (access patterns) over time. Synaptic dynamics introduce nonlinearity in the  $I_{presynaptic}$  current of the LIF neuron models to imitate biological short-term potentiation or depression. The dynamics

for presynaptic path resistance of eq. (4), (7), (9), (11), and (13) are expressed as follows:

$$R_{XY} = \frac{1}{\frac{e^{-\frac{t}{\tau_d^Y}} - e^{-\frac{t}{\tau_r^X}}}{\tau_d^Y - \tau_r^X} * \left( W_{ex}^X I_{ex}^X + \sum W_{XY}(t) I(t) \right)} \quad (14)$$

with  $X$  and  $Y$  denoting the postsynaptic neuron, and a presynaptic neuron, respectively.  $I_{ex}^X$  and  $W_{ex}^X$  are constant biasing current and weight for  $X$ .  $W_{XY}$  is the weight between  $Y$  and  $X$ , while  $\tau_d^Y$  is the decay time for the synaptic dynamics of  $Y$ , and  $\tau_r^X$  is the rise time for the synaptic dynamics of  $X$ .

#### c) Synaptic Plasticity

As stated earlier, to capture the dynamic nature of PLU access, our ST-SNN model should update the synaptic weight based on activity-dependent plasticity. To do so, we incorporate voltage-based spike time-dependent plasticity (STDP) to update the excitatory synaptic weight of ST-SNN model. On the contrary, homeostatic plastic is adopted for inhibitory to excitatory weight update. Moreover, we employ delta learning to synaptic weights between SV and RO neurons.

**(i) Voltage-based STDP:** As PLU's access changes with time, we adopt the STDP model to update synaptic weight rather than applying popular back-propagation algorithms. The use of STDP eases the computational complexity by employing a biologically plausible weight update algorithm such as Hebbian learning to control synaptic weights. Thus, for updating a synaptic weight, anticipating presynaptic and postsynaptic neuron behavior is crucial for STDP.

In STDP, long-term potential (LTP) refers to strengthening the synaptic weight. LTP occurs when a presynaptic neuron firing happens before a postsynaptic neuron firing. Conversely, long-term depression (LTD) reduces the synapse connection between neurons. LTD occurs when a postsynaptic neuron activates before a presynaptic neuron fires. A voltage-based STDP ensures proper stabilization of all synaptic weights compared with other STDPs. Hence, we adopt a voltage-based STDP to update synaptic weights between excitatory neurons in our ST-SNN model; such STDP is defined as follows:

$$\begin{aligned} \frac{dW}{dt} = & A_{LTP} x_j(t) F(V_i(t) - \Delta_{LTP}) F(y_i(t) - \Delta_{LTD}) \\ & - A_{LTD} I(t) F(z_i(t) - \Delta_{LTD}) \end{aligned} \quad (15)$$

Here,  $x$  is a low-pass filtered version of the input, while  $y$  and  $z$  are low-pass filtered versions of postsynaptic voltages.  $A_{LTP}$  and  $A_{LTD}$  are the amplitudes and  $\Delta_{LTP}$  and  $\Delta_{LTD}$  are the thresholds of long-term potentiation and depression, respectively, and  $F(\cdot)$  signifies the activation function.

**(ii) Homeostatic plasticity:** To prevent interference among different PLU's, we adopt homeostatic plasticity to update inhibitory to excitatory synaptic weights, which only activate the corresponding PLU and inhibit others. Like the SoftMax of NN, this plasticity helps to identify the relevant parameters and nullify the unstable dynamics through lateral inhibition that could be introduced during the training period. The

homeostatic plasticity considered for this work is described as follows:

$$\frac{dW}{dt} = A_{inh}(v(t) - 2r_0\tau)I(t) + A_{inh}u(t)v(t) \quad (16)$$

Here,  $u$  and  $v$  are low-pass filtered versions of the postsynaptic voltages, and  $A_{inh}$  and  $\tau$  are the amplitude and threshold for the rated input  $r_0$ , respectively.

(iii) **Delta Learning:** In a supervised bio-inspired network, the delta learning rule is a back-propagation technique that helps in refining synaptic connections in a readout network [42]. In the context of PLU access prediction, the delta rule helps in updating the synaptic connection between the supervisor and readout neurons by reflecting the prediction error. The delta learning is defined as follows:

$$W = \alpha(PLU_{Predict} - PLU_{target}) \quad (17)$$

Here,  $PLU_{Predict}$  is the prediction of white space by the ST-SNN model in the previous sub-frame, where “0” means it predicts white space and “1” reflects anticipation of PLU access. Meanwhile,  $PLU_{target}$  is the actual PLU access that the energy/blind detector infers. Unlike  $PLU_{Predict}$ ,  $PLU_{target}$  is detected in the current sub-frame, which also has two possible values: “0” for white space and “1” for PLU access. For correct prediction, i.e.,  $PLU_{Predict}$  and  $PLU_{target}$  are the same,  $W = 0$ , meaning no change in weight (synaptic connection) is needed. In case of wrong prediction, we need to adjust the dynamics of the synaptic connections. For instance, if the ST-SNN predicts white space but the energy detector points out PLU access, the synaptic weight is decreased by  $-\alpha$ , meaning the next time it will not predict white space for similar scenarios to avoid interference. On the other hand, if the ST-SNN model predicts active PLU while there is white space, the synaptic weight is increased by  $\alpha$  to predict the white space correctly and increase the overall spectrum usage. It is to be noted that we consider  $\alpha = 0.4$  [42].

## V. RESULTS & DISCUSSION

We have validated the performance of our ST-SNN model for spectrum usage prediction using data from a commercial LTE network. Such a network belongs to Verizon, which is a prominent provider of cell services. The data was collected from antennas on Base-station located in the suburban area of Elkridge, MD, USA. We have selected this specified location and carriers to ensure an acceptable signal-to-interference-plus-noise ratio. Moreover, we have sustained system synchronization using the time source to discipline clocks and maintained a fixed timing alignment from a GSPDO (global positioning system disciplined oscillator). We note that these methods can be used for either FDD or TDD. More specifically, for TDD, in order to prevent time error as well as achieve tight synchronization, there are mechanisms in place to understand the DL/UL ratio so that downlink and uplink monitoring can be preserved. For instance, to align UL packets for both FDD and TDD, a cyclic prefix correlation can be used on each packet. We note that the UL/DL configuration is in SIB1 (System Information Block 1), which should be easily intercepted because it occurs every other frame and is

TABLE II: Detailed Information about Collected Signal

Cell ID	80	Name	700c
Center Frequency	751 MHz	Mode	FDD
Bandwidth	10 MHz	Band	13
Downlink	751.00 MHz	Earfcn DL	5230
Uplink	782.00 MHz	Earfcn UL	23230

broadcasted data (not encrypted) [50]. More information about the collected signals is provided in Table II. The data reflects continuous monitoring of PLU access patterns over 15.8 K frames and helps to visualize the spectrum thronging and white space.

We use MATLAB r2022a to evaluate the performance. The collected data reflects the dynamic behavior of 50 different *Physical Resources Blocks* (PRBs), which can be accessed both spectrally and temporally by different PLUs. Hence, in the ST-SNN implementation we consider 50 neurons for each of the read-out, supervisor, and inter-neuron network. It is to be noted that a read-out neuron represents one PRB; through the anticipation utilization of PRBs, we decode PLUs access patterns and predict white space in the spectrum. We note that the number of PRBs can be inferred from the LTE base-station control frame, as explained below.

### A. Implementation Details

#### 1) Pre-processing of LTE Data

**Base-station Signal Pre-processing:** After collecting live data from the base-station, we have performed demodulation to find medium access reservations. In an LTE-specified reservation, a sub-frame comprises 14 symbols with 12 sub-carriers for each PRB. Hence, the demodulated signal, which originally reflects  $600 \times 140$  blocks, is mapped to 50 PRBs with 10 sub-frames for each of them, as illustrated in Fig. 5(a). Since each sub-frame is 1 ms in timescale, a frame is 10 ms wide. As illustrated in Fig. 5(b), *channel status indicator* (CSI) usually occupies 1-3 symbols in a sub-frame. Therefore, the data transmission slot per sub-frame varies from 13-11 symbols. However, decreasing the CSI size will increase the probability of false detection [3]. While there are other symbols such as reference symbols and other channel symbols, we assume that CSI glosses over them.

**White Space Detection:** Although any detection mechanism can be pursued to determine white spaces in the frequency spectrum, we adopt the energy detector. Such a detector monitors the energy of the fourth symbol of each sub-frame to check for PLU medium access, as illustrated in Fig. 5(c). The white space profile provided by the energy detector for the considered dataset is shown in 5(d).

**Pattern in collected data:** As our proposed ST-SNN model anticipates PLU access both spectrally and temporally, we have checked whether our collected data exhibits any patterns. For that, we have analyzed the overall utilization of PRBs over time, and calculated the average allocated PRBs per sub-frame. The results are shown in Fig. 6, where the utilization of PRBs over the sub-frame is observed to be almost uniform, with about 5% of average variation from the mean (32 PRBs).

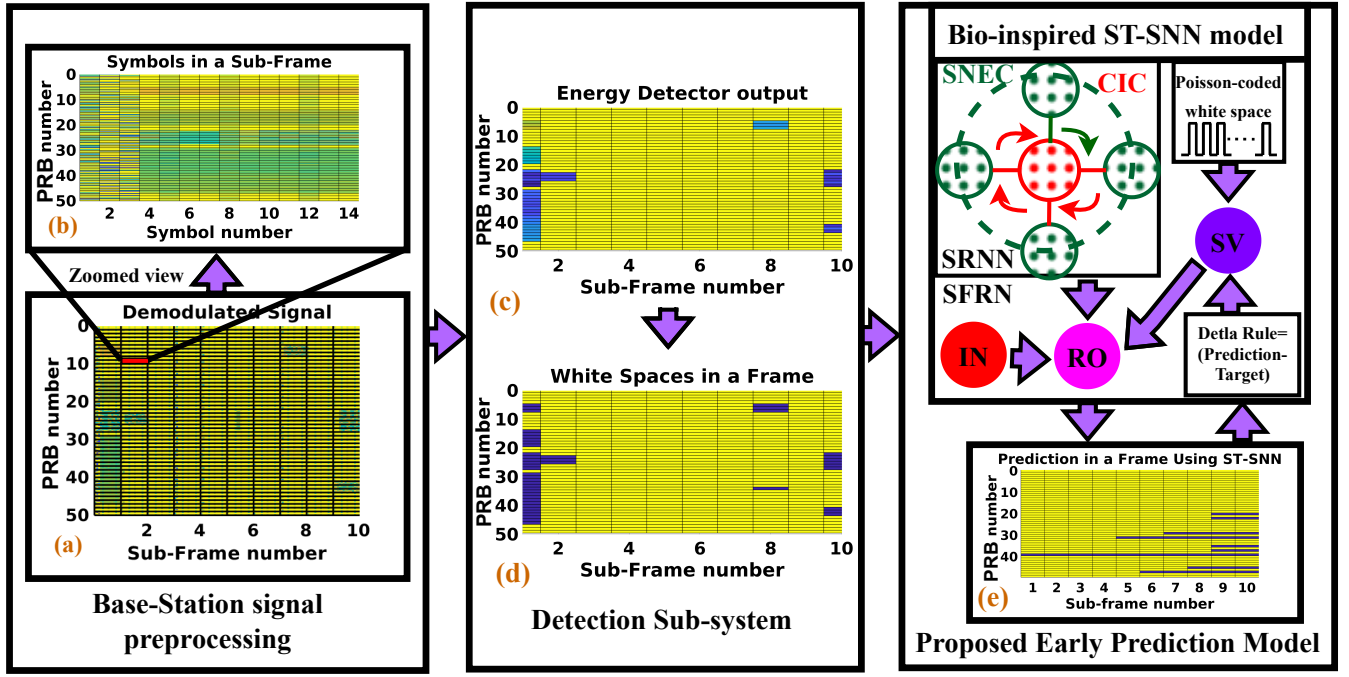


Fig. 5: An overview of the processing of live LTE data and the training of our proposed ST-SNN model. Part (a) represents an LTE frame, where the x-axis shows the number of sub-frames and y-axis reflects the number of resource blocks in the frequency domain. Each small rectangle denotes a resource block and contains resource elements, which look like granular pixels. A zoomed version of this resource block (sub-frame) is depicted in part (b), which reflects the energy level of the resource elements in such a sub-frame. The x-axis in part (b) represents the number of symbols while the y-axis corresponds to sub-carriers. The first three symbols of a sub-frame have random colors and are referred to as the control channels; the remaining symbols almost have the same color and are called the shared channels. The colors reflect the energy levels. Except control channels, the energy level is almost the same for the shared channel in the same row. Hence, by measuring the energy level of any of the shared symbols (say symbol no 4), we can detect the presence of user equipment. Averaging the energy of shared channels in a sub-frame (resource block) is shown in part (c), which is the output of an energy detector. Later, we pass through an energy threshold to generate a truth map indicating whether the resource block is used by the user equipment, or it is a white space. Part (d) shows such a map, where dark blue represents the white space and yellow represent the occupied resource block. In part (e), we represent how our proposed bio-inspired ST-SNN model learns changes in the resource block status over time.

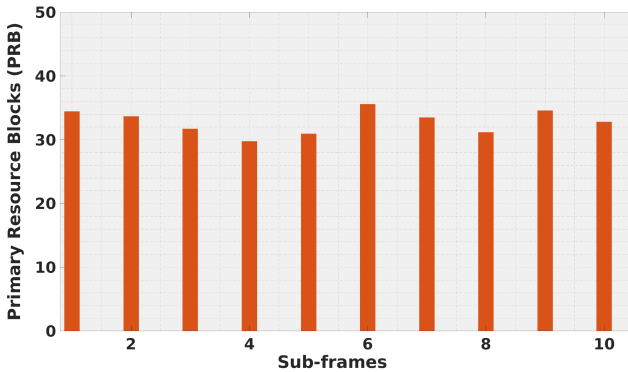


Fig. 6: The average number of PRBs used in each sub-frame in the collected data. Around 5% variation, on average, is observed among the sub-frames, indicating almost uniform utilization of the resource blocks.

Hence, there exist patterns in our collected data that suffice for the implementation of a predictive model.

## 2) ST-SNN Model Implementation

**Model Architecture:** The implementation of our ST-SNN model is illustrated in Fig. 5. For the SRNN part, we chose 8K excitatory neurons to form SNECs, and 2K inhibitory neurons to form CIC. These neurons are connected randomly with a probability of 0.2. The data reflects 50 PRBs and

hence we chose 50 RO neurons, 50 SVs, and 50 INs to form the spectral networks. These ROs, SVs, and INs are connected one-to-one, except the SNECs of the SRNN, which connect all-to-all with RO neurons. In Fig. 5, we show one RO connection with the corresponding SN, IN, and SNECs. SN uses Poisson's encoded sub-frame white-space generated in Fig 5(d), whereas RO predicts the corresponding PRB allocation dynamics. Based on the delta rule, i.e., eq. (17), the difference between the predicted and actual value is used to update the SV's interpretation of the targeted sequence.

**Training Parameters:** To capture both temporal and spectral vacancies in the spectrum, our proposed ST-SNN model deploys two different SNN models, specifically, a temporal SRNN and a spectral SRFN, where the neuronal dynamics and plasticities vary. Hence, we resort to different neuronal, synaptic dynamics, and synaptic plasticity parameters to model these temporal SRNN and spectral SRFN.

(i) **SRNN parameters:** It has been shown in [43], [49] that maintaining a 4:1 ratio of excitatory to inhibitory neurons, would help in building the recurrent dynamics around the inhibitory cluster. We have experimented with different excitatory neuron counts, subject to the aforementioned excitatory to inhibitory neuron ratio. Our experiments have concluded that employing 8K excitatory SNEC neurons and 2K CIC neurons suits our SRNN model and provides the required time

scale for capturing temporal vacancies. In addition to neuron count setting, Table III lists additional parameters and synaptic weight values [43], [51]. There are two noteworthy settings in Table III. First, the synaptic weights of inhibitory connections are negative, showing the inhibition to the membrane potential, which is required to prevent a neuron from misleadingly firing. Second, the excitatory-excitatory synaptic weight is not initialized to zero, specifically, with a non-zero value with probability 0.2. This is because assigning a zero value will fix the RNN topology as the synaptic weights that are zero remain zeros [43], [51], [52].

TABLE III: Parameter settings for evaluating the SRNN model, where E and I refer to excitatory and inhibitory neurons, respectively.

Description	Symbols	Value
SNEC(E neurons in SRNN)	-	8000
CIC(I neurons in SRNN)	-	2000
Recurrent connection probability in SRNN	$p$	0.2
Initial E-to-E connection in SRNN	$W_{EE}$	$2.83 \text{ pF}$
E-to-I connection in SRNN	$W_{IE}$	$1.96 \text{ pF}$
I-to-E connection in SRNN	$W_{EI}$	$-62.87 \text{ pF}$
I-to-I connection in SRNN	$W_{II}$	$-20.91 \text{ pF}$

(ii) SRFN parameters: Since the used LTE dataset reflects 50 PRBs, we chose 50 SVs, 50 ROs, and 50 INs to build the SRFN neurons. The chosen parameter settings for initial synaptic connections are listed in Table IV [43]. To create WTA architecture, negative synaptic weights are initially assigned for readout-to-interneuron neurons. Finally, we note that in essence Tables III and IV enlist hyper-parameters, whose values are chosen through several iterations. These values could differ based on the ST-SNN design objectives.

TABLE IV: Parameter settings for evaluating the SRFN model, where S denotes supervisor neurons, R means readout neuron and In means interneurons.

Description	Symbols	Value
SV (supervisor neurons in SRFN)	-	50
IN (interneurons in SRFN)	-	50
RO (readout neurons in SRFN)	-	50
Initial S-to-R connection in SRFN	$W_{RS}$	$200 \text{ pF}$
In-to-R connection in SRFN	$W_{RI}$	$200 \text{ pF}$
Initial R-to-I connection in SRFN	$W_{HI}$	$-200 \text{ pF}$
Initial SNECs-to-R connection in SRFN	$W_{RT}$	$2.81 \text{ pF}$

TABLE V: The parameter settings used in neuronal dynamics.

Description	Symbol	Value
Membrane Capacitance	$C_m$	$300 \text{ pF}$
Exponential activation slope	$\delta_e$	$2 \text{ mV}$
Leaky resistance	$R_L$	$66.67 \text{ M}\Omega$
Excitatory reversal potential	$E_E$	$0 \text{ mV}$
Inhibitory reversal potential	$E_I$	$-75 \text{ mV}$
Membrane potential threshold	$V_{th}$	$-52 \text{ mV}$
Sub-threshold adaptation	$\alpha$	$4 \text{ nS}$
Adaptive current time constant	$\tau_{IA}$	$100 \text{ ms}$
Adaptive threshold increase constant	$A$	$10 \text{ mV}$
Adaptive current increase constant	$\beta$	$1000 \text{ pA}$

(iii) Neuronal dynamics parameters: As discussed earlier, we chose five different types of neurons for modeling ST-SNN. The parameters for these neurons are listed in Table V [51]. Unlike conventional activation functions, in bio-inspired neurons, researchers observed changes in membrane potential to determine whether a neuron fires or not. Hence,

the parameters are defined in terms of electrical circuitry to mimic an actual biological neuron.

Now, in order to maintain the same excitatory and inhibitory time constant of  $20 \text{ ms}$  [43] for both excitatory and inhibitory neurons, we calculate the leaky resistance of membrane potential as  $66.67 \text{ M}\Omega$ . This is required to maintain the resting potential ( $E$ ) for all neurons, specifically  $-70 \text{ V}$  for excitatory neurons and  $-62 \text{ V}$  for inhibitory neurons. When the resting potential of the neural membrane exceeds the threshold  $V_{th}$  (Fig. 1(b)), a spike is generated in our case. After generating a spike, all neurons are set to a reset voltage. All neurons have the same reset voltage of  $-60 \text{ mV}$ . Along with reset, all neurons have an absolute refractory period (Fig. 1(b)), which is  $5 \text{ ms}$  for the SRNN network and  $1 \text{ ms}$  for the SRFN network. During this refractory period, the neuron is inactive, meaning it will not produce any spikes.

TABLE VI: The parameter settings for synaptic dynamics.

Description	Symbol	Value
Decay time constant (excitatory)	$\tau_d^E$	$6 \text{ ms}$
Decay time constant (inhibitory)	$\tau_d^I$	$2 \text{ ms}$
Rise time constant (excitatory)	$\tau_r^E$	$1 \text{ ms}$
Rise time constant (inhibitory)	$\tau_r^I$	$0.5 \text{ ms}$
External input rate (SNEC)	$I_{ex}^{SNEC}$	$4.5 \text{ KHz}$
External input rate (CIC)	$I_{ex}^{CIC}$	$2.5 \text{ KHz}$
External input rate (interneuron)	$I_{ex}^H$	$1.0 \text{ KHz}$
External input synaptic weight	$W_{ex}^X$	$1.6 \text{ PF}$

(iv) Synaptic dynamics parameters: To ensure the synchrony of the SRNN and SRFN with external output, we need synaptic dynamics. Such dynamics introduce nonlinearity in the neuron model. In Table VI, we list the synaptic dynamics parameters for our ST-SNN model [43]. In the table, the external input rate helps to generate the corresponding spike train with the help of the Poisson process. Hence, we use current in the eq. (14). These Poisson rates are adjusted in order to learn different PLU access patterns.

TABLE VII: The parameter settings used in Voltage-based STDP. Here, we use LPF for low pass filter and TC for time constant.

Description	Symbol	Value
LTD amplitude	$A_{LTD}$	$.0020 \text{ pAmV}^{-1}$
LTP amplitude	$A_{LTP}$	$.0014 \text{ pAmV}^{-1}$
LTD threshold	$\Delta_{LTD}$	$-70 \text{ mV}$
LTP threshold	$\Delta_{LTP}$	$-49 \text{ mV}$
TC for LPF (postsynaptic LTD)	$\tau_y$	$10 \text{ ms}$
TC for LPF (postsynaptic LTP)	$\tau_z$	$7 \text{ ms}$
TC for LPF (presynaptic x), x in SRNN	$\tau_x^{SNEC}$	$3.5 \text{ ms}$
TC for LPF (presynaptic x), x in SRFN	$\tau_x^R$	$5 \text{ ms}$

(v) Synaptic and homeostatic plasticities: For voltage-based STDP, we consider several design parameters as listed in Table VII. In order to get the presynaptic input  $x$ , and postsynaptic outputs  $y$  and  $z$  of the eq. (15), we use the low-pass filter time constant listed in Table VII. Similarly, for homeostatic plasticity, the parameters are listed in Table VIII. It is to be noted that Tables V, VI, VII, and VIII are bio-inspired SNN parameters, which are mostly adopted from [43], [49], [51], [52].

### 3) Evaluation Metrics and Baseline

Since this work promotes a predictive model for white space availability, we compare the performance with the

TABLE VIII: The parameter settings used in homeostatic plasticity. Here, we use LPF for low pass filter and TC for time constant.

Description	Symbol	Value
Amplitude	$A_{inh}$	$10^{-5} \text{ AHz}$
Target firing rate	$r_0$	$3 \text{ Hz}$
TC for low-pass filter	$\tau_{r0}$	$20 \text{ ms}$

conventional approach which relies only on detectors. Hence, we have chosen the baseline detection sub-system with energy and blind detectors for comparison. Moreover, we evaluate the advantages of the bio-inspired ST-SNN over competing ML-based predictive methods. We consider a conventional (1-dimensional) LSTM [37], 2-dimensional LSTM [37], and Temporal Convolutional Network (TCN) [38] as alternative options since: (i) 1-D LSTM is known to be effective in dealing with dynamic applications and hence suits the time series nature of PLU's medium access, (ii) 2-D LSTM can capture correlation over-time and infer PLU's access pattern, and (iii) TCN also is effective in detecting access patterns, especially for multi-attribute time series data [53]. Moreover, these ML models are simple to implement and have few trainable parameters compared to a Convolutional Neural Network (CNN) models, and can thus be executed on SNU-based edge devices.

To demonstrate the superiority of our proposed predictive model over the detection system, we examine the maximum number of symbols that an SNU may transmit successfully within a sub-frame. Moreover, we evaluate its precision, recall, F1-score, and confusion matrix. To gauge suitability for edge devices, we compare the computational complexity of our ST-SNN to that of the aforementioned competing models.

#### 4) Experiment Setup and Parameters

As mentioned above, our dataset reflects the access pattern of 50 different PRBs over 15.8K frames. As each frame has 10 sub-frames, we have a total of 158K sub-frames. Moreover, since each sub-frame represents  $1 \text{ ms}$ , we need to emulate the temporal network for  $1.58 \times 10^5 \text{ ms}$  to capture the dynamic access pattern of PLUs. As aforementioned, we chose 50 RO, SV, and IN neurons in the spectral network to represent each distinct PRBs. We provide spectral information of each sub-frame as an input to SV neurons. It is to be noted that in LTE each PRB has 12 sub-carriers; however, since we encode information using a Poisson sequence, we only need one SV per PRB. Moreover, the 14 symbols of each sub-frame are represented in the temporal network by  $1 \text{ ms}$  slot length. Thus, ST-SNN captures  $(50 \times 12) \times (158K \times 14)$ , i.e.,  $(600 \times 2.2M)$ , symbol spaces by only emulating the temporal network of  $1.58 \times 10^5 \text{ ms}$  across 50 different RO neurons for each epoch. On the contrary, other ML methods would need  $12 \times 50 \times 14$  features, which emulate over 158K time steps for each epoch.

#### B. Experimental Results

Out of the  $7.9M$  ( $50 \times 158K$ ) available spaces, PLUs are active in  $5.01M$ , and the rest,  $2.8M$ , are white spaces where PLUs are inactive. This implies that  $39.2M$  ( $2.8M \times 14$ ) is the number of available symbol spaces in the dataset, which can be used opportunistically by the SNU for their

TABLE IX: Effective data-transfer slots identified by detectors and predictor at 10 dB SNR.

Method	Avail. Data Xfer Symb	White Space Accuracy	Effective Data Xfer Symbols
	Sub-frame Data-Set		
Energy Detector	8	23.1 M	99.96%
Blind Detector	8-11	27.4 M	98.91%
ST-SNN Predictor	13	37.5 M	96.6%

data transmission. Since traditional detectors use some of these available symbols to check PLU access, fewer symbols could be allocated to SNUs, and the overall white space utilization is significantly less than the availability. Hence, we report white space utilization among energy detectors, blind detection, and our proposed predictive model.

#### 1) White Space Utilization

Table IX captures the successful utilization of white spaces by secondary users. The effectiveness of our predictive model is compared with detection-only methods. First we compare the available sub-frames and potential secondary data transfer in terms of symbol count. As mentioned earlier, the energy-based detector checks the fourth symbol in each sub-frame to determine the presence of PLUs. If no PLU is detected, CRN usually takes space for two symbols to assign opportunistic access to SNUs. Therefore, the available slots for sending the data for an SNU with energy-based detectors are usually 8 symbols, starting from the seventh to the fourteenth. Since the blind detector checks the second symbol up to the fourth symbol, more symbols could be made available, specifically up to  $27.4M$  for the considered dataset. Both energy and blind detectors perform very well for higher SNR values [3]. Given the focus on the effect of access predictability, we choose SNR value about  $10 \text{ dB}$  to ensure the effectiveness of the detection sub-system. Hence, energy and blind detectors yield around 99.96% and 98.91% white-space detection accuracy, respectively. Consequently, the energy detector's effective data transfer slots are approximately  $23.0M$  symbols per PLU in our collected data. Meanwhile, the blind detector allows  $27.1M$  symbols for SNU access, which is approximately 17% greater than the energy detector.

On the other hand, the ST-SNN model predicts the PLU behavior before the start of a sub-frame and allows the SNU to transfer data if the PLU is not predicted to be active. If the prediction is accurate, SNU immediately starts transferring the data from the second symbol, which leaves up to 13 symbols for data. Therefore, the available data transfer slots are approximately  $37.5M$  symbols for the ST-SNN model. However, if our the prediction is wrong, then there will be an inference with PLU. As ST-SNN model predicts in the previous sub-frame, it has lower accuracy than the detector as illustrated in Table III. Hence, we adopt a detector to detect the presence of PLU in the current sub-frame and provide feedback to the predictor through delta learning. Such a method of adding a detector in ST-SNN model has two advantages: first, it will avoid interference with the PLU if our prediction is wrong, and second, it will improve the accuracy



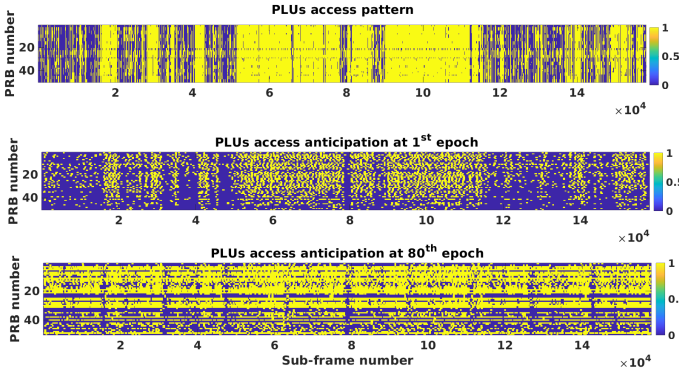


Fig. 7: PLU's dynamics behavior anticipation by our proposed ST-SNN model.

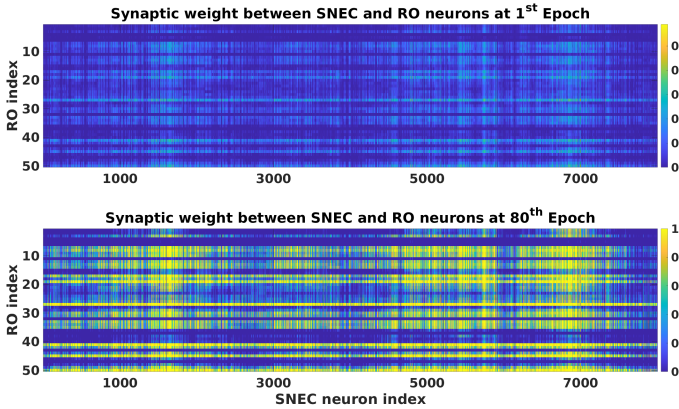


Fig. 8: Learning synaptic weights between SNEC and RO neurons over time.

of our predictor. In the case of a wrong prediction, to avoid inference with PLU, we revoke SNU to utilize the slot and penalize the predictor through delta learning. It is to be noted that we only use an energy detector to avoid interference and improve the accuracy of our predictor. The timing or the effective data transfer slots of the predictor are independent of the detector. As stated earlier, although the prediction accuracy of ST-SNN is lower than conventional detectors, it is quite high compared with other conventional machine learning predictors [14], and consequently, the overall effective data slots are around  $36.2M$  symbols. Hence, the ST-SNN model enables the use of more symbols within the available white spaces. Table IX shows that the ST-SNN model achieves about 57% and 34% improvement over the energy and blind detectors, respectively.

### 2) PLU Access Prediction Accuracy

Fig. 7 shows the ST-SNN output for the first and last epochs with Poisson's encoded targeted sub-frame to anticipate the dynamically changing access pattern of the considered 50 PRBs over  $1.58 \times 10^5$  ms. As shown in Fig. 7, at the first training epoch, the ST-SNN model fails to effectively predict the behavior of PLUs, which can be attributed to the existence of unsaturated synaptic weight between SNEC and RO neurons, as indicated in Fig. 8. Indeed, the unsaturated weight is responsible for an inadequate postsynaptic spike, which causes interference and synchronization error among

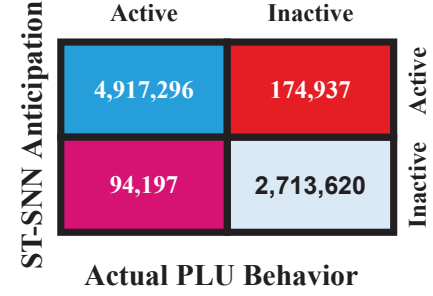


Fig. 9: The confusion matrix showing the ST-SNN anticipation over actual PLU access.

PLUs. Hence, the difference between the actual access pattern and the predicted one for such a sub-frame is relatively high, which degrades accuracy. Consequently, the ST-SNN demonstrates an accuracy of 51.75% with root means square error (RMSE) of 0.4835 at the first epoch.

Nevertheless, with the increased epoch count, the ST-SNN model learns the PLU's access pattern as illustrated in Fig. 7. Such a way of capturing PLU access dynamics is possible through the learning synaptic weight between RO and SNEC neurons through STDP, homeostatic plasticity, and delta learning. Hence, the difference between ST-SNN predicted PLU's access pattern and actual one is comparatively less at 80 epoch, which in turn elevates the accuracy of ST-SNN to about 96.6% with an RMSE of 0.4148. Although the behavior of PLUs is stochastic, the RMSE achieved by our ST-SNN model is 90% less than that of the cooperative prediction approach of [14], which was 3.90.

### 3) ST-SNN Performance

Fig. 9 shows the confusion matrix of the ST-SNN model. As aforementioned, we found that out of the  $7.9M$  resource blocks in our collected data, PLUs were operational in  $5.01M$  blocks, i.e., 63.4% of the original data. Hence, white spaces account for the remaining 36.5%, specifically,  $2.88M$  resource blocks. Fig. 9 indicates that the ST-SNN model successfully detected  $2.71M$  white spaces by accurately anticipating when PLUs were inactive. However, out of the  $269K$  wrong predictions, only  $94K$  times the ST-SNN model falsely detected PLUs inactive while they were indeed active. On the other hand, ST-SNN wrongfully deemed PLUs as active while there were white spaces available for  $174K$  times. The missed white spaces could have been assigned to SNUs. Hence, to evaluate the ST-SNN performance, we calculate precision and recall along with accuracy. It is to be noted that, in our case, both precision and recall are equally crucial, as maximizing recall will avoid interference with PLUs. In contrast, growing the precision will ensure proper utilization of white space. For our ST-SNN model, the observed accuracy is 96.6%, and the precision and recall for predicting PLU access are 96.56% and 98.12%, respectively. Moreover, the F1-score for our model is 0.97, as illustrated in Fig. 10. These values imply the viability of the ST-SNN model.

Additionally, we also perform experiments with varying the number of RO neurons in the SFRN network. As stated earlier,

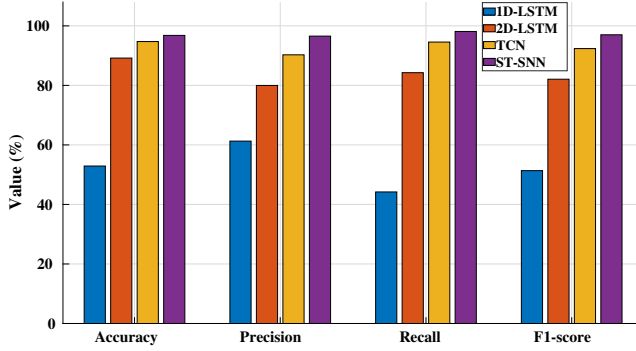


Fig. 10: Comparing the accuracy, precision, recall and F1-score of our ST-SNN model to alternative LSTM-based models [37] and TCN [38].

TABLE X: Implementation complexity of ST-SNN and other baseline models.

Method	Trainable Parameters	Accuracy	FLOPS	Activation
LSTM (1-D) [37]	215.1 M	52.91%	307.53 G	Sigmoid, tanh
LSTM (2-D) [37]	273.8 M	89.18%	424.32 G	Sigmoid, tanh
TCN [38]	42.1 M	94.73%	86.81 G	Relu
ST-SNN Predictor	16.4 M	96.6%	21.78 G	Leaky integrate & fire

we adapt RO neurons to capture spectral vacancies. Hence, we want to show the effect of changing the number of RO neurons on overall spectral utilization. As we consider 50 neurons to mimic the change in activity of 50 PRBs in our collected dataset, we change these numbers to 20 and 60. For the case of 20 RO neurons, our SRFN architecture will have 20 SV neurons as well as 20 IN neurons, which trims the number of parameters from 16.4 M to 16.1 M. However, 20 SVs will only be able to capture the PLU access pattern of the first 20 PRBs, and consequently decreases the utilization rate to 38.12%, meaning that out of 2.88M white spaces, we are only able to use 1.1M. For the case of 60 RO neurons, the SRFN architecture will have an increased number of SVs, INs, and ROs, which increases the number of parameters to 16.48 M; however, the utilization rate will be constant as the first 50 SVs track the access patterns of 50 PRBs. Hence, we should have atleast equal number of ROs to PRBs in order to achieve the highest white space utilization.

### C. Comparison with Competing Approaches

We compare our ST-SNN model with three different models, described as follows:

- We have built a 1-D stacked LSTM model with three layers (two hidden layer and one output flatten layers) having a 158K time-step, along with 8400 features as input. These features reflect 50 PRBs, 12 sub-carriers per PRB, 14 symbols per sub-frame. Each hidden LSTM layer consists of 500 units with sigmoid and hyperbolic tangents as activation functions, which is slightly different from [37]. After the last LSTM hidden layer, we use three fully connected layers of

size 400, 120, and 50 to predict white space. Moreover, we use adaptive momentum estimation (ADAM) as optimizer and learning rate 0.001 for 200 epochs.

- A 2-D stacked LSTM model is built with three layers, where the first two are hidden layers with 500 units and the last one is a flatten output layer with the same number of units for the above 1-D model. Moreover, we use the same optimizer, epochs, and learning rate as in 1-D. However, the difference between 1-D and 2-D models is in the input data. In the case of 1-D LSTM, we are only able to provide temporal data as input, whereas we use both time and frequency domain data to the 2-D LSTM, which is similar to [37].
- For implementing TCN, we resort to the parameters listed in [38], [53]. We use three stacked layers containing 32 residual blocks, where we have an increased number of dilation factors, where  $d = 1, 2, 4, 8, 16, 32, 64$ , and 128 for casual convolution. Moreover, we use 32 as the number of channels, 2 as the kernel size, ADAM as the optimizer, and 0.001 as the learning rate. More details about TCN implementation can be found in [53]. After the last stacked layer, we employ an output flatten fully collected layer with the size of 400, 120, and 50 to predict the white space.

The accuracy, precision, recall, and F1 score for these ML models are compared with our proposed ST-SNN model in Fig. 10. Among the LSTM models, the 2-D stacked LSTM is superior, providing an accuracy of 89.18%, with an F1-score of 82.08%; such a performance is due to the consideration of both temporal and spectral features of the PLU's access pattern. Meanwhile, the 1-D LSTM has the least accuracy of 52.91% among all baseline approaches. On the other hand, TCN provides the best accuracy of 94.73% among the baseline ML-based models. Such accuracy is very close to our proposed ST-SNN model and is attributed to the fact that the TCN architecture incorporates CNN along with the temporal network as means to capture spectral and temporal vacancy. CNN considers each data frame as an image to find the correlation with the next frame; hence, it yields better accuracy. Moreover, TCN has a better F1 score of 92.37%, which is also the best among other baseline models; however, it is lower than our proposed ST-SNN model. In addition, it is computationally demanding and does not fit edge devices.

Moreover, we compare the implementation complexity of the ST-SNN and baseline models in Table X. In addition to having the least accuracy of 52.91%, the 1-D LSTM needs 215.1M trainable parameters and 302.53G FLOPS (floating point operation per seconds); such accuracy grows by 37% when using 2-D inputs for LSTM. Such a leap in accuracy comes at the price of 22% increase in the parameter size (from 215.1M to 273.8M) as well as 37.97% increase in complexity (from 307.53G FLOPS to 424.32G FLOPS). Meanwhile, TCN performs better than the 1-D and 2-D LSTM models. As shown in the table, TCN yields accuracy of 94.73% with 42.1 M parameters, which is almost 85% less than the 2-D LSTM. Moreover, TCN needs less number of FLOPS. However, as shown in the Table X, our proposed ST-SNN model significantly outperforms all these baseline approaches in terms of accuracy, size trainable parameters, and total

TABLE XI: FPGA Implementation of SNEC and CIC neurons using Vitis-HLS tools for target device VC709 board.

Resource utilization	IP Block		
	Initialization	SNEC neurons	CIC neurons
LUT	6913	39	38
LUTRAM	134	1	1
FF	4533	89	86
DSP	227	0	0
BRAM	22	0	0

FLOPS count. The major reduction in the number of trainable parameters as well as complexity achieved by the ST-SNN design is due to adopting a bio-inspired neural model with distinct temporal and spectral networks. In other words, ST-SNN does not need 12 sub-carriers per PRB, or to consider 14 symbols per sub-frame, as it is encoded with Poisson spikes. Overall, our ST-SNN design is not only effective but also lightweight and is well-suited for edge devices and other resource-constrained applications.

To further validate the robustness of our proposed ST-SNN model, we generated synthetic data of the same length (158K sub-frame) where the number of active PLUs in a frame is randomly picked. The objective of considering such data is to control the distribution of PLU access and study the effect on performance. We compare our proposed ST-SNN model with the TCN model, which performed best among all considered baseline approaches. The results show that despite the random variation in the PLU population, and consequently spectrum usage, our ST-SNN model continues to achieve high accuracy of around 95.26%, which is very close to the observed 96.6% figure when using live data. Our model also continues to outperform the TCN model, which yields 94% accuracy for the synthetic data.

Finally, to show that the bio-inspired ST-SNN model could work in real-time, we use Vitis-HLS, a high-level synthesis compiler, to turn our system C implementation of spiking SNEC neurons and CIC neurons into register transfer level (RTL). It is to be noted that we convert our Matlab implementation of SNEC and CIC neurons into system C. For FPGA implementation, our target device is the Xilinx Virtex-7 (VC709 evaluation board), and the resource allocation for implementing these neurons is listed in Table XI, which is less than 10% of the overall resources of the VC709 evaluation board. Realization of the entire prediction-based ST-SNN system requires implementing the other parts of the architecture as well as the pre-processing and detection sub-systems, and is part of our future work.

## VI. CONCLUSION

In this paper, we have studied the viability of predicting white spaces in the frequency spectrum for CRN. We have presented an ST-SNN model to capture the dynamics of PLU access and predict opportunities for SNU transmissions. The performance of our ST-SNN model is validated using live LTE data. The validation results have confirmed the effectiveness of the ST-SNN model as an alternative to state-of-art detection techniques by boosting both the spectral efficiency and the rate of successful data transfer. We have also compared the

complexity of our ST-SNN model with competing LSTM- and TCN-based approaches. The results have shown that ST-SNN requires limited computational and energy resources, and stands out as a lightweight solution that suits edge devices. In the future, we plan to incorporate ST-SNN with 5G networks, where the conventional detectors yield low accuracy.

## REFERENCES

- [1] A. Agarwal and R. Gangopadhyay, "Predictive spectrum occupancy probability-based spatio-temporal dynamic channel allocation map for future cognitive wireless networks," *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 8, p. e3442, 2018.
- [2] Y.-C. Liang, Y. Zeng, E. C. Peh, and A. T. Hoang, "Sensing-throughput tradeoff for cognitive radio networks," *IEEE transactions on Wireless Communications*, vol. 7, no. 4, pp. 1326–1337, 2008.
- [3] B. W. Stevens, J. Ray, E. Daugherty, J. S. Everett, Y. Yang, and M. F. Younis, "Cognitive resource analyzer for cellular network ecosystems," *IEEE Trans. on Cogn. Comm. and Net.*, vol. 8(2), pp. 733–749, 2022.
- [4] S. Wang, Z.-H. Zhou, M. Ge, and C. Wang, "Resource allocation for heterogeneous cognitive radio networks with imperfect spectrum sensing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 3, pp. 464–475, 2013.
- [5] S. Stotas and A. Nallanathan, "On the throughput and spectrum sensing enhancement of opportunistic spectrum access cognitive radio networks," *IEEE Trans. on Wireless Comm.*, vol. 11(1), pp. 97–107, 2011.
- [6] H. Chen, M. Zhou, L. Xie, K. Wang, and J. Li, "Joint spectrum sensing and resource allocation scheme in cognitive radio networks with spectrum sensing data falsification attack," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9181–9191, 2016.
- [7] M. S. Gupta and K. Kumar, "Progression on spectrum sensing for cognitive radio networks: A survey, classification, challenges and future research issues," *J. of Net. & Comp. App.*, vol. 143, pp. 47–76, 2019.
- [8] N. Chaudhary and R. Mahajan, "Identification of spectrum holes using energy detector based spectrum sensing," *International Journal of Information Technology*, vol. 13, no. 3, pp. 1243–1254, 2021.
- [9] C. Lee and W. Lee, "Exploiting spectrum usage patterns for efficient spectrum management in cognitive radio networks," in *2010 24th IEEE Int'l Conf. on Advanced Info. Net. & App.*, 2010, pp. 320–327.
- [10] M. Karimi, S. M. S. Sadough, and M. Torabi, "Improved joint spectrum sensing and power allocation for cognitive radio networks using probabilistic spectrum access," *IEEE Systems Journal*, vol. 13, no. 4, pp. 3716–3723, 2019.
- [11] J. Kim and J. P. Choi, "Sensing coverage-based cooperative spectrum detection in cognitive radio networks," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 5325–5332, 2019.
- [12] F. Obite, A. D. Usman, and E. Okafor, "An overview of deep reinforcement learning for spectrum sensing in cognitive radio networks," *Digital Signal Processing*, vol. 113, p. 103014, 2021.
- [13] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209–2221, 2013.
- [14] B. Sliwa, R. Falkenberg, and C. Wietfeld, "Towards cooperative data rate prediction for future mobile and vehicular 6g networks," in *2020 2nd 6G Wireless Summit (6G SUMMIT)*. IEEE, 2020, pp. 1–5.
- [15] F. A. Bhatti, M. J. Khan, A. Selim, and F. Paisana, "Shared spectrum monitoring using deep learning," *IEEE Trans. on Cogn. Comm. and Net.*, vol. 7(4), pp. 1171–1185, 2021.
- [16] M. A. Hossain, R. M. Noor, K.-L. A. Yau, S. R. Azzuhri, M. R. Z'aba, and I. Ahmedy, "Comprehensive survey of machine learning approaches in cognitive radio-based vehicular ad hoc networks," *IEEE Access*, vol. 8, pp. 78 054–78 108, 2020.
- [17] V. P. Rekkas, S. Sotirioudis, P. Sarigiannidis, S. Wan, G. K. Karagiannidis, and S. K. Goudos, "Machine learning in beyond 5g/6g networks—state-of-the-art and future trends," *Electronics*, vol. 10, no. 22, p. 2786, 2021.
- [18] X. Zhou, G. Y. Li, D. Li, D. Wang, and A. C. Soong, "Probabilistic resource allocation for opportunistic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 9, no. 9, pp. 2870–2879, 2010.
- [19] A. Sultana, X. Fernando, and L. Zhao, "An overview of medium access control strategies for opportunistic spectrum access in cognitive radio networks," *Peer-to-Peer Net. & App.*, vol. 10(5), pp. 1113–1141, 2017.



- [20] L. Lu, X. Zhou, U. Onunkwo, and G. Y. Li, "Ten years of research in spectrum sensing and sharing in cognitive radio," *EURASIP journal on wireless comm. and networking*, vol. 2012, no. 1, pp. 1–16, 2012.
- [21] A. Ghasemi and E. S. Sousa, "Collaborative spectrum sensing for opportunistic access in fading environments," in *First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005*. IEEE, 2005, pp. 131–136.
- [22] E. C. Peh, Y.-C. Liang, Y. L. Guan, and Y. Zeng, "Power control in cognitive radios under cooperative and non-cooperative spectrum sensing," *IEEE Transactions on wireless communications*, vol. 10, no. 12, pp. 4238–4248, 2011.
- [23] M. Ozger and O. B. Akan, "On the utilization of spectrum opportunity in cognitive radio networks," *IEEE Communications Letters*, vol. 20, no. 1, pp. 157–160, 2015.
- [24] Q. Wu, G. Ding, J. Wang, and Y.-D. Yao, "Spatial-temporal opportunity detection for spectrum-heterogeneous cognitive radio networks: Two-dimensional sensing," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 516–526, 2013.
- [25] S. Chowdhury, "Resource allocation in cognitive radio networks using stackelberg game: A survey," *Wireless Personal Communications*, vol. 122, no. 1, pp. 807–824, 2022.
- [26] A. T. Hoang, Y.-C. Liang, D. T. C. Wong, Y. Zeng, and R. Zhang, "Opportunistic spectrum access for energy-constrained cognitive radios," *IEEE Trans. on Wireless Comm.*, vol. 8, no. 3, pp. 1206–1211, 2009.
- [27] C. Wu, Y. Wang, and Z. Yin, "Energy-efficiency opportunistic spectrum allocation in cognitive wireless sensor network," *EURASIP Journal on Wireless Comm. and Networking*, vol. 2018, no. 1, pp. 1–14, 2018.
- [28] K. Yang, Z. Huang, X. Wang, and X. Li, "A blind spectrum sensing method based on deep learning," *Sensors*, vol. 19, no. 10, p. 2270, 2019.
- [29] R. Amr, N. A. Zaher, S. M. Gasser, and S. K. Eldiasty, "A new deep spiking architecture for reconstruction of compressed data in cognitive radio networks," *IEEE Access*, 2022.
- [30] G. Eappen, T. Shankar, and R. Nilavalan, "Cooperative relay spectrum sensing for cognitive radio network: Mutated mwaa-snn approach," *Applied Soft Computing*, vol. 114, p. 108072, 2022.
- [31] X. Chen, H. Zhang, A. B. MacKenzie, and M. Matinmikko, "Predicting spectrum occupancies using a non-stationary hidden markov model," *IEEE wireless communications letters*, vol. 3, no. 4, pp. 333–336, 2014.
- [32] V. K. Tumuluru, P. Wang, and D. Niyato, "A neural network based spectrum prediction scheme for cognitive radio," in *2010 IEEE international conference on communications*. IEEE, 2010, pp. 1–5.
- [33] K. Hamedani, L. Liu, S. Liu, H. He, and Y. Yi, "Deep spiking delayed feedback reservoirs and its application in spectrum sensing of mimo-ofdm dynamic spectrum sharing," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 34, no. 02, 2020, pp. 1292–1299.
- [34] P. Chauhan, S. K. Deka, B. C. Chatterjee, and N. Sarma, "Cooperative spectrum prediction-driven sensing for energy constrained cognitive radio networks," *IEEE Access*, vol. 9, pp. 26 107–26 118, 2021.
- [35] M. Ozturk, M. Akram, S. Hussain, and M. A. Imran, "Novel qos-aware proactive spectrum access techniques for cognitive radio using machine learning," *IEEE Access*, vol. 7, pp. 70 811–70 827, 2019.
- [36] L. Yu, Y. Guo, Q. Wang, C. Luo, M. Li, W. Liao, and P. Li, "Spectrum availability prediction for cognitive radio communications: A dcg approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 476–485, 2020.
- [37] M. A. Aygöl, M. Nazzal, A. R. Ekti, A. Görçin, D. B. da Costa, H. F. Ateş, and H. Arslan, "Spectrum occupancy prediction exploiting time and frequency correlations through 2d-lstm," in *2020 IEEE 91st Vehicular Technology Conf. (VTC2020-Spring)*. IEEE, 2020, pp. 1–5.
- [38] A. Ghasemi and J. Parekh, "Deepair: Enabling data-driven dynamic spectrum sharing via scalable forecasting," *IEEE Transactions on Cognitive Communications and Networking*, vol. 9, no. 1, pp. 16–27, 2022.
- [39] K. Xie, Z. Zhang, B. Li, J. Kang, D. Niyato, S. Xie, and Y. Wu, "Efficient federated learning with spike neural networks for traffic sign recognition," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9980–9992, 2022.
- [40] J. D. Nunes, M. Carvalho, D. Carneiro, and J. S. Cardoso, "Spiking neural networks: A survey," *IEEE Access*, vol. 10, pp. 60 738–60 764, 2022.
- [41] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. Le Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, S. Menzel *et al.*, "2022 roadmap on neuromorphic computing and engineering," *Neuromorphic Computing and Engineering*, vol. 2, no. 2, p. 022501, 2022.
- [42] C. B. Calderon, T. Verguts, and M. J. Frank, "Thunderstruck: The acdc model of flexible sequences and rhythms in recurrent neural circuits," *PLoS computational biology*, vol. 18, no. 2, p. e1009854, 2022.
- [43] A. Maes, M. Barahona, and C. Clopath, "Learning spatiotemporal signals using a recurrent spiking network that discretizes time," *PLoS computational biology*, vol. 16, no. 1, p. e1007606, 2020.
- [44] A. Destexhe, M. Rudolph, and D. Paré, "The high-conductance state of neocortical neurons in vivo," *Nature reviews neuroscience*, vol. 4, no. 9, pp. 739–751, 2003.
- [45] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on neural networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
- [46] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *Journal of neurophysiology*, vol. 94, no. 5, pp. 3637–3642, 2005.
- [47] X. Zhou, M. Sun, G. Y. Li, and B.-H. F. Juang, "Intelligent wireless communications enabled by cognitive radio and machine learning," *China Communications*, vol. 15, no. 12, pp. 16–48, 2018.
- [48] M. G. Khoshkholgh, K. Navaie, and H. Yanikomeroglu, "Access strategies for spectrum sharing in fading environment: Overlay, underlay, and mixed," *IEEE Trans. on Mobile Comp.*, vol. 9(12), pp. 1780–1793, 2010.
- [49] A. Maes, M. Barahona, and C. Clopath, "Learning compositional sequences with multiple time scales through a hierarchical network of spiking neurons," *PLoS computational biology*, vol. 17, no. 3, p. e1008866, 2021.
- [50] *Long Term Evolution (LTE) and LTE-Advanced Pro.* John Wiley Sons, Ltd, 2017, ch. 4, pp. 211–333. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119346913.ch4>
- [51] A. Litwin-Kumar and B. Doiron, "Formation and maintenance of neuronal assemblies through synaptic plasticity," *Nature communications*, vol. 5, no. 1, p. 5319, 2014.
- [52] —, "Slow dynamics and high variability in balanced cortical networks with clustered connections," *Nature neuroscience*, vol. 15, no. 11, pp. 1498–1505, 2012.
- [53] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.



applications of spike neural networks in wireless communication.



He has published over 350 technical papers in refereed conferences and journals. Dr. Younis has five granted and two pending patents. In addition, he serves/served on the editorial board of multiple journals and the organizing and technical program committees of numerous conferences. Dr. Younis is a fellow of the IEEE and the IEEE communications society.

**Md Mehedi Hassan Galib** received the B.Sc. degree in electrical and electronic engineering from Islamic University of Technology, Bangladesh, in 2008, the M.Sc. in electronics & radio engineering from Kyung Hee University, South Korea, in 2015, and the M.Sc. in computer engineering from the University of Maryland Baltimore County (UMBC), MD, USA, in 2023. Currently, he is pursuing a Ph.D. in computer engineering at UMBC, MD, USA. He is a member of the Embedded Systems and Networks Laboratory (ESNet), where his research focuses on

**Prof. Mohamed Younis** is currently a professor in the CSEE department of UMBC. Before joining UMBC, he was with the Aerospace Electronic Systems R&D division of Honeywell International Inc. He also participated in the development of the Redundancy Management System, which is a key component of the Vehicle and Mission Computer for NASA's X-33 space launch vehicle. Dr. Younis' technical interest includes network architectures and protocols, Internet of Things, fault tolerant computing, secure communication and cyber-physical



**Brian W. Stevens** (Member, IEEE) received his Ph.D. from the University of Maryland at Baltimore County where he studied cognitive radio research. He is a Senior Computer Engineer at Geon Technologies, LLC. His interests include wireless networks, embedded programming, and digital signal processing.