Hidden Markov Model Spectrum Predictor for Poisson Distributed Traffic

Rodrigo F. Bezerra, Jacir L. Bordim, Marcus V. Lamar, Marcos F. Caetano
Dept. Computer Science
University of Brasília
Brasília - DF, Brazil
rodrigo.bezerra@aluno.unb.br,{bordim,lamar,mfcaetano}@unb.br

***Abstract*—Static spectrum allocation policies allied with the increasing demand for higher data rates stimulated the pursuit of alternative spectrum allocation strategies. In this context, Opportunistic Spectrum Access (OSA) has been considered an alternative to allow licensed portions of the spectrum to be shared with unlicensed users. OSA requires unlicensed users to identify unused portions of the spectrum for opportunistic access that minimizes possible interference with the licensed users. Accurate mechanisms to avoid interference and improve the spectrum usage is highly desirable. This work investigates the performance of a traditional Hidden Markov Model (HMM) predictor where the licensed traffic follows a Poisson distribution. The results show that, under the evaluated settings, traditional HMM predictor can improve spectrum usage up to 15% at the expense of a high rate ($\approx 50\%$) of inaccurate forecasts of idle periods. Based on these results, this paper proposes two techniques to optimize prediction performance and reduce inaccurate forecasts rate. Using the proposed enhancements inaccurate forecasts rate was reduced to only 6%. An additional benefit was observed when reducing collision with the licensed user, that is an improvement in the amount of slots effectively used by the unlicensed users from 15% to 23%.**

***Index Terms*—Hidden Markov Model, opportunity forecasting, Poisson Distributed Traffic, Cognitive Radio, opportunistic spectrum access.**

I. Introduction

Numerous spectrum usage measurement studies have been conducted [1]–[3] in the past showing that some frequencies are barely or eventually occupied by the licensed users (a.k.a primary user (PU). As a consequence of the static allocation, where each service has one reserved channel, there is a spectrum shortage for new services. Cognitive Radio (CR) emerged as an alternative technology to deal with the spectrum scarcity problem as it enables nodes to perform Opportunistic Spectrum Access (OSA), allowing coexistence with other users in the same frequency without interference. Modeling PU's behavior is the key topic to achieve collision-free opportunistic communications. In order to characterize the PU traffic in a time-slotted system, spectrum sensing methods usually employ energy detection (ED) mechanisms [4]. Such strategies enable unlicensed secondary users (SU) to employ cognitive techniques to share the same frequencies with PUs and improve spectrum usage.

Hidden Markov Models (HMM) have been used to perform spectrum prediction in several network environments [5]–[9]. Most of these works predict the occupancy of one slot ahead, and do not indicate the duration of the idleness. That characteristic limits the usage performance of the available slots. In [5], [6], this problem is solved by using a different channel representation, where each symbol denotes the state and its duration. By doing that, the input for the HMM-based predictor consider the busy and idle duration during the training phase, allowing it to predict the duration of the next idle period. Machine learning techniques have been employed for spectrum prediction, such as Neural Networks (NN) [10]–[13] and linear prediction [14], [15]. As reported in [16], increasing the intermediate NN layers increases their computational complexity and accuracy. On the other hand, the complexity of an HMM predictor is lower than an NN predictor, making HMM an interesting alternative to predict spectrum opportunities.

HMMs have been used as a candidate for online predictor, which means all parameters can be, efficiently, updated before each prediction. Despite the better prediction accuracy of NNs, it has been used as an offline predictor as it requires prior training or retraining each time the PU characteristics changes. In this work, the performance of a traditional HMM predictor is evaluated for a PU that follows a Poisson distributed traffic. The HMM implementation is similar to the one proposed in [6]. Simulation results show that applying a traditional HMM incurs in a high number of incorrect predictions, which may prevent successful opportunistic usage. To overcome this limitation, this paper proposes two modifications to optimize the prediction performance and reduce collisions of an HMM-based predictor where the PU traffic is modeled as Poison process. More precisely, we remove under-trained HMMs from the predictor and apply a strategy, called "calibration", which reviews predictions with high probability of collision. Results show that the proposed enhancements allowed to reduce collision rate to less than 7%. Employing these techniques, we have obtained better accuracy and performance. As low collision rate is achieved, the number of used correctly identified opportunities increased from 15% to 23%.

The following sections are organized as described: Sec. II formally defines the HMM theory. Sec. III describes the training and prediction procedures. Sec. IV presents the obtained results using the traditional predictor. Sec. V proposes two techniques to improve performance. Finally, Sec. VI closes the article and presents the future work.

II. HMM Theory

In order to completely describe the prediction procedure, begin by introducing a formal definition of the HMM used in this work. For further details, we refer the interested reader to [16]–[18]. Let $Q \in \{1, 2, ..., \Omega\}$ be a variable denoting a hidden state, where Ω is the total number of states. A hidden state produces observation symbols, denoted by $E \in \{1, 2, ...\Gamma\}$, where Γ is the maximum value for each observation. As a result, a sequence of hidden states $Q_n = \{Q_1, Q_2, ..., Q_n\}$ generates a sequence of observations $E_n = \{E_1, E_2, ..., E_n\}$. Considering this, a Hidden Markov Model (HMM) can be defined as a tuple $\lambda = (\pi, A, E)$, where π is a matrix of initial probabilities with size $\Omega \times 1$, A is the transition matrix with size $\Omega \times \Omega$ defined as

$$A = \begin{vmatrix} a_{1,1} & ... & a_{1,\Omega} \\ ... & ... & ... \\ a_{\Omega,1} & ... & a_{\Omega,\Omega} \end{vmatrix} \quad (1)$$

where an element $a_{i,j} \in A$ represents the probability of being in the state i in a moment t and change to the state j in a moment $t+1$. E is the emission matrix of size $\Omega \times \Gamma$ and can be defined as

$$E = \begin{vmatrix} e_{1,1} & ... & e_{1,\Gamma} \\ ... & ... & ... \\ e_{\Omega,1} & ... & e_{\Omega,\Gamma} \end{vmatrix} \quad (2)$$

where element $e_{i,j}$ represents the probability of being in the state i and the observation symbol j is generated. Fig. 1 shows the topology of an HMM for a number of states $\Omega = 2$, Γ observation symbols for each state Q, and the respective transition probabilities $a_{i,j}$ and observation probabilities $e_{i,j}$.

III. HMM Spectrum Prediction

Spectrum scarcity have fostered the quest for alternatives to improve spectrum utilization. In this context, a primary user (PU) may grant channel access to secondary users (SU) under certain conditions. In general, it is expected that SUs should be capable of identifying spectrum opportunities based on spectrum or traffic analysis. This section describes the HMM prediction procedure used in this work. In what follows, we consider that the PU and SU operate over a single channel, where the SU is capable of continuously monitor this channel. Its main components are depicted in Fig. 2. As shown in the figure, the spectrum sensing module senses the frequencies that the radio is locked on and acquires information which is then passed to the length extraction module to extract features of the incoming traffic. These features are used initially for training the HMMs and then stored for latter spectrum opportunities forecast. A Secondary User (SU) may use the prediction procedure to obtain spectrum opportunities information to gain access to channel to transmit its data. The SU may resort to different medium control access (MAC) strategies coordinate with other SUs that may attempt to transmit concurrently. This latter step, however, is outside the scope of this paper. In what follows, we detail the main components of the prediction model.

A. Spectrum Sensing

The spectrum sensing module acquires channel information and produces a binary occupancy output $B \in \{0, 1\}$, which is stored for spectrum decisions [19]. In a common use of this representation 0 denotes that the channel is idle and 1 denotes that the channel is busy in a particular slot. An HMM defined with two hidden states is traditionally used to model the behavior of the aggregated PU traffic received by the SU [16]. The sequence B is obtained by the SU as an aggregated traffic received from the PU in its radio range. Since the SU has no knowledge of the traffic pattern, this is considered *hidden* in terms of HMM modeling.

B. Length Extraction

The *Length Extraction* module analyses the binary occupancy output and extracts the lengths of idle and busy states, forming the sequence $D = \{d_1, d_2, ..., d_k\}$, where k is the length of the sequence D. Being τ the maximum value observed for a single channel, I_v denote an idle sequence and B_v denote a busy sequence of length v, for $v \in \{1, ..., \tau\}$, respectively. So, an element in D can take either values I_v or B_v, creating a sequence $D = \{..., d_{n-1}, d_n, ...\}$, where $d_{n-1} \in \{\mathcal{I}_1, ..., \mathcal{I}_\tau\}$ and $d_n \in \{\mathcal{B}_1, ..., \mathcal{B}_\tau\}$. For a binary occupancy output sequence $B = \{00111111000000001110001111\}$, the length extraction module outputs a sequence $D = \{\mathcal{I}_2, \mathcal{B}_6, \mathcal{I}_7, \mathcal{B}_3, \mathcal{I}_3, \mathcal{B}_4\}$.

C. Training

Before training the system is unable to predict, so all elements in the output of *Length Extraction* are directed to the *training* block. The training module receives a representative sequence, called $D_{training}$, to setup the HMM parameters. For each idle length $\mathcal{I}_j$, $(1 \leq j \leq \tau)$ observed in $D_{training} \in \{1, ..., \tau\}$, an HMM is trained using at most $\mathcal{S}$, $(\mathcal{S} \geq 1)$ sequences with the same size. For the training purpose, Baum-Welch forward-backward algorithm [20] is used to set the parameters for all HMMs. The training process produces $\Sigma = \{\sigma_1, \sigma_2, ..., \sigma_\tau\}$ trained HMMs. Note that our interest is to predict idle states (i.e. spectrum opportunities), hence we concentrate this work to train HMM to recognize idle sequences.

D. Memory

Once trained, the output sequence D of the *Length Extraction* is directed to the *Memory* block. The sequence D is continuously available to this block, where the last l symbols from sequence D are stored in the Memory block. Lets say d_n is the last element of the sequence D in a instant n, then $D_n = \{d_{n-l}, ..., d_n\}$. In the same way, $D_{n+1} = \{d_{n-l+1}, ..., d_{n+1}\}$. As soon as the preceding block sends another idle symbol, the previous one is discarded.

E. Prediction Procedure

The prediction block receives the trained HMMs from the training module and a sequence D_n of length l, which are used to predict the length of the next opportunity (that

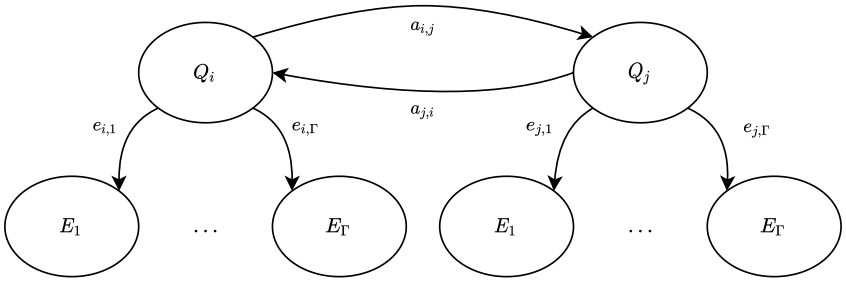

Fig. 1. HMM topology representing a number of states $\Omega = 2$.

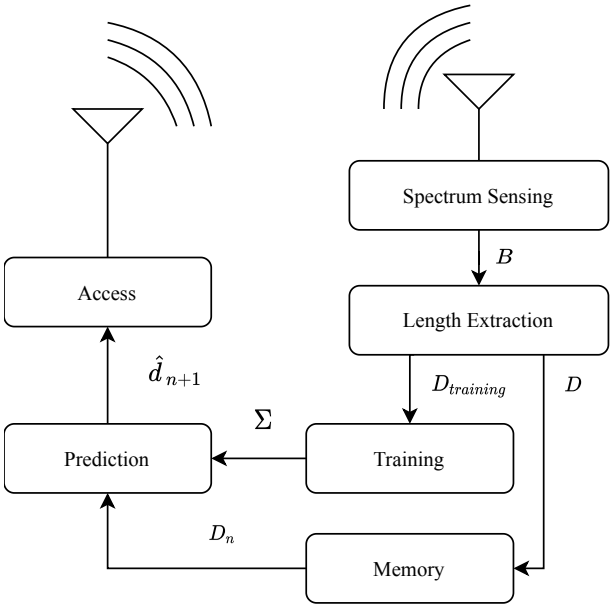

Fig. 2. Predictor system model.

is, d_{n+1}). The likelihood computation follows the procedure described in [6], [20]. The probability of the state sequence $Q_n = (Q_{n-l+1}, Q_{n-l+2}, ..., Q_n)$ occur, for an HMM with parameters λ, is given by

$$Pr(Q_n|\lambda) = \prod_{i=n-l+1}^{n} a_{i,i+1}, \tag{3}$$

where $a_{i,i+1}$ is the transition probability from state i to state j for the HMM λ. The next step is to calculate the likelihood of an observation sequence D_n for the state sequence Q_n, which is given by

$$Pr(D_n|Q_n, \lambda) = \prod_{i=n-l+1}^{n} Pr(D_i|Q_i, \lambda), \tag{4}$$

with $Pr(D_i|Q_i, \lambda)$ being the probability of the state Q_i emitting the symbol D_i. After this, the likelihood ζ of an observation sequence D_n with an HMM with parameters λ is given by the sum of $Pr(D_i|Q_i, \lambda)$ evaluated for all possible state sequences Q_n, defined as

$$\eta(D_n, \lambda) = \sum_{\forall Q_n} Pr(D_i|Q_i, \lambda). \tag{5}$$

This likelihood could be calculated using Baum-Welch forward procedure [20]. In order to obtain an estimation $\hat{d}_{n+1}$ for the next symbol d_{n+1}, the HMM with higher likelihood is chosen as the predicted symbol, defined as

$$\hat{d}_{n+1} = \underset{i=\{1,...,\tau\}, \lambda_i \in \Sigma}{\arg\max} \log \eta(d_n, \lambda_i). \tag{6}$$

IV. Simulation Results

This section presents numerical results obtained using the prediction procedure defined in Sec. III. We study the prediction performance with respect to observation length l and the number of training sequences $\mathcal{S}$ used in each HMM. Each HMM has been configured to use 2 hidden states. In order to characterize the PU behavior, five sets of the sequence D with 100,000 samples were generated based on Poisson distribution with mean of 15 slots. A slot, in the scope of this work, is the minimum period of time needed to determine channel occupancy by spectrum sensing. Each sample represents the duration of an idle or busy sequence of slots. A confidence interval of 95% was calculated and is properly represented in the following results. The channel occupation by the PU is around 50% in this scenario.

To evaluate the prediction performance two metrics were investigated in this work. Firstly, we calculate the mean-squared error (MSE) between the prediction and the available for each slot, defined as

$$MSE = \frac{1}{n} \sum_{i=1}^{k} (d_i - \hat{d}_i)^2, \tag{7}$$

where k is the number of predictions, d_i the length of available slots in the symbol i and $\hat{d}_i$ is the output prediction. This metric aim to measure how distant a prediction is from the available number of idle slots. The second metric is the collision rate, that measure how many predictions generate collisions with PUs, i. e., a prediction $\hat{d}_{n+1}$ is longer than

the available number of idle slots d_{n+1}. So, we can define a collision $P(n)$ as

$$P(n) = \begin{cases} 0 & \text{if } \hat{d}_{n+1} \leq d_{n+1} \\ 1 & \text{if } \hat{d}_{n+1} > d_{n+1} \end{cases}. \quad (8)$$

Therefore, for each $P(n)$ we can calculate the collision rate as

$$C = \frac{\sum_{i=1}^{n} P(i)}{n}. \quad (9)$$

Both metrics were evaluated varying the length of the evaluation sequence l and also varying the number of training sequences $\mathcal{S}$ provided for each HMM.

The prediction performance is shown in Fig. 3, varying the length of the evaluation sequence (as well as the training sequence) in the interval of $1 \leq l \leq 20$. As we can see, the MSE is around 95 for values $l \geq 7$. On the other hand, for values $l < 7$, the performance starts degrading, reaching 170 for $l = 2$. When l increases, collision rate remains in the same level, achieving the best value of 50.85% for $l = 10$.

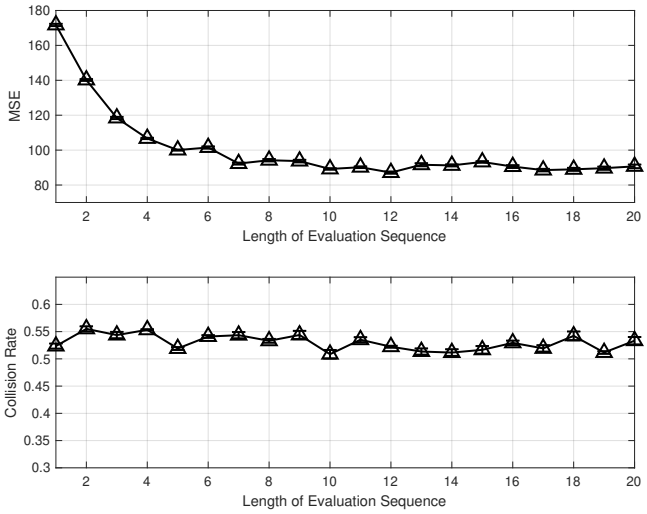

Fig. 3. MSE and Collision Rate varying the length of evaluation sequence l

The number of training sequences provided to the HMM also could influence the predictor performance. The results using from 10 to 100 sequences to train each HMM are shown in Fig. 4. The better results for both parameters were obtained using 50 sequences for training.

All idle slots in the elements of the sequence D can be grouped in three categories: idle slots that have been correctly identified by the HMM predictor; idle period which has not been identified by the HMM and are wasted; overestimated idle slots. This last category we call as "collision" and its use by the SU may interfere with the PU. Fig. 5, depicts the possible situations and represents each group of slots. Potentially used slots are the ones that would be used whether this prediction procedure is implemented and $\hat{d}_{n+1} \leq d_{n+1}$. As shown in the figure, when predicting $\hat{d}_{n+1} = \mathcal{I}_4$, four slots are potentially used. Potentially unused slots are the left slots

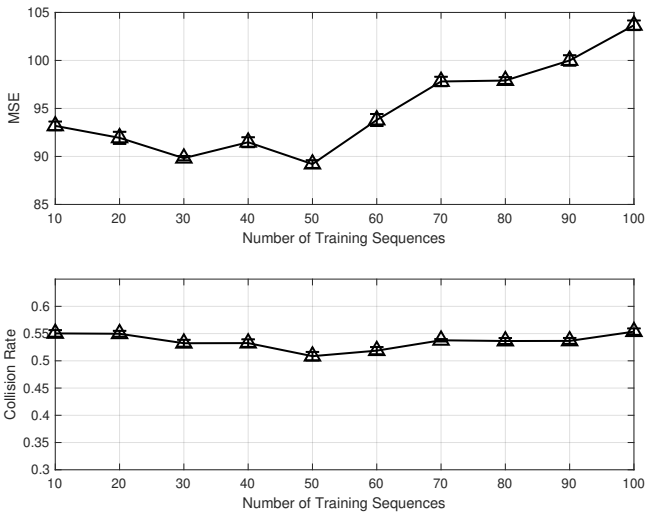

Fig. 4. MSE and collision rate for each quantity of training sequences.

in this idle period, i.e, tree out of seven are potentially unused. Finally, when the $\hat{d}_{n+1} > d_{n+1}$, all idle slots are counted as potential collisions.

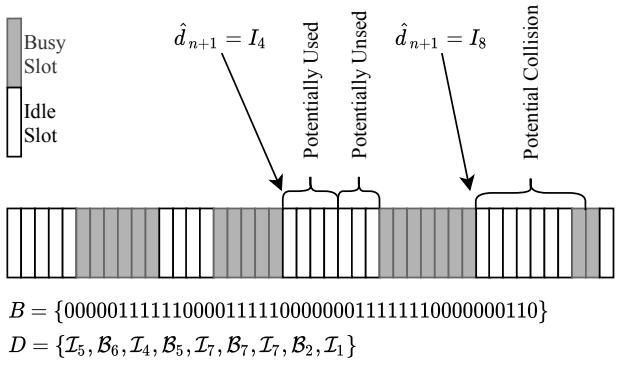

Fig. 5. Idle slots usage when prediction do not result in collision and when it results in collision.

Each idle or busy duration is denoted by one symbol that represents its type and duration, for example, in Fig. 5, $\mathcal{I}_5$ represents an idle period of five slots and $\mathcal{B}_6$ a busy period of six slots. As an initial analysis, we verified how collisions are distributed for each predicted idle duration. The probability of collision per predicted idle duration is shown in Fig. 6. Collisions occur mainly when $\hat{d}_{n+1} > \mathcal{I}_{15}$, and as can be seen in Table I, which represents 89.86% of all collisions. Predictions where $\hat{d}_{n+1} \leq \mathcal{I}_{15}$ generates 10.14% of all collisions. A residual amount of collisions occurred when the predicted symbol is $\hat{d}_{n+1} \leq \mathcal{I}_{10}$.

Based on the results presented in Fig. 6 and Table I, the most part of collisions occur when predicting $\hat{d}_{n+1} \geq \mathcal{I}_{15}$, therefore any performance improvement method must deal with those predictions. To do so, we have analyzed how changing those predictions affects MSE and collision rate.

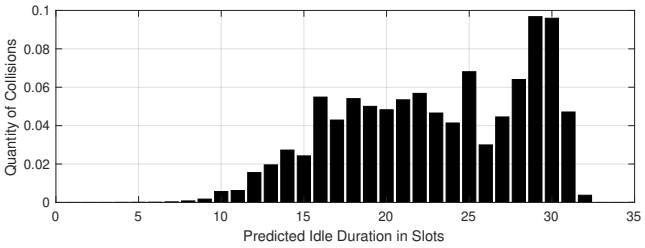

Fig. 6. Collisions per predicted idle duration in number of slots.

TABLE I
COLLISIONS PER PREDICTED IDLE DURATION IN THE HMM PREDICTOR

Predicted Idle Duration in Slots	Probability
$1 \leq \hat{d}_{n+1} \leq 10$	0.0086
$11 \leq \hat{d}_{n+1} \leq 15$	0.0928
$16 \leq \hat{d}_{n+1} \leq 20$	0.2502
$21 \leq \hat{d}_{n+1} \leq 25$	0.2663
$26 \leq \hat{d}_{n+1} \leq 30$	0.3312
$31 \leq \hat{d}_{n+1} \leq 35$	0.0509

V. ENHANCEMENTS

In order to lower MSE and collision rate presented in the Sec. IV, we have studied the effects of two changes: remove under-trained HMMs from the predictor and reducing prediction with high probability of collisions. We show that the benefits of reducing collisions also includes a higher usage of idle slots for the SU.

Under-trained HMMs in a predictor can lead to unbalanced probabilities, which means the predictor could be influenced by that HMM. So, training control aims to remove under-trained HMMs from the predictor, i.e., do not use HMMs for prediction of symbols $D_i \in \{1, ..., \tau\}$ that has less than $\mathcal{S}$ occurrences in $D_{training}$. The results using the modified predictor, where all under-trained HMMs were removed, are depicted in Fig.7. This change has reduced the collision rate and MSE for all length of evaluation sequence, achieving the better result using $l = 10$. For $1 \leq l \leq 6$, a linear behavior of MSE was observed when using the modified predictor when the traditional predictor expressed an exponential characteristic. When the evaluation sequence becomes shorter, the predictor becomes more susceptible to unbalanced probabilities. Besides the reduction on the collision rate, the MSE reduced to less than 30% (best case), meaning that the predictions are much closer to the current idle duration.

A. Calibration

In this section we analyzed the influence of the predictions with high probability of collision. Let us consider an initial prediction of $\hat{d}_{n+1}$. Then, if $\hat{d}_{n+1} \geq c$, where $c > 0$, is constant calibration value, the new prediction for the next slot becomes $\hat{d}_{n+1} - c$. Otherwise, the prediction remains unchanged $\hat{d}_{n+1}$. The results are shown in Fig. 8. The calibration was able to reduce the collision rate up to 6.4% with $c = 13$, in contrast with 50% using the traditional predictor. When

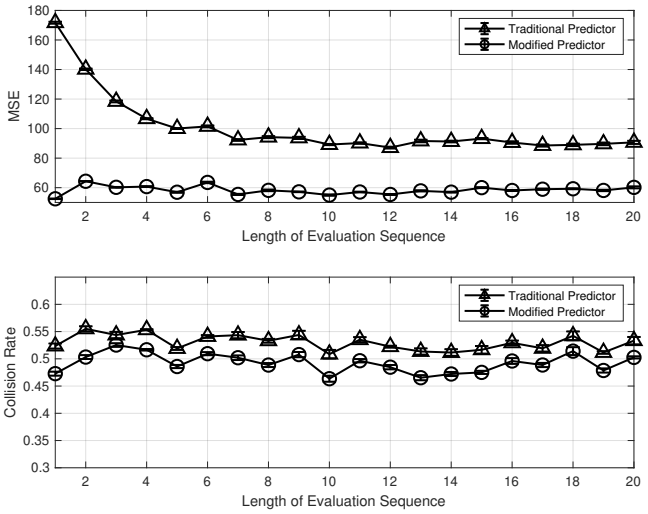

Fig. 7. MSE and collision rate for predicting using under-trained HMMs and removing under-trained HMMs.

$c > 13$, the overall performance starts degrading. Despite the increase in the MSE, no side effects were noted. Using calibration with the traditional predictor also reduce collision rate, but the better results were achieved when using the two proposed techniques simultaneously. A trade-off could be established here. When SU observes high collision rate, it could increase c in order to reduce collisions. SU should increase c until collision rate increase, achieving the better calibration.

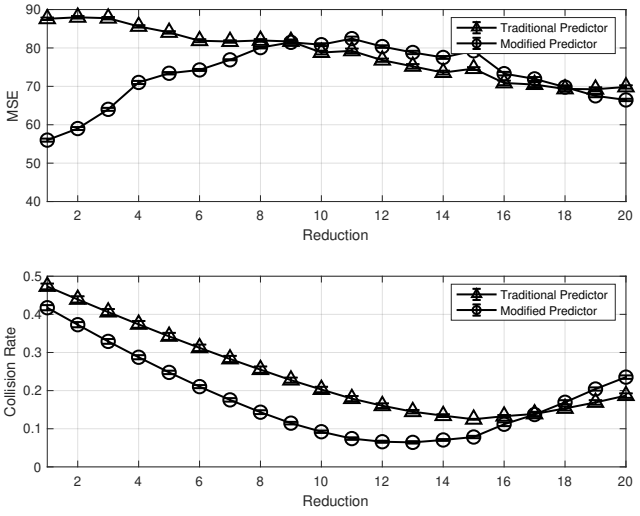

Fig. 8. Reducing predictions for symbols with high collision probability.

Beyond the advantage obtained in the collision rate, remove under-trained HMMs and applying calibration brings another improvement for the SU. As presented in Fig. 9, the number of potentially used slots has increased reached 47% of the available idle slots, against 30% without any modifications (upper graph). Another improvement is shown in the bottom graph,

where the number of slots classified as potential collision was reduced from 46% to 9%. This leads to energy consumption reduction as SU stops colliding with PUs. The middle graph shows that potentially unused slots has also increased, but it has a neutral effect since SU and PU are not affected when it occurs.

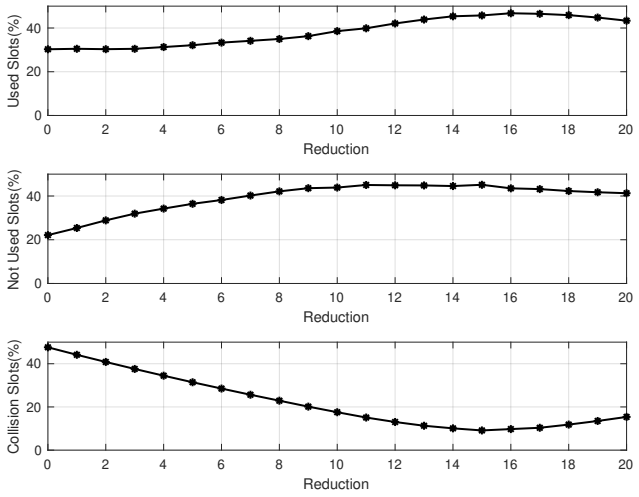

Fig. 9. Slot usage overview.

VI. Conclusion

In this work a traditional HMM predictor is used to perform opportunity forecast in wireless networks. This predictor aim to identify white spaces that may be used for opportunistic access. We consider that the PU behavior follows a Poisson distribution. Simulation results show that applying a traditional HMM incurs in a high number of incorrect predictions, which may prevent successful opportunistic usage. To overcome this limitation, this paper proposes two modifications to optimize the prediction performance and reduce collisions of an HMM-based predictor where the PU traffic is modeled as Poison process. More precisely, we removed under-trained HMMs and used a "calibration strategy", which consists in review predictions with high probability of collision, to reduce the collision rate to less than 7%. The results have shown that reducing collisions also improves the amount of slots effectively used by the SU. This may help reducing SU's energy expenditure. We also established an optimization procedure to find the optimal parameters for training and prediction in an HMM-based predictor. Future work will consider other traffic patterns and analyze the impact of channel occupation on the prediction performance. For comparison purposes, future work will also consider the evaluation of other methods of prediction (such as NN) for a performance-complexity analysis.

Acknowledgment

This work was partially supported by the University of Brasília (UnB), Department of Computer Science (CIC/UnB), Dean of Postgraduate (DPG/UnB) and Dean of Research and Innovation (DPI/UnB).

References

[1] M. Höyhtyä, A. Mämmelä, M. Eskola, M. Matinmikko, J. Kalliovaara, J. Ojaniemi, J. Suutala, R. Ekman, R. Bacchus, and D. Roberson, "Spectrum occupancy measurements: A survey and use of interference maps," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 2386–2414, Fourthquarter 2016.

[2] Y. Chen and H. Oh, "A survey of measurement-based spectrum occupancy modeling for cognitive radios," *IEEE Communications Surveys Tutorials*, vol. 18, pp. 848–859, Firstquarter 2016.

[3] O. Holland, H. Kokkinen, S. Wong, V. Friderikos, A. Raman, M. Dohler, and M. Lema, "Changing availability of tv white space in the uk," *Electronics Letters*, vol. 52, no. 15, pp. 1349–1351, 2016.

[4] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE Communications Surveys Tutorials*, vol. 11, pp. 116–130, First 2009.

[5] A. Saad, B. Staehle, and R. Knorr, "Spectrum prediction using hidden markov models for industrial cognitive radio," in *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp. 1–7, Oct 2016.

[6] A. Saad, H. F. Schepker, B. Staehle, and R. Knorr, "Whitespace prediction using hidden markov model based maximum likelihood classification," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pp. 1–7, April 2019.

[7] I. A. Akbar and W. H. Tranter, "Dynamic spectrum allocation in cognitive radio using hidden markov models: Poisson distributed case," in *Proceedings 2007 IEEE SoutheastCon*, pp. 196–201, March 2007.

[8] C. Park, S. Kim, S. Lim, and M. Song, "Hmm based channel status predictor for cognitive radio," in *2007 Asia-Pacific Microwave Conference*, pp. 1–4, Dec 2007.

[9] H. Eltom, S. Kandeepan, B. Moran, and R. J. Evans, "Spectrum occupancy prediction using a hidden markov model," in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–8, Dec 2015.

[10] P. A. L. Ferreira, S. S. Fernandes, R. R. Bezerra, M. V. Lamar, and J. L. Bordim, "Recurrent neural networks for transmission opportunity forecasting," in *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pp. 382–383, Oct 2016.

[11] A. F. Cattoni, M. Ottonello, M. Raffetto, and C. S. Regazzoni, "Neural networks mode classification based on frequency distribution features," in *2007 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications*, pp. 251–257, Aug 2007.

[12] S. Mohammadjafari, E. Kavurmacioglu, J. Maidens, and A. Bener, "Neural network based spectrum prediction in land mobile radio bands for iot deployments," in *2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pp. 31–36, April 2019.

[13] M. F. Caetano, M. R. Makiuchi, S. S. Fernandes, M. V. Lamar, J. L. Bordim, and P. Solís Barreto, "A recurrent neural network mac protocol towards to opportunistic communication in wireless networks," in *2019 16th International Symposium on Wireless Communication Systems (ISWCS)*, pp. 63–68, Aug 2019.

[14] A. Eltholth, "Forward backward autoregressive spectrum prediction scheme in cognitive radio systems," in *2015 9th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pp. 1–5, Dec 2015.

[15] M. T. Ozden, "Adaptive multichannel sequential lattice prediction filtering method for cognitive radio spectrum sensing in subbands," in *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems*, pp. 961–968, Oct 2012.

[16] G. Ding, Y. Jiao, J. Wang, Y. Zou, Q. Wu, Y. Yao, and L. Hanzo, "Spectrum inference in cognitive radio networks: Algorithms and applications," *IEEE Communications Surveys Tutorials*, vol. 20, pp. 150–182, Firstquarter 2018.

[17] W. Zucchini, I. Macdonald, and R. Langrock, *Hidden Markov Models for Time Series - An Introduction Using R*. 06 2016.

[18] W. Khreich, E. Granger, A. Miri, and R. Sabourin, "A survey of techniques for incremental learning of hmm parameters," *Information Sciences*, vol. 197, pp. 105 – 130, 2012.

[19] J. Mitola and G. Q. Maguire, "Cognitive radio: making software radios more personal," *IEEE Personal Communications*, vol. 6, pp. 13–18, Aug 1999.

[20] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. USA: Prentice-Hall, Inc., 1993.