Electronic Communications of the EASST
Volume 080 (2021)

Conference on Networked Systems 2021 (NetSys 2021)

Time- and Frequency-Domain Dynamic Spectrum Access: Learning Cyclic Medium Access Patterns in Partially Observable Environments

Sebastian Lindner, Daniel Stolpmann, Andreas Timm-Giel

14 pages

Guest Editors: Andreas Blenk, Mathias Fischer, Stefan Fischer, Horst Hellbrueck, Oliver Hohlfeld, Andreas Kassler, Koojana Kuladinithi, Winfried Lamersdorf, Olaf Landsiedel, Andreas Timm-Giel, Alexey Vinel
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Time- and Frequency-Domain Dynamic Spectrum Access: Learning Cyclic Medium Access Patterns in Partially Observable Environments

Sebastian Lindner[1], **Daniel Stolpmann**[2], **Andreas Timm-Giel**[3]

[1]sebastian.lindner@tuhh.de, [2]daniel.stolpmann@tuhh.de, [3]timm-giel@tuhh.de
Hamburg University of Technology, Institute of Communication Networks, Germany

Abstract: Upcoming communication systems increasingly often tackle the spectrum scarcity problem through the coexistence with legacy systems in the same frequency band. Cognitive Radio presents popular methods for Dynamic Spectrum Access (DSA) that enable coexistence. Historically, DSA meant a separation solely in the frequency domain, while in recent years it has been extended through the dimension of time, by employing Machine Learning to learn semi-deterministic and cyclic medium access patterns of the legacy system that are observed through channel sensing. When this pattern is learnable, then a new system can utilize a neural network and predict future medium accesses, thus steering its own medium access. We investigate this novel and more fine-grained version of DSA, propose a predictor and show its capability of reliably predicting future medium accesses of a legacy system in an aeronautical coexistence scenario. We extend the predictor to the case of partial observability, where only a narrowband receiver is available, s.t. observations are limited to a single sensed channel per time slot. In particular, we propose a custom loss function that is tailored to partially observable environments. In the spirit of Open Science, all implementation files are released under an open license.

Keywords: Cognitive Radio, Communication System Coexistence, Dynamic Spectrum Access, Artificial Neural Networks, Machine Learning

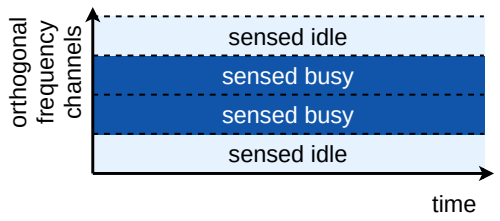
1 Introduction

The allocation of frequency spectrum to communication systems is decided by the local political body, and at present this finite resource has become very sparse, making it difficult to obtain the license to utilize a frequency band suitable for communication when designing a new system. Two options exist: 1) move to unutilized high-frequency spectrum and work around problematic properties, or 2) share already utilized spectrum with legacy systems.

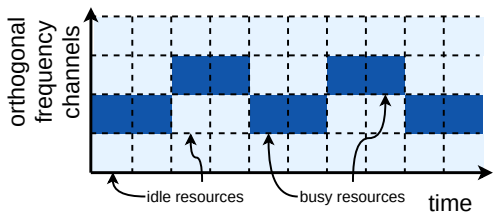
Research on the first approach is manifold and focuses mainly on Physical (PHY) layer design. Research on the second approach is becoming increasingly popular. Here the field of Cognitive Radio (CR) research must be named, where in its terms, a legacy user that is licensed to use a particular frequency band is called the Primary User (PU), while a novel Secondary User (SU) is made "cognitive" in such a way that it operates on the same frequency band – it *coexists* with the PU – without the two systems causing interference upon each other. To achieve this, in particular, data link layer methods are required, so that coexistence is achieved through channel

access separation in time, frequency or both.

In this paper, a dynamic Mobile Ad-hoc Network (MANET) scenario is considered. In our earlier work in [LFT20] we have shown that Artificial Neural Networks (ANNs) can be capable of learning a PU's medium access pattern over time and frequency if certain properties hold. In particular, the PU's pattern must be cyclic and semi-deterministic. While these properties do not apply to all communication systems, they do apply to some. In particular, the authors are involved in a German research project on a novel, aeronautical communication system called *L*-band Digital Aeronautical Communications System (LDACS), which shall be a key component within the ICAO Global Air Navigation Plan's Future Communications Infrastructure. It may provide both communications and navigation to LDACS-equipped aircraft through air-to-ground and air-to-air links. LDACS will have to coexist with possibly several legacy systems that operate on the lower *L*-band. Most prominently, the Distance Measuring Equipment (DME) system, which provides navigation to aircraft and whose medium access pattern *is* cyclic and deterministic, has been determined in [EHS12]. For LDACS to be deployable, it must be ensured that no interference is caused on DME operation by new LDACS users, which may be achieved through a channel block list at the LDACS user, so that it refrains from channels utilized by local DME operation as proposed in [BS21]. While safe for standardization, this approach is not beneficial to spectral efficiency. We propose that a SU should *learn* the PU's medium access strategy, and then use Dynamic Spectrum Access (DSA) so that simultaneous access on the medium is avoided. Recently DSA has been proposed to be the dynamic selection of frequency *and* time resources, as opposed to just frequency resources – see Fig. 1. Our contributions are as follows. In



(a) Historically DSA has been used to dynamically select frequency channels s.t. interference on the PU is avoided.



(b) When possible, DSA should dynamically select (time, frequency)-resources s.t. interference on the PU is avoided.

Figure 1: The added dimension of time to DSA.

the related work we motivate the extended DSA through further possible application domains. A brief overview on research in this direction is given, where recently numerous publications have been made that utilize usually Machine Learning (ML) or Game Theory to achieve coexistence. We extend our earlier work by lifting an unrealistic assumption that had been made: instead of assuming a fully observable system state, we show how the method can be applied to a partially observable one, and how this affects performance. We demonstrate the feasibility of the learning task on the example of coexistence with DME using a simple Recurrent Neural Network (RNN) and provide first steps towards the reduction of learning time.

2 Related Work

In recent years, the topic of DSA has gained popularity. In [LCBM12], it is considered how a group of Wi-Fi Access Points (APs) could self-configure their channel choice distributively and without communication. It is emphasized that coordination through communication usually suffers from the interference range being substantially larger than the communication range, and so a method that relies solely on *channel sensing* would be considerably more general than one that requires dedicated control data exchange. Therefore, a distributed graph coloring solution algorithm based on channel sensing is proposed. However, it is also acknowledged that due to channel characteristics depending on the particular frequency, interference regions will be channel-dependent. And so the channel allocation problem is actually equivalent to "a more general multi-graph coloring task". Their algorithm initially selects a channel uniformly random, and selection probabilities are adapted based on the observed performance: if communication does not succeed, the selection probability is multiplicatively decreased; other channels' probabilities are increased accordingly so that a selection probability distribution is maintained. It corresponds to the historic coining of DSA as the dynamic selection of frequency channels.

The authors of [MMRC20] propose DSA to meet "the growing demands of forthcoming and deployed wireless networks", and in particular they focus on the coexistence of Long Term Evolution (LTE)-License-Assisted Access (LAA) with IEEE 802.11 Wi-Fi. For this, they formulate a variant of Q-Learning, which both 802.11 APs and LTE evolved NodeBs (eNBs) use for subchannel selection. These base stations each serve several users, and the subchannel selection is fully distributed: no central controller or information exchange takes place between the base stations. As for the formulation of the Reinforcement Learning (RL)-based Q-agent, which each base station embodies, the authors base the agent's reward on its achieved throughput, which should be maximized. Concretely, each channel may be in one of four states: idle, successful transmission, collision, contention. The actions are then the selection of channels. The approach could be summarized as a ML-based distributed graph coloring solution where learning agents are selfish and aim to maximize their own throughput.

The paper in [AMT10] is considerably closer to our assumption and ponders the question of SUs that may sense a single channel per time slot: a partially observable environment. However, the authors assume these channels as i.i.d., which we do not. Their approach aims to find each channel's mean availability, which is used to rate them in such a way that least-utilized channels are selected for SU transmissions most. Most interestingly, they prove that the *regret*, which is the subtraction of the numbers of successful transmissions in the optimal case and in the learned case, shows logarithmic growth over time. However, they find that all users rank channels mostly in the same way, and to prevent the collisions that would follow from all users selecting the same channels, these selections are randomized.

The extension of DSA by the dimension of time has, to our knowledge, first been proposed in 2017 by the authors of [WLGK17], [WLGK18]. The paper [WLGK18] proposes a Deep Q-Network for developing a channel selection policy when channels are correlated and modeled as Markov chains. The Q-agent is positively rewarded if it selects an idle channel, which is consequently sensed and if found idle, used to transmit a packet. It could be described as a predictive Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) protocol. Afterwards, the concept has received considerably more attention, and has been directly extended by [TR20]. There

the neural network architecture is extended by a Long Short Term Memory (LSTM) layer and Double Deep-Q learning. It is conceptually similar to our proposal, but differs in some fundamental assumptions. First, their CSMA/CA variant neglects the receiving node, which needs to know which channel the transmitter node's Q-agent selects next in order to be able to receive a packet. This is why we focus instead on establishing the foundation of making reliable predictions first, which can *then* be forwarded to any Medium Access Control (MAC) protocol that utilizes them. Second, they select *one* channel per time slot to try and transmit, where we provide a prediction on *all* channels' availabilities. This coincides with the first point, s.t. our method can be viewed as a general data link layer function that should be connected to a fitting MAC protocol.

To close in on our application domain, when fine-grained resource selection of both frequency *and* time are the goal, then the PU's medium access over time must follow some pattern for it to be learned and accounted for. A random access-based CSMA/CA scheme intuitively presents no such pattern, but may still do through application layer behavior. The resource scheduler in LTE networks is up to the operator, and it can be expected that it will occupy its frequency bands fully. In both cases, all PU-used bands would be learned as *mostly busy*, and then the here-proposed method, too, degrades to channel selection. We are more interested in showing the effectiveness in coexistence scenarios where a more fine-grained resource allocation *can* succeed, and so we will present the aeronautical coexistence scenario for our evaluation in Sec. 3, and here briefly present another one that we deem applicable from the field of wireless sensor networks. One publication out of many is [PAG+13], which motivates a *traffic aware scheduling algorithm* for data-centric scheduling that is multi-channel, time-synchronized and duty-cycled, and which has been integrated into the IEEE 802.15.4e Time Slotted Channel Hopping (TSCH) protocol, standardized in [IEE]. Putting it briefly, communicating nodes agree on a particular resource in the (time, frequency)-grid, and then repeatedly follow the same *hopping sequence* of finite length, and so attempt to avoid interference with other networks. This leads to long-lasting links between sensor nodes and a data aggregator, that are completely cyclic. So while TSCH proposes a pseudo-random-channel-hopping-based interference mitigation, which evenly utilizes all available frequency channels, we propose a learning-based interference mitigation that selectively utilizes frequency channels and time slots within *if* they are predicted as idle; and such an IEEE 802.15.4 system has already been shown in [WLGK18] to be a good candidate for such learning-based coexistence.

3 System Model

We consider a SU communication system, whose communication resources are derived from both Time Division Multiplexing (TDM) and Frequency Division Multiplexing (FDM). Time is organized into discrete time slots of identical duration, and the frequency band is split into $\mathscr{C} = \{1, \ldots, c\}$ orthogonal narrowband channels with identical bandwidths. The users of the SU network attempt to opportunistically access the medium. A PU communication system is present, which operates on the same frequency band. The number of PUs is unknown, but the superposition of their medium accesses on the particular frequency channels can be observed through channel sensing. Definitions of time slots and frequency channels between PUs and SUs

are not required to be identical. Each channel therefore has its own availability statistic, and these are local to each SU due to different geographic positions and the hidden node problem. Depending on the PU's channel access strategy, there may be a correlation between the channels w.r.t. their availability. In fact, a finite, repeating (cyclic) and (semi)-deterministic PU channel hopping sequence is assumed as in [LFT20], where the next frequency channel may be drawn from this finite sequence with probability $1-\varepsilon$, and with ε no hop is made. At $\varepsilon = 0.5$ the PU behavior is most unpredictable, while at $\varepsilon \in \{0,1\}$ it is fully deterministic. Here a fully deterministic PU strategy with $\varepsilon = 0$ is assumed. It is each SU's goal to reliably learn the availability statistics of the channels s.t. predictions about future utilization can be made. These predictions can then be used to steer the local medium access – the SU network needs to organize among its users, but this is outside the scope of this paper. Instead, the focus is on making such predictions. We assume a single hardware receiver to be available for the learning task at each SU.

3.1 Environment model

We consider two different types of receivers that are available to the SU, where the first is a specialized version that allows full observation of the medium. It is capable of sensing over a broad frequency range, and then differentiating between received radio signals on the narrowband frequency channels, making out independent radio signals on each channel. The complexity of such a receiver depends on the width of the frequency band. In the example of LDACS, it is likely to operate on 500 kHz wide channels from the range 960–1164 MHz, as shown in [EHS12]. This would require the specialized receiver to detect signals in 408 narrowband channels on a 204 MHz wide band. Optimistically speaking, a complex and very expensive receiver is needed. Other communication systems with fewer channels and a smaller band *may* be realizable with such a receiver. The second version is a simple narrowband receiver, which can be tuned to a single narrowband frequency channel and detect radio signals within, but provides no information about signals on the other channels.

3.1.1 Fully observable environment model

In a fully observable scenario, we assume a specialized receiver to be available at the SU, which is capable of sensing all channels at each time slot, and can determine the availability of each channel. At time t, a sensing result would be X_t as in Eq. 1.

$$X_t = \{x_{t,1}, \ldots, x_{t,c}\} \text{ where } \forall i = 1, \ldots, c : x_{t,i} = \begin{cases} 1 & \text{if channel } i \text{ was found idle} \\ -1 & \text{if channel } i \text{ was found busy} \end{cases} \tag{1}$$

3.1.2 Partially observable environment model

In a partially observable scenario, only a narrowband receiver is available at the SU, which can be tuned to a single frequency channel during a time slot to obtain a measurement, but cannot give information about the other channels. Therefore, at time t, a sensing result $\tilde{X}_{t,j}$ would yield

an observation for a selected channel j, and give no information about any other channel as in Eq. 2.

$$\tilde{X}_{t,j} = \{\tilde{x}_{t,1}, \ldots, \tilde{x}_{t,c}\} \text{ where } \forall i = 1, \ldots, c : \tilde{x}_{t,i} = \begin{cases} 1 & \text{if channel } i = j \text{ was found idle} \\ 0 & \text{if } i \neq j \\ -1 & \text{if channel } i = j \text{ was found busy} \end{cases} \tag{2}$$

3.1.3 Primary user channel access behavior model

The PU accesses the medium according to its own strategy. It is initially unknown to us, and it is our goal to *learn* it from sensing samples X_t or $\tilde{X}_{t,j}$ over time. We evaluate the learning approach on the DME system. It is used for navigation purposes, where ground stations are configured to transmit at a fixed center frequency on a 1 MHz-wide channel from the range 960 – 1215 MHz, according to [ESCG11] and [EHS12]. An interrogating aircraft sends a request pulse pair, to which the ground station replies at a frequency offset of ±63 MHz and a fixed delay of 12 – 36 µs that depends on the mode.

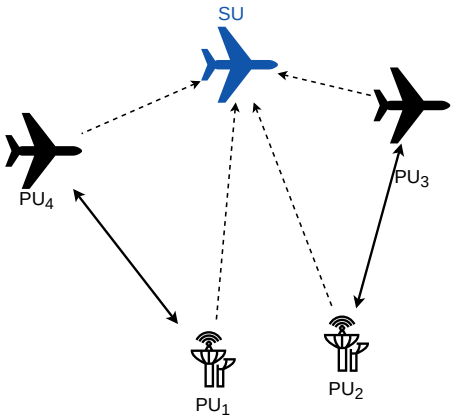
We abstract this operation, and consider a channel at time slot t as utilized by DME if an interrogation or response signal was present in its duration. There is a correlation between the interrogation and response frequency channels, where a response signal on the response channel follows an interrogation signal on the interrogation channel. An LDACS-equipped aircraft may be in the presence of several DME ground stations, which each serves a number of users. For example, consider Fig. 2a, where one LDACS user may receive interrogation signals of two DME users, and response signals of two DME ground stations.

With this, as the SU observes the medium over time, it may uncover a resource utilization as in Figs. 2b, 2c. Note that propagation delays in aeronautical communication are non-negligible: according to [LE12], DME provides service up to a transmission range of 740 km, which imposes a propagation delay of almost 2.5 ms, and the interference range is typically much larger than the transmission range. The relative positions between SU and PUs are therefore of utmost importance, and mobility changes the spectral view at the SU. In the example in Figs. 2b, 2c, time is discretized into slots such that interrogation and response signals are separated into two consecutive slots. Time could be discretized differently and the number of PUs could change, but the critical aspect is that DME medium access is *cyclic*, showing a pattern over time which changes only as users move.

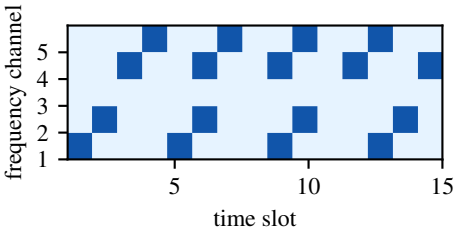
3.2 Prediction model

3.2.1 Channel access pattern learning

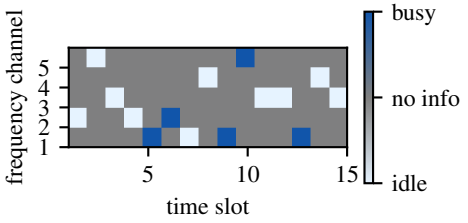
As full or partial observations are made, these are used to train a predictor. Earlier work in [LFT20] and [FLT20] have shown that a LSTM RNN works very well for this task. In a nutshell, this type of neural network is capable of learning correlations over time, which fits well with the prediction of the time series that PU medium access can be viewed as. The earlier papers present a detailed discussion on the learning characteristics of such RNNs, such as learning time, the



(a) Example case where two DME ground stations serve one user each. All radio signals arrive at the SU.



(b) Fully observable sampling at each time slot.



(c) Random partial sampling at each time slot.

Figure 2: Exemplary view of sampled PU resource utilization over time at the SU.

values predictions converge towards, and consider continuous learning where mobility changes the learning target over time for deterministic and for Markovian PU channel access patterns.

Training data is obtained over time during SU operation, where at time t signal reception is used to construct the current observation X_t as in Eq. 1. This observation is used as the target (or label) for the previous observation X_{t-1}, training the predictor to output X_t given X_{t-1}. The input into the learning process therefore lags behind the sensing process by one time slot, as a target is required before training can commence.

As for the RNN architecture, our focus is on feasibility and not on the optimization of the architecture. Thus, a minimalistic LSTM RNN as depicted in Fig. 3 is used, where the observation is put into an LSTM layer of 128 neurons, which is fully connected to a c-neuron output layer. This means a prediction on each of the c channels' availability at $t+1$ is obtained at time t. The hyperbolic tangent activation function ensures LSTM layer outputs to be from $[-1,\ 1]$.

Custom loss function In partially observable environments, as Eq. 2 states, only one element in $\tilde{X}_{t+1,j}$ will be non-zero (contain information); *which* element this is depends on the selection of the sensed channel j. This selection is uniformly random over all channels, and so each channel has a probability of $\frac{1}{c}$ to be selected. Fix one channel j. Over time, channel j is sampled equally often as all other channels. However, the vast majority $\frac{c-1}{c}$ of all observations will be of *other*

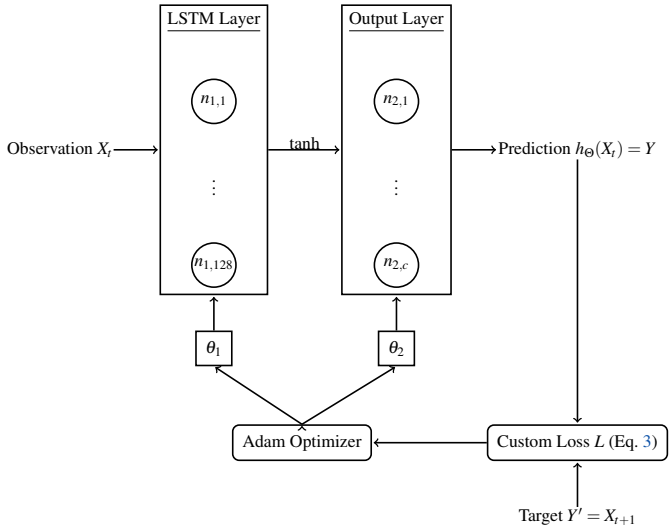


Figure 3: Our LSTM RNN architecture.

channels, where $\tilde{X}_{t+1,k\neq j}$ will contain an entry of zero at channel j. The go-to loss function for such applications is the Mean Squared Error (MSE): $\text{MSE} = \frac{1}{c}\sum_{i=1}^{c}(Y_i - Y_i')^2$. With this, every time channel j *is* selected, the new observation on this channel will push the prediction towards 1 (idle) or -1 (busy), however most of the time the prediction on channel j will be pushed towards 0 (no info). Predictions on this channel will tend towards 0 as $t \to \infty$ or $c \to \infty$.

Therefore, a custom loss function is proposed, which takes into account the special circumstance of having partial information. The neural network's optimizer takes the loss value and applies gradient descent, i.e. it computes the gradient of the loss function and updates the neural weights in the opposite direction, thus reducing the loss. This leads to the just-described behavior of adjusting weights to predict a zero when the respective channel was not sensed, since this constitutes the target output that is provided to the optimizer. To circumvent this issue, the MSE is modified and instead the custom loss function L as given in Eq. 3 is used.

$$L = \max_i \left[\left(Y_i' \cdot (Y_i - Y_i') \right)^2 \right] \tag{3}$$

The target Y' must be zero in all positions but the selected channel, and so the only non-zero loss value is where Y' is non-zero. Consequently, the optimizer is given no incentive to change neural weights to reflect the zero elements in the target, but *only* attempts to output the target element that is non-zero. The function L practically boils down to the two cases of the non-zero label element $y' \in \{-1, 1\}$, so

$$L = \begin{cases} f_1 = (y-1)^2, & \text{if } y' = 1 \\ f_2 = (y+1)^2, & \text{if } y' = -1 \end{cases}$$

with $y \in \mathbb{R}$. These functions f_1 and f_2 are continuously differentiable. The custom loss function is only used for partially observable environments, while the MSE is used for fully observ-

able ones, since the maximum in Eq. 3 increases the learning time when all target values can be optimized jointly.

4 Results

All simulation results are obtained using Python and TensorFlow v2.5.0. Simulations are repeated 50 times and results split into batch means of 5 runs each, s.t. 95 % confidence intervals can be computed. The source code to replicate the results is provided under an open license in [Lin21].

4.1 Learning time

The learning time in partially observable environments is expected to increase in contrast to fully observable ones. To explain, consider the difficulty of reliably predicting the next time slot with only partial observations. Only $\frac{1}{c}$ of the information is gained per time slot, or in other words, the state space increases from $\mathscr{S} = \{-1, 1\}$ to $\mathscr{S}_0 = \{-1, 0, 1\}$ with the introduction of the "no information" state 0, and $|\mathscr{S}_0|^c = 3^c > |\mathscr{S}|^c = 2^c$ when all channels are considered. To evaluate this hypothesis, the coexistence scenario depicted in Fig. 2 is taken where one SU shall learn the superposition of channel accesses by two DME ground stations and two DME interrogators on $c = 5$ frequency channels. Channels 1 and 2 are utilized by one DME operation, channel 3 is always idle and channels 4, 5 are used by the other DME operation. Fig. 4 shows the neural network's learning process over time, where the loss is the MSE or Eq. 3 resp., and the accuracy is the binary accuracy of rounding prediction values to $r(y) = -1$ if $y < 0$ else 1, comparing it to the integer target, and averaging over all channels.

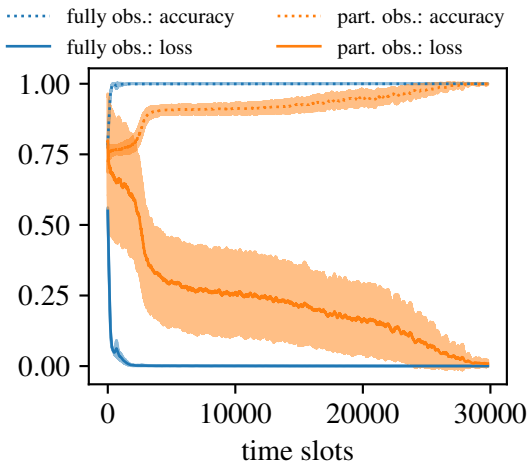


Figure 4: LSTM loss and accuracy over time for fully and partially observable models for the DME coexistence scenario visualized in Fig. 2.

The hypothesis is clearly supported as the fully observable case achieves a loss of zero at about $\frac{1}{20}$ of the training samples compared to the partially observable case. For partial observability, a random initialization of neural weights as well as randomly sensing the next channel lead to relatively large confidence intervals before convergence.

To investigate different difficulties in the learning targets, Fig. 5 depicts the time until convergence for the fully and partially observable system models over an increasing number of frequency channels. New DME ground stations and users are placed as channels 5 and 7 are added, so that the channel access pattern grows and the learning target becomes more difficult. Convergence is defined as that state of the model where the next consecutive 1000 slots are cor-

rectly predicted. Both environment models show an exponential growth in learning time as the number of frequency channels increases, while for partially observable scenarios this learning time is close to one order of magnitude larger and grows with c.

4.2 Learning time reduction

Fig. 5 shows that the learning time may become the limiting factor. This process cannot be sped up through more computing power, as it is the sparse learning samples that are limiting. Real-world channel measurements yield these observations, and these must have taken place before the learning model can be trained. In an exemplary LDACS network, when five channels are used by DME, the learning time translates to about 6 min under an assumed time slot duration of 12 ms. This may be acceptable when the pattern stays relatively constant as the user traverses the interference range of more than 1000 km, but for fast-changing environments, the PU's medium access pattern might have already changed by the time a now-obsolete pattern has been learned. Therefore, the reduction of learning time is crucial. A first step in this direction is the aggregation of multiple samples into an observation matrix M of l rows, where each row holds one observation vector, so $M = [X_{t-l+1}, X_{t-l+2}, \ldots, , X_t]'$. When a new observation X_t is made, the oldest observation is popped from M, and the new observation appended. With this, a sequence that shows the correlations over time is presented to the predictor. Note that this may put the necessity of an LSTM layer into question. The LSTM's memory cells enable the neural network to establish correlations over arbitrarily long periods of time. With this, the proposed method becomes general, as when it is applied on other (non-DME) coexistence scenarios, the cycle length of the PU's medium access pattern may not be known a-priori. An LSTM-provided memory can establish patterns over small input matrices that do not encode the full pattern, while a feed-forward neural network cannot learn the pattern if the input matrix is too small. Therefore, we argue *for* the LSTM layer, but emphasize that removing it for known PU pattern lengths can reduce the computational overhead.

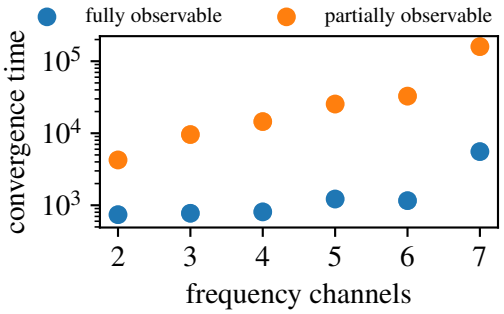
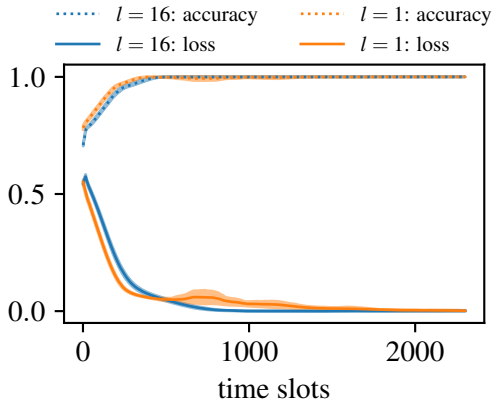


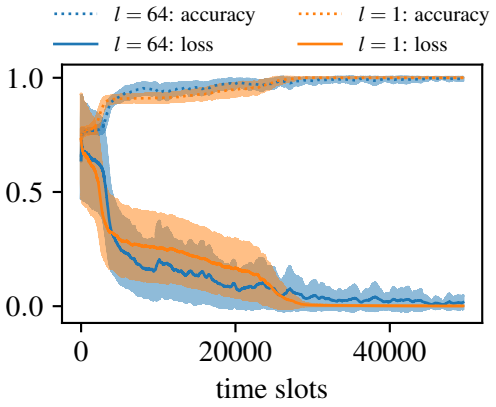
Figure 5: LSTM convergence time in slots over the number of channels, where with more channels also more DME users are added, and so the pattern that should be learned becomes more complicated.

Fig. 6a shows that the learning time in the fully observable case can be significantly reduced as loss and accuracy converge sooner when $l = 16$ samples are aggregated. The same behavior had been discovered in our previous work in [LFT20], where it was concluded that when a training sample contains the pattern that should be learned, convergence is reached much faster than if correlations *between* samples must be established through the LSTM memory cells. In Fig. 6c the substantial decrease in the training time required until convergence is shown, which follows a bathtub curve in the fully observable case with a minimum at $l = 16$ for this particular pattern. Finding the optimal l is then application-specific.

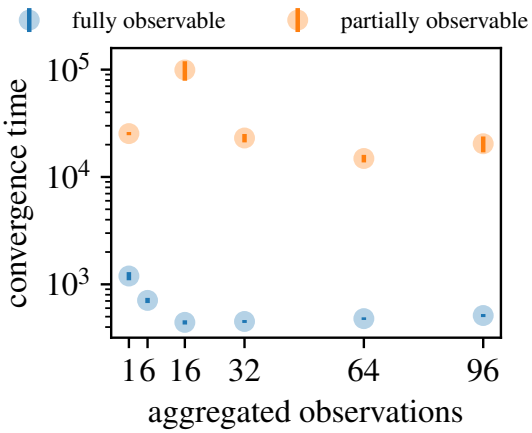
For partial observability, Fig. 6c shows relatively chaotic behavior with a minimum at $l = 64$. The time until convergence shows relatively large variance, and 6 out of 50 runs with $l = 16$ did not converge at all within 250000 slots. An evaluation of the training process in Fig. 6b likewise shows a large variance. This may be explained by the "usefulness" of an input matrix M, which is largely determined by the random choice of the channel sampling: it may contain useful information about the pattern to learn, or it may not. This information content varies between inputs, and utilizing matrices of little information is not beneficial for training. Thus, the strong variance in learning and convergence times, and the much later convergence in Fig. 6b appear. To reduce the learning time in partially observable environments, another method that aims at finding highly informative training inputs may be required.



(a) Learning behavior in fully observable case.



(b) Learning behavior in partially observable case.



(c) Convergence times for both cases.

Figure 6: A comparison of convergence times in Fig. 6c for the aggregation of different numbers l of observations into an input matrix M. The respective best values of $l = 16$ and $l = 64$ were selected for fully and partially observable scenarios respectively, and the LSTM learning behaviors are shown in Figs. 6a and 6b.

5 Discussion

It has been shown that if the coexisting PU system exhibits a cyclic and semi-deterministic medium access pattern, then through the utilization of a simple LSTM RNN, this pattern can be

learned as the medium is observed, and so DSA can be realized both in frequency *and* in time. An aeronautical scenario has been investigated, but it must be emphasized that the method is general and can be applied to the coexistence with any PU system if the required characteristic of a semi-deterministic cyclic medium access pattern holds, and related work has already demonstrated the applicability to other systems.

The predictor has been adapted to partially observable environments, where a single narrow-band receiver can obtain observations of a single frequency channel in each time slot. In order to achieve this, a custom loss function has been proposed, which is tailored to the specifics of partial information. Through this loss function, the RNN is prevented from adjusting its neural weights for unknown channel states.

As expected, the learning time increases drastically when the amount of information per time slot is reduced due to partial observability. The complexity of the PU's channel access pattern is the driving factor here. Since observations are obtained over time, it has become clear that the efficient utilization of available observation samples is crucial for effective deployment of this method. A first step into this direction has been the aggregation of several observation vectors into an observation matrix. This could significantly reduce the learning time in both fully and partially observable scenarios. However, for the partially observable case, a large variance in the learning behavior could be observed, and apparently the input matrix size must be chosen with care. A more stable method for learning time reduction should be found.

Future work should encompass this further reduction of the learning time as well as the utilization of predictions through a MAC protocol. In particular, our next focus lies in Reinforcement Learning, where a learning agent could be tasked with the selection of the next frequency channel that should be sensed, as related work has already shown this to be feasible. This is in contrast to the current method of a uniformly random selection of the next channel to sense. If channel availability statistics are i.i.d., then the information gained from sensing any channel is the same as sensing any other channel; sensing channels at uniformly-random is an optimal strategy in such a case. However, we consider cyclic channel access patterns with correlations in time and frequency: if a DME request channel is now busy, the corresponding DME response channel *will* be busy soon. Observing the correlation between these two channels gains much information. Another channel may be unutilized by local DME operation: initially sampling it will reveal it always idle – this is new information – but after having learned it is idle, sensing it again yields no new information; periodically checking that it *remains* idle is however still required. This may realize self-supervised channel sampling, as the agent learns to maximize the information gain per sampling round.

Another approach could be transfer learning, where a prediction model trained on one task could use its knowledge to apply it to a changed pattern when user mobility is considered. Our previous work [LFT20] already suggests that the proposed RNN architecture can adapt to changed learning targets e.g. through mobility, but the subject has not been investigated thoroughly so far. Similarly, knowledge transfer and student-teacher models could improve learning times for users that have just entered a geographic area as they are taught by existing users that have already trained their models.

Acknowledgements: This work was funded by the Hamburg University of Technology as part of its I^3 project *Machine Learning in Aeronautical Communications*.

Bibliography

[AMT10] A. Anandkumar, N. Michael, A. Tang. Opportunistic spectrum access with multiple users: Learning under competition. In *INFOCOM*. Pp. 1–9. IEEE, 2010.

[BS21] M. A. Bellido-Manganell, M. Schnell. Feasibility of the Frequency Planning for LDACS Air-to-Air Communications in the L-band. In *Integrated Communications, Navigation and Surveillance Conference (ICNS)*. IEEE, Apr. 2021.

[EHS12] U. Epple, F. Hoffmann, M. Schnell. Modeling DME interference impact on LDACS1. In *2012 Integrated Communications, Navigation and Surveillance Conference*. IEEE, 2012.

[ESCG11] U. Epple, M. Schnell, G. A. Center (DLR), Germany. Overview of Interference Situation and Mitigation Techniques for LDACS. In *2011I EEE/AIAA 30th Digital Avionics Systems Conference*. IEEE, 2011.

[FLT20] L. Fisser, S. Lindner, A. Timm-Giel. Predictive scheduling and opportunistic medium access for shared-spectrum radio systems in aeronautical communication. In *2020 Deutscher Luft- und Raumfahrtkongress*. Oct. 2020.
doi:10.25967/530331

[IEE] IEEE 802.15.4-2020 - IEEE Standard for Low-Rate Wireless Networks.
https://standards.ieee.org/standard/802_15_4-2020.html

[LCBM12] D. Leith, P. Clifford, V. Badarla, D. Malone. WLAN channel selection without communication. *Computer Networks* 56(4):1424–1441, Mar. 2012.
doi:10.1016/j.comnet.2011.12.015

[LE12] S. C. Lo, P. Enge. Assessing the capability of distance measuring equipment (DME) to support future air traffic capacity. *NAVIGATION, Journal of the Institute of Navigation* 59(4):249–261, 2012.

[LFT20] S. Lindner, L. Fisser, A. Timm-Giel. Coexistence of Shared-Spectrum Radio Systems through Medium Access Pattern Learning using Artificial Neural Networks. In *2020 32nd International Teletraffic Congress (ITC 32)*. Pp. 165–173. IEEE, 2020.

[Lin21] S. Lindner. Code Release for Time- and Frequency-Domain Dynamic Spectrum Access: Learning Cyclic Medium Access Patterns in Partially Observable Environments. Sept. 2021.
https://doi.org/10.5281/zenodo.5195407

[MMRC20] S. Mosleh, Y. Ma, J. D. Rezac, J. B. Coder. Dynamic Spectrum Access with Reinforcement Learning for Unlicensed Access in 5G and Beyond. In *VTC2020-Spring*. P. 7. Antwerp, Belgium, 2020.

[PAG+13] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, T. Engel. On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH. *IEEE Sensors Journal* 13(10):3655–3666, 2013. Publisher: IEEE.

[TR20] S. Tomovic, I. Radusinovic. A Novel Deep Q-learning Method for Dynamic Spectrum Access. In *2020 28th Telecommunications Forum*. Pp. 1–4. Nov. 2020. doi:10.1109/TELFOR51502.2020.9306591

[WLGK17] S. Wang, H. Liu, P. H. Gomes, B. Krishnamachari. Deep reinforcement learning for dynamic multichannel access. In *International Conference on Computing, Networking and Communications (ICNC)*. Pp. 257–265. 2017.

[WLGK18] S. Wang, H. Liu, P. H. Gomes, B. Krishnamachari. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE Transactions on Cognitive Communications and Networking* 4(2):257–265, 2018. doi:10.1109/TCCN.2018.2809722