

Performance Analysis of Long Short-Term Memory-Based Markovian Spectrum Prediction

NIRANJANA RADHAKRISHNAN¹,
SITHAMPARANATHAN KANDEEPAN¹, (Senior Member, IEEE), XINGHUO YU¹, (Fellow, IEEE),
AND GIANMARCO BALDINI², (Senior Member, IEEE)

¹School of Engineering, RMIT University, Melbourne, VIC 3000, Australia

²European Commission, Joint Research Centre, 21027 Ispra, Italy

Corresponding author: Niranjana Radhakrishnan (s3752128@student.rmit.edu.au)

This work was supported by RMIT Research Stipend Scholarship (RRSS-SC).

ABSTRACT Dynamic Spectrum Access (DSA) solutions equipped with spectrum prediction can enable proactive spectrum management and tackle the increasing demand for radio frequency (RF) bandwidth. Among various prediction techniques, Long Short-Term Memory (LSTM) is a deep learning model that has demonstrated high performance in forecasting spectrum characteristics. Although well-performing, the theoretical characterization of LSTM prediction performance has not been well developed in the literature. Therefore, in this article, we examine an LSTM based temporal spectrum prediction model and characterize its prediction performance through theoretical analysis. To this end, we analyze the LSTM prediction outputs over simulated Markov-model-based spectrum data and spectrum measurements data. Our results suggest that the predicted scores of the LSTM based system model can be described using mixtures of truncated Gaussian distributions. We also estimate the performance metrics using the mixture model and compare the results with the observed prediction performance over simulated and measured datasets.

INDEX TERMS Spectrum prediction, long short-term memory, probability of error, performance modeling.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

CURRENT regulatory measures for spatial and temporal allocation of radio frequency (RF) spectrum have inadequacies that result in the under-utilization of frequencies and the inability to manage the growing bandwidth demand due to an abundance of devices and applications [1], [2]. Dynamic Spectrum Access (DSA) solutions are widely adopted to ensure efficient spectrum utilization, as they allow the opportunistic usage of spectrum by unlicensed users when licensed users, known as Primary Users (PU), are inactive. Cognitive Radio (CR) is a technology that enables intelligent wireless communications and realizes DSA functionalities. Unlicensed users, known as Secondary Users (SU), adopt CRs to perform cognitive functions that are broadly related to radio environment perception, reconfigurability, and learning. Specifically, these functions include spectrum sensing, spectrum decision, spectrum sharing, and spectrum mobility

The associate editor coordinating the review of this manuscript and approving it for publication was Ding Xu¹.

and can help utilize resources more efficiently [3]. However, cognitive functions incur significant time delays, which can cause transmission collisions or even degrade the spectrum utilization efficiency [2], [4].

Predicting the spectrum usage characteristics is recognized as an important step towards managing spectrum more efficiently in DSA scenarios. Spectrum prediction techniques infer unmeasured characteristics such as channel occupancy status, duty cycle, and signal power level based on historical spectrum data [5], [6]. These techniques leverage the correlations in the data to make future predictions in the time, frequency, and spatial domains. The predicted results can be applied to identify spectrum holes and infer spectrum evolution. These techniques can also help minimize delays, reduce energy consumption, and improve the system throughput [4], [6].

Time-domain or temporal spectrum prediction typically involves techniques such as Auto-Regressive (AR) time series models, Markov chain based models, Bayesian inference, Machine Learning (ML) models, and Deep Learning (DL) models [6]. However, many conventional prediction methods

require a priori information to make predictions and cannot effectively capture the long-term dependencies in the data [7]. In contrast, Long Short-Term Memory (LSTM) is a deep learning model that learns the long-term and short-term dependencies in the input data. LSTM hidden units consist of memory cells that maintain the state variables and non-linear gating units that control the update mechanism of the hidden unit. It is a Recurrent Neural Network (RNN) variant that performs well with sequence data and has been adopted to perform modulation classification, interference classification, and spectrum prediction in various studies [8].

LSTMs have shown good performance in spectrum prediction applications related to LTE mobile traffic prediction, cognitive aerospace communications, vehicular communications, machine-type communications etc [9], [10]. LSTM has also shown improved prediction performance compared to other techniques such as Auto-Regressive Integrated Moving Average (ARIMA), Feed-Forward Neural Networks etc., in various spectrum prediction works [9], [11], [12]. Although various works establish the superior capabilities of LSTM models to leverage correlations in the observations and predict more accurately, the theoretical analysis and insights regarding predictions have not been well developed in the literature. These aspects motivate us in this article to extensively analyze the prediction performance of LSTM for the temporal prediction of the future spectrum occupancy status.

B. LITERATURE REVIEW

Recent years have seen significant progress in spectrum prediction with a variety of predictive solutions that adopt algorithms such as clustering analysis, Hidden Markov Models (HMM), RNNs, and so on [13]. Conventional spectrum prediction methods fail to capture long-term temporal correlations in the data and require a priori information about the PU activity distribution to make predictions [7]. However, deep learning models such as LSTMs are capable of learning long-term and short-term correlations in the data and do not require a priori information to make predictions [6]. LSTMs have also consistently shown very high prediction performance for various spectrum prediction applications such as frequency hopping sequence prediction, cognitive aerospace communications, vehicular communications, etc., with simulated and measured data [14]–[16]. For instance, in [11], an LSTM based framework predicts the Power Spectral Density (PSD) values over measured datasets from terrestrial and satellite networks. The classification accuracy of the LSTM model is found to be high and stable compared to that of a Multi-Layer Perceptron (MLP) network, over a wide range of frequencies. In [12], a 2D-LSTM based spectral-temporal model predicts the PSD values with better precision, recall, and F1-score compared to the baseline AR, Bayesian Inference, and 1D-LSTM models over different frequency ranges of a measured spectrum dataset. In [17], a spectrum access strategy integrated with an LSTM based time-frequency prediction model achieves higher system throughput and reduces

energy consumption. The LSTM model proposed in this work shows higher prediction accuracy than the baseline HMM and MLP models. However, performance metrics such as Root Mean Square Error (RMSE) are not examined in this work.

The impact of parameters such as nature of the channel, input structure, number of future prediction steps, and model hyperparameters on LSTM prediction performance is also studied in many spectrum prediction works. The prediction accuracy of HMM, MLP, LSTM, and a proposed Reinforcement Learning (RL) based model with trained MLP sub-models are compared for bursty ISM band spectrum data in [18]. The proposed RL based MLP method adaptively takes action as per the bursty input. The results suggest that, on average, the proposed RLMLP model shows higher prediction accuracy than the LSTM model for bursty spectrum data. In [5], a Convolutional LSTM spectrum prediction model is developed to perform long-term joint spatial-temporal-spectral prediction of signal power level over measured UHF bands. Based on the results, the proposed model shows stable RMSE performance over different channels but shows degradation as the number of future time steps increases. A more sophisticated Convolutional LSTM based model, STS-PredNet, is proposed in [19], which shows a stable RMSE performance for multi-time step spatial-temporal-spectral prediction, with no degradation. In [20], cooperative spectrum prediction using LSTM local predictors is adopted for multi-time step prediction in a heterogeneous CR environment. The prediction performance of the local predictor is investigated in terms of the mean prediction error and the total probability of prediction error for a measured spectrum dataset over UHF bands. The relation between the performance metrics and parameters such as the window size and cooperative range are also studied. However, a detailed analysis of the total probability of error is not attempted.

Although high performing, LSTMs could pose challenges in terms of the computational burden and handling missing data. For spectrum situation prediction in a dense wireless ecosystem in [21], deep learning models such as Deep Neural Network (DNN), Convolutional Neural Network (CNN), and LSTM are trained with RF measurement traces of complicated coexistence scenarios. These models predict the type, form, and number of users in a frequency band for the test data. The results in [21] suggest that LSTMs show better prediction accuracy for large input window sizes but also consume more than 5x the training time of DNN and CNN models. Similarly, an LSTM based spectrum prediction model is implemented on a Field Programmable Gate Array (FPGA) for resilient military communication applications [22]. The models are trained and tested over measured IEEE 802.11p frames. The results suggest that the LSTM model outperforms the baseline ARIMA, Moving Average, and Naïve models even at low precision of number representation. Moreover, higher accuracy is demonstrated by floating-point implementations of the LSTM, but at the cost

of higher latency and resource utilization. A similar trend is observed in [23], where an LSTM based spectrum sensing framework shows high detection performance and accuracy under a low Signal-to-Noise Ratio (SNR) regime, at the cost of increased training time.

To reduce the computational overhead associated with LSTM based models, different methods have been proposed in the literature. For instance, to reduce the computational cost and improve the scalability of an LSTM based model, the dimensionality of spectrum data is reduced using tensor decomposition in [7]. Their proposed method also achieves lower normalized prediction error than the baseline models such as AR, Support Vector Machine (SVM), CNN, and LSTM models. To reduce the computational overhead due to training, an improved initialization method that enables fast LSTM training convergence compared to traditional initialization methods is proposed for spectrum prediction over Markovian data in [24]. Finally, in regards to handling missing data, a tensor completion based method is proposed in [25] to recover missing spectrum data values from a corrupted measured spectrum dataset. The proposed method in tandem with an LSTM predictor shows the highest prediction accuracy over a recovered data sequence, compared to other models such as K-Nearest Neighbours (KNN), Seasonal ARIMA (SARIMA), and MLP.

As discussed in this section, several research papers have adopted LSTM models for spectrum prediction and established various aspects such as their higher computational overhead, superior prediction capabilities compared to other baseline models, etc. However, the theoretical characterization related to LSTM prediction have not yet been well developed. Such a theoretical analysis could provide insights into LSTM prediction performance under different scenarios such as cooperative prediction, and also help to build improved prediction models. Therefore, in this article, we present novel insights that extend the theoretical understanding of LSTM based spectrum prediction. We analyze the LSTM prediction outputs for a Markovian spectrum occupancy data and propose a mixture model based analytical form of prediction performance metrics such as the probability of error. We also verify the results over simulated and RF measurements data from a realistic environment.

C. CONTRIBUTIONS

In our work, we assume that a wireless sensor or CR senses a channel at a fixed location over time. The data captured by the sensor is generated using a two-state discrete-time Markov model with the states representing Idle (0) and Busy (1) spectrum status. We adopt an LSTM based system model with a sequence input layer, an LSTM layer, a dense layer, a softmax layer, and a classification layer to perform the prediction. To analyze the predicted results, we train the LSTM based model with multiple independent Markov input sequences and extract the histograms of the prediction scores at the softmax layer output. Based on the results, we propose a mixture of truncated Gaussian distributions to model

the prediction score outputs and estimate the performance, such as the probability of error, using the proposed mixture distribution. Finally, we compare the theoretical and observed values of the performance metrics over simulated data and RF measurement data from a realistic environment. To sum up, our key contributions in this article are:

- *Modelling the prediction score of output classes:* We conduct a statistical analysis of the predicted outputs of the LSTM based model for a Markovian input. We train an LSTM based system model with multiple independent inputs and obtain the histograms of the prediction scores of the model. We propose that the prediction scores of the LSTM based system model for the output classes can be modelled well using a mixture of truncated Gaussian distributions.
- *Analysis of Prediction Performance Metrics:* We propose an analytical expression for the probability of error of the LSTM model, based on the mixture model proposed. Further, we verify the proposed expression on simulated and measured spectrum datasets, and also compare with the observed mean prediction errors. We also estimate other prediction performance metrics using the proposed model and compare them with the observed values.

The remainder of this article is organized as follows. Section II discusses the system model adopted. The modelling and error analysis are discussed in sections III and IV respectively. Section V presents the model verification and numerical results. Section VI concludes this article.

II. SYSTEM MODEL

In this work, we assume a SU equipped with a wireless sensor or a CR that senses a channel temporally at a fixed location. The sensed observations are in the form of a binary sequence representing the PU activity, i.e., the occupancy status of the channel by the PU. To perform spectrum prediction, we adopt an LSTM based model for the SU. This model learns the parameters for making forecasts from historical observations and predicts the spectrum occupancy status for a single future time step based on the current spectrum occupancy observation.

Figure 1 shows the LSTM based model adopted by the SU for spectrum prediction. At time instant t , the model in fig. 1 processes an input, x_t and makes a prediction of the output class, \hat{y}_t . The input, x_t , corresponds to the current spectrum occupancy observations at the SU, and the predicted output, \hat{y}_t , represents the future occupancy status. The sequence input layer passes x_t to the LSTM layer, which learns the temporal correlations and patterns in x_t . The LSTM layer output is then mapped to the prediction results by a dense layer. A softmax layer computes the prediction scores for each output class based on the dense layer activations, and the classification layer computes the cross-entropy loss during training.

In sections II-A - II-D, we explain the PU activity model and the LSTM based system model in more detail.

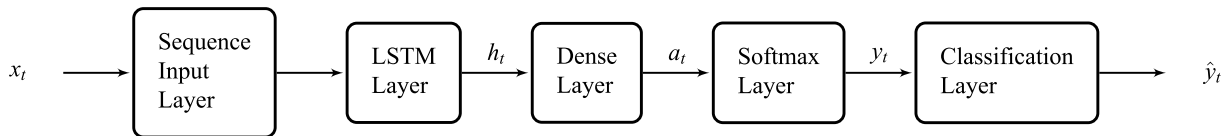


FIGURE 1. System model.

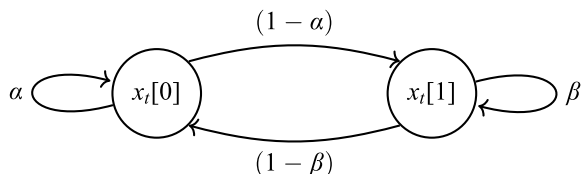


FIGURE 2. Spectrum occupancy model using a two-state Markov process.

A. INPUT DATA MODEL

Spectrum usage models are often adopted to represent PU activity and directly impact the SU system performance in DSA solutions [6]. These models are used to evaluate spectrum prediction techniques and are classified as time, frequency, and spatial domain models. Temporal spectrum usage models include Continuous-Time Markov Chains (CTMC) and Discrete-Time Markov Chains (DTMC) [6]. DTMC models are widely adopted spectrum usage models used to evaluate system performance metrics such as throughput and packet delay. These models can reproduce statistical properties associated with PU activity in realistic environments [6], [26]. Depending on the stationarity of the system, a stationary or non-stationary DTMC can be adopted [6].

A DTMC is fully characterized by the probability of the initial state and its transition probability matrix, P [27]. A two-state DTMC with states, $x_t[0]$, $x_t[1]$, and a transition probability matrix, P , as in (1), is diagrammatically shown in fig. 2.

$$P = \begin{pmatrix} \alpha & (1 - \alpha) \\ (1 - \beta) & \beta \end{pmatrix} \quad (1)$$

For the future time instant, $(t + 1)$, the system remains in the current state, $x_t[0]$, with probability, α , and transits with probability, $(1 - \alpha)$. Similarly, at time, $(t + 1)$, the system remains in the current state, $x_t[1]$, with probability, β and transits to state, $x_t[0]$, with probability, $(1 - \beta)$.

Our study assumes that the PU spectrum occupancy follows a first-order, two-state DTMC process with a binary state space, $S = \{0, 1\}$. State 0 stands for $x_t[0]$ in fig. 2, and indicates an idle channel with no traffic. Similarly, state 1 stands for $x_t[1]$ in fig. 2, and indicates that the PU occupies the channel. Further, we assume the initial state to be 0, and we limit the scope of our work to cases where α and β are equal in (1). A similar study can be extended to $\alpha \neq \beta$ cases as well.

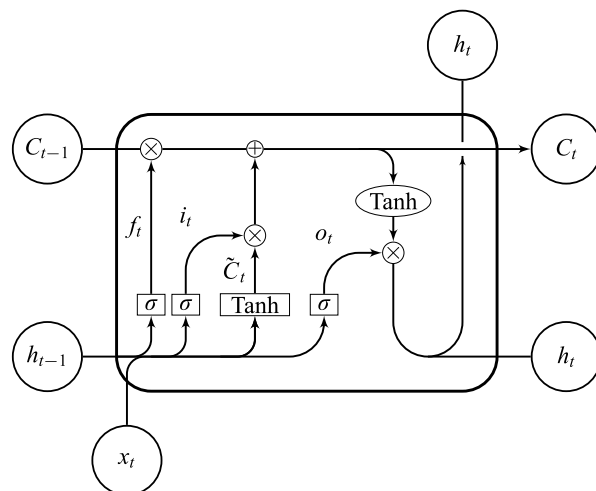


FIGURE 3. A variant of an LSTM hidden unit.

B. LSTM LAYER

Long Short-Term Memory (LSTM) is a RNN architecture that has become state-of-the-art for learning problems involving sequence data such as speech recognition, handwriting recognition, and language modelling [28]. The LSTM cell structure allows long-term retention of information and eliminates the vanishing gradients problem common to vanilla RNNs [8]. An LSTM cell or hidden unit consists of a memory cell and a few non-linear gating units, as shown in fig. 3. The memory cell stores the information about the context or history of the inputs as an internal cell state, C_t , and the gates control the update mechanism of C_t . The forget gate controls the removal or retention of information within the memory cell, as in (2).

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \quad (2)$$

The input gate activations and the cell candidate gate activations are used internally to compute the update values for C_t , as in (3), (4).

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \quad (3)$$

$$\tilde{C}_t = \tanh(w_C \cdot [h_{t-1}, x_t] + b_C) \quad (4)$$

The updated cell state is computed by combining the activations of the forget gate, input gate and cell candidate gate as in (5), where, $*$ represents the element-wise product of matrices.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

Finally, the output gate controls the flow of C_t for the computation of the cell output, h_t , as in (6), (7).

$$o_t = \sigma(w_o.[h_{t-1}, x_t] + b_o) \quad (6)$$

$$h_t = o_t * \tanh(C_t) \quad (7)$$

In (2) - (4) and (6), the weight parameter matrices of the forget gate, input gate, cell candidate gate and the output gate are denoted as w_f , w_i , w_C , and w_o , respectively. The bias parameter vectors of the gates are represented as b_f , b_i , b_C , and b_o , respectively. The gate activation vectors are represented by f_t , i_t , \tilde{C}_t , and o_t . These gate activations are computed as the matrix-vector product (.) between the corresponding weight parameter matrix and a concatenated vector of the past cell output, h_{t-1} , with the current cell input, x_t , denoted as $[h_{t-1}, x_t]$. The respective bias parameter vector is added to the computed product and the result is passed onto a gate activation function. The forget gate, input gate, and output gate have a sigmoid activation function represented by σ , whereas, the cell candidate gate has the tanh activation function. In our implementation, we adopt the LSTM layer with one LSTM cell.

C. DENSE LAYER

The LSTM layer activation vector, h_t , is passed onto a dense or fully-connected layer, as in fig. 1. This layer consists of dense layer hidden units that compute the layer output, a_t , as in (8),

$$a_t = w_d.h_t + b_d \quad (8)$$

Here, the matrix-vector product of the weight parameter matrix, w_d , with the input, h_t , is computed and added with the bias parameter vector, b_d . In our implementation, the dense layer has two hidden units. The rationale behind this choice is that the dense layer hidden units will map the LSTM activations to the two possible output classes, viz. idle and busy.

D. SOFTMAX LAYER

In the system model of fig. 1, the dense layer output, a_t , is passed onto a softmax layer that applies the softmax activation function on a_t . The softmax and classification layers are adopted for classification tasks, where the classification layer computes the cross-entropy loss during model training. In this study, we represent the softmax function output as y_t , which are the scores predicted by the LSTM based model for the output classes, given the input, x_t . The prediction score, y_t , is given by,

$$y_t(i) = \text{softmax}(a_t(i)) = \frac{\exp(a_t(i))}{\sum_j \exp(a_t(j))} \quad (9)$$

where $a_t(i)$ and $y_t(i)$ are the i^{th} element of these vectors, respectively. In our implementation, y_t contains two elements corresponding to each output class. To reflect the meaning of the output classes, we adopt the indexing convention as,

$i = 0, 1$ and $j = 0, 1$. Additionally, due to the nature of the softmax function, y_t has the properties given by,

$$0 \leq y_t(i) \leq 1 \quad (10)$$

$$\sum_i y_t(i) = 1 \quad (11)$$

III. MODELLING THE DISTRIBUTIONS OF LSTM OUTPUTS FOR SPECTRUM PREDICTION

As explained in section II, the system model of fig. 1 processes a stochastic input, x_t by passing it through neural network layers. Each layer processes its inputs and produces a corresponding layer activation or output. To analyze the performance of the spectrum prediction model, we consider the testing phase of a trained LSTM based model. During this phase, the weight and bias parameters are assumed to be constants.

As the input x_t is a stochastic input in our system model implementation, we can consider the LSTM layer output, h_t , as a random variable. We assume that h_t follows a probability density function (PDF), $f_H(h_t)$, denoted as,

$$h_t \sim f_H(h_t) \quad (12)$$

Similarly, the cumulative distribution function (CDF) of h_t is denoted as $F_H(h_t)$. Following the LSTM layer, the dense layer computes its output using h_t as given by (8). Using (8) and (12), the PDF and CDF of the dense layer output, a_t , can be obtained as,

$$f_A(a_t(i)) = \frac{1}{|w_d(i)|} f_H\left(\frac{a_t(i) - b_d(i)}{w_d(i)}\right) \quad (13)$$

$$F_A(a_t(i)) = F_H\left(\frac{a_t(i) - b_d(i)}{w_d(i)}\right) \quad (14)$$

Here, i can take values viz. 0, 1. Similarly, using (9), the CDF of the softmax output predicted scores can be expressed in terms of $F_H(h_t)$ as,

$$F_Y(y_t(i)) = F_H\left(\frac{1}{\Delta w_d(i)} \left[\ln\left(\frac{y_t(i)}{1 - y_t(i)}\right) + \Delta b_d(i) \right]\right) \quad (15)$$

where, Δw_d is defined as,

$$\Delta w_d(i) = \begin{cases} w_d(0) - w_d(1) & \text{for } i = 0 \\ w_d(1) - w_d(0) & \text{for } i = 1 \end{cases} \quad (16)$$

Similarly, Δb_d is defined as,

$$\Delta b_d(i) = \begin{cases} b_d(0) - b_d(1) & \text{for } i = 0 \\ b_d(1) - b_d(0) & \text{for } i = 1 \end{cases} \quad (17)$$

The definitions in (16) and (17) are adopted to concisely express the distribution in (15).

Considering (15) - (17), arriving at a closed-form expression for the distributions of h_t , a_t , and y_t is challenging because the LSTM cell consists of non-linear operations with feedback, resulting in a higher-order system. As an analytical approach to determine the distribution of the LSTM outputs is non-trivial, we adopt a modelling approach as explained in later sections.

A. MODELLING METHODOLOGY OF LSTM OUTPUTS

In this study, spectrum prediction is performed as a classification problem in supervised learning. The system model in fig. 1 is trained with the input data generated as per the two-state DTMC model in section II-A. To train the LSTM model, a binary input sequence, S , is generated for $T = 10000$ time steps and reshaped into the training input matrix, D , of dimensions $(T - 2) \times 2$. This reshaping is according to a sliding window of size, 2, and overlap, 1. The row entries of D corresponding to time step, t , are the past and present binary occupancies, viz. $S(t - 1)$, and $S(t)$. The training label corresponding to the input, $D(t)$ is the categorical occupancy value, i.e., *Busy* or *Idle*, corresponding to $S(t + 1)$.

In our system model of fig. 1, the LSTM layer consists of an LSTM cell, and the dense layer consists of two hidden units. The weight and bias parameters of the system model layers are initialized as per conventional initialization schemes [29], [30]. The optimizer algorithm is selected as ADAM optimizer during training. The model hyper-parameters, such as the learning rate, number of epochs, etc., are systematically tuned to achieve a suitable reducing trend of the training cost curve. For instance, to select the learning rate, training was conducted for different values of this parameter. Initially, a learning rate value of 1 was adopted and systematically scaled down to a value of 0.005, while observing the training cost curve. Among the different values, a learning rate of 0.05 resulted in a suitable reducing trend of the training losses. The remaining hyper-parameters of the system were selected in a similar manner to achieve a suitable reducing trend of the training costs. Overall, these steps are hereafter referred to as a training run. The trained LSTM based system model is tested with an independent Markovian test input with the same α value. This testing procedure is referred to as a test run hereafter.

In order to characterize the prediction performance of the LSTM model, fig. 4 shows the framework adopted for modelling the LSTM outputs. As in fig. 4, N independent test runs are conducted for the LSTM system model, and the predicted output sequences, y_t , are analyzed. Further, the average histograms of the predicted scores, y_t , given the target spectrum occupancy, S_t , are obtained from $N = 1000$ independent test runs for a given α . The averaged histogram can be treated as an estimate of the conditional density of y_t , given S_t . For instance, fig. 5 shows these averaged histograms of different output classes for $\alpha = 0.7$. These indicate that the distributions of the predicted scores, conditioned on the training label, exhibit Gaussian-like features, with multiple peaks, bell-like symmetric curves, and having mean and variance parameters. Further details of the modelling methodology in fig. 4 are described in the following sections III-B and III-C.

B. PROPOSED MIXTURE MODEL

A mixture with appropriate component distributions is adopted to model the predicted scores, considering the

features of the averaged histograms in fig. 5. The proposed mixture model of the predicted scores, given the target spectrum occupancy is represented as $f_Y(y_t|S_t)$ and consists of Truncated Gaussian distributions (TGD), $\lambda_T(\cdot)$, as in (18), (19),

$$f_Y(y_t(i)|S_t = Idle) = \sum_{m=1}^M \theta_{im} \lambda_T(y_t(i); \nu_{im}, \delta_{im}, a, b) \quad (18)$$

$$f_Y(y_t(i)|S_t = Busy) = \sum_{k=1}^K \pi_{ik} \lambda_T(y_t(i); \mu_{ik}, \sigma_{ik}, a, b) \quad (19)$$

where, for a given test run:

- t is the time step
- $y_t(i)$ is the predicted score for the output classes
- i takes values 0, 1 corresponding to output classes
- K, M are the number of mixture components in each model
- π, θ stand for the proportions of the mixture components
- $\lambda_T(\cdot)$ stands for the PDF of a TGD component
- $[a, b]$ is the truncation interval
- μ, ν stand for the means of the parent Gaussian component distribution (before truncation)
- σ, δ are the standard deviation of the parent Gaussian component distribution (before truncation)

As mentioned in (18) and (19), $\lambda_T(\cdot)$ is the PDF of a TGD component, that is defined in general as,

$$\lambda_T(x; \mu, \sigma, a, b) = \begin{cases} \frac{1}{\sigma} \frac{\phi(x')}{\Phi(b') - \Phi(a')} & \text{if } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

where,

$$\begin{aligned} x' &= \frac{x - \mu}{\sigma} \\ a' &= \frac{a - \mu}{\sigma} \\ b' &= \frac{b - \mu}{\sigma} \end{aligned} \quad (21)$$

In (20), $\phi(\cdot)$ is the PDF of a standard normal variable, z , given by,

$$\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right) \quad (22)$$

and $\Phi(\cdot)$ is the CDF of z , given by,

$$\Phi(z) = \frac{1}{2} \left(1 + \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right) \right) \quad (23)$$

The CDF of a TGD is given by

$$\Lambda_T(x; \mu, \sigma, a, b) = \frac{\Phi(x') - \Phi(a')}{\Phi(b') - \Phi(a')} \quad (24)$$

In (20) and (24), μ and σ represent the mean and standard deviation parameters of the parent normal distribution of the variable, x ; $[a, b]$ is the interval of truncation such that,

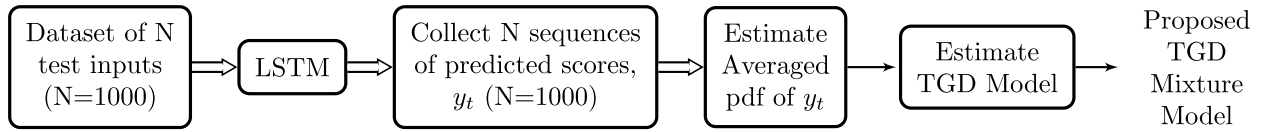


FIGURE 4. Modelling framework.

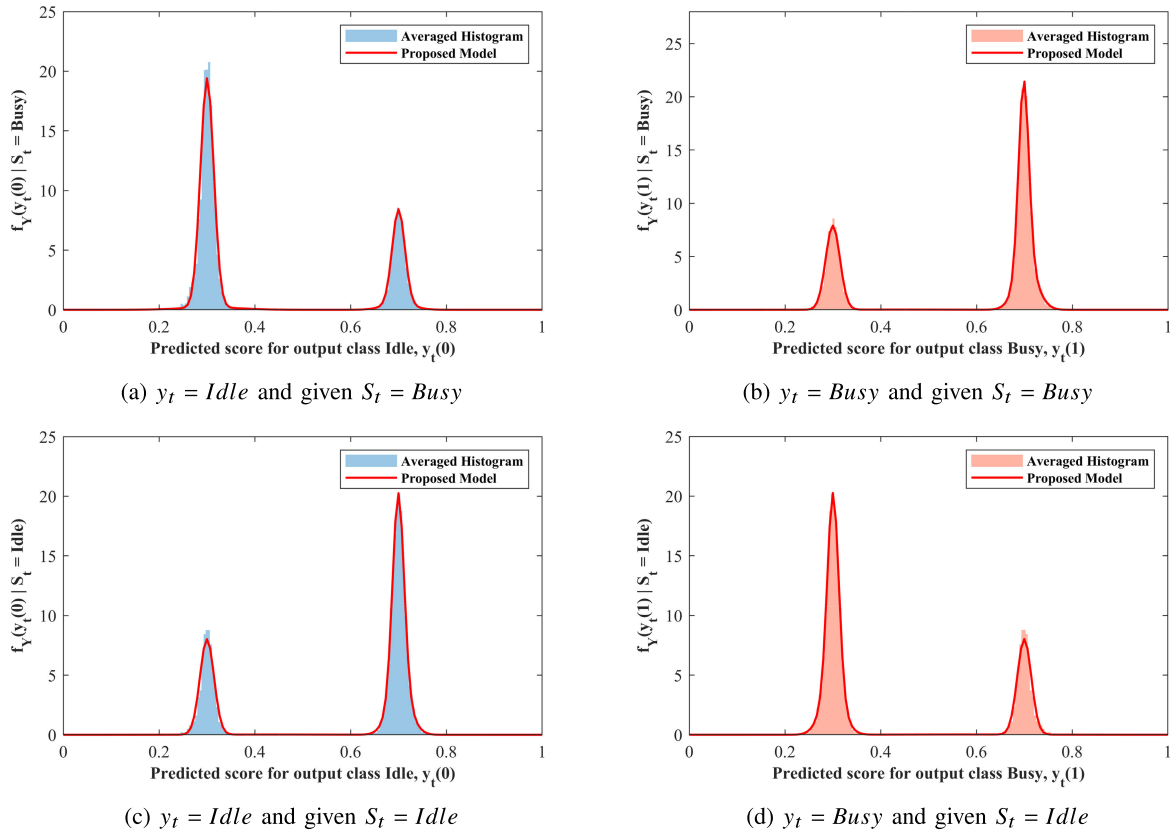


FIGURE 5. Comparison of proposed model and averaged histograms of predicted scores for $\alpha = 0.7$.

$a \leq x \leq b$. The mean and variance parameters of a TGD is given by,

$$\mu' = \mu + \frac{\phi(a') - \phi(b')}{\Phi(b') - \Phi(a')} \sigma \quad (25)$$

$$\sigma'^2 = \sigma^2 \left[1 + \frac{a'\phi(a') - b'\phi(b')}{\Phi(b') - \Phi(a')} - \left(\frac{\phi(a') - \phi(b')}{\Phi(b') - \Phi(a')} \right)^2 \right] \quad (26)$$

As we know from (10) that y_t takes values in the range $[0, 1]$, the proposed model for $f_Y(y_t|S_t)$ must also take non-zero values within the same range. Therefore, we have chosen the truncation interval, $[a, b]$, for the TGD components in (18) and (19) as $[0, 1]$. Moreover, the sum of the component proportions in each model of the form of (18) and (19) is equal to 1. This ensures that the corresponding PDF normalizes to 1.

C. MODEL PARAMETER ESTIMATION

As shown in fig. 4 and explained in section III-A, the averaged PDF of predicted scores are estimated based on multiple independent test runs. Further, we have adopted the following steps to model the predicted scores data using the proposed mixture of TGD components,

- 1) Estimate the mixture model parameters.
- 2) Truncate the PDF of mixture components to the required range of y_t , i.e., $[0, 1]$.
- 3) Recombine the TGD components to get the resulting mixture model.

These steps also account for the property of the predicted scores as in (10). To estimate the mixture model parameters in step 1, we have adopted the Expectation Maximization (EM) algorithm. The EM algorithm computes the maximum likelihood estimates of statistical model parameters iteratively [31]. This algorithm is usually computationally

less intensive, works well with missing data, and is an intuitive choice to fit mixture distributions. For each observation, the EM algorithm computes the posterior probability that the observation belongs to a particular Gaussian Mixture component. The algorithm then uses the computed posterior probabilities to estimate the model parameters by the Maximum Likelihood principle. The proposed mixture models obtained by steps (1) - (3) are superimposed over the corresponding averaged histograms in fig. 5. The corresponding CDFs of the predicted scores based on the proposed mixture model are shown in fig. 6. The model parameters of (18) and (19) estimated using EM algorithm are tabulated for different values of input model parameter, α as shown in tables 1 and 2. We have selected the number of mixture components, $K = M = 4$ in (18) and (19), as this performed well for our study.

D. MODEL VERIFICATION

The proposed mixture model as described in sections III-A - III-C is verified over a simulated data. The results for verification are shown in fig. 7 for a simulated dataset with $\alpha = 0.8$, wherein the theoretical mixture model is compared with the averaged histograms of the predicted scores for simulated test data. From fig. 7, the proposed mixture model fits the averaged histograms of the predicted scores well.

IV. PERFORMANCE OF LSTM BASED SPECTRUM PREDICTION

A. PROBABILITY OF ERROR OF LSTM BASED SPECTRUM PREDICTION

In this section, we present the discussion on the analytical expression for the probability of error of the LSTM based prediction model. To this end, we utilize the mixture model of the predicted scores, as described in section III-B.

We can define the probability of error, P_e , for the LSTM based system model in fig. 1 as,

$$P_e = Pr(y_t(0) \leq y_t(1)|S_t = Idle)Pr(S_t = Idle) + Pr(y_t(0) \geq y_t(1)|S_t = Busy)Pr(S_t = Busy) \quad (27)$$

where, $y_t(0)$ and $y_t(1)$ are the predicted scores of the LSTM model for output classes, *Idle* and *Busy*, respectively. As per the Markovian model, we have,

$$Pr(S_t = Idle) = \frac{\alpha}{\alpha + \beta} \quad (28)$$

$$Pr(S_t = Busy) = \frac{\beta}{\alpha + \beta} \quad (29)$$

In our study, we consider the case of $\alpha = \beta$ for simplicity. As per eqns. (28) and (29), we have,

$$Pr(S_t = Idle) = Pr(S_t = Busy) = \frac{1}{2} \quad (30)$$

In addition, as per (11), we have,

$$y_t(0) + y_t(1) = 1 \quad (31)$$

Applying (30) and (31) on (27), we get a concise form for P_e as

$$P_e = \frac{1}{2} \left(Pr(y_t(0) \leq \frac{1}{2}|S_t = Idle) + Pr(y_t(1) \leq \frac{1}{2}|S_t = Busy) \right) \quad (32)$$

As we model the conditional distributions of the predicted scores, $y_t(0)$ and $y_t(1)$, as mixtures of truncated Gaussian distributions, we utilize the CDF of such a distribution in (32). Applying (18), (19), and (24) on (32), we get the probability of error as,

$$P_e = \frac{1}{2} \sum_{m=1}^M \theta_{0m} \Lambda_T(\frac{1}{2}; \nu_{0m}, \delta_{0m}, 0, 1) + \frac{1}{2} \sum_{k=1}^K \pi_{1k} \Lambda_T(\frac{1}{2}; \mu_{1k}, \sigma_{1k}, 0, 1) \quad (33)$$

Based on our study, we observe that the error performance is also impacted by the amount of correlation in the data. To substantiate this, we present the autocovariance performance based on theoretical analysis and simulations in section IV-B.

B. AUTOCOVARANCE FUNCTION

In this section, we present the theoretical form of the autocovariance function of a two-state DTMC, as shown in fig. 2. The eigen values of the transition probability matrix, P , can be derived as,

$$\lambda_1 = 1 \quad (34)$$

$$\lambda_2 = (\alpha + \beta - 1) \quad (35)$$

The corresponding eigen vectors can be found as,

$$v_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (36)$$

$$v_2 = \begin{bmatrix} \alpha - 1 \\ \frac{1 - \beta}{1} \\ 1 \end{bmatrix} \quad (37)$$

Assuming that eigenvector decomposition can be performed, we can write P as follows,

$$P = C \Lambda C^{-1} \quad (38)$$

where, $\Lambda = diag(\lambda_1, \lambda_2)$ and C is the matrix with v_1 and v_2 as its columns. The k -step transition probability matrix can be calculated as (39), shown at the bottom of the next page.

The autocorrelation function can be expressed by definition as,

$$R_{xx}(d) = \sum_{x_n \in \{0,1\}} \sum_{x_{n+d} \in \{0,1\}} x_n x_{n+d} P(x_{n+d}|x_n) P(x_n) \quad (40)$$

For the initial probabilities of states given by,

$$\pi^{(0)} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (41)$$

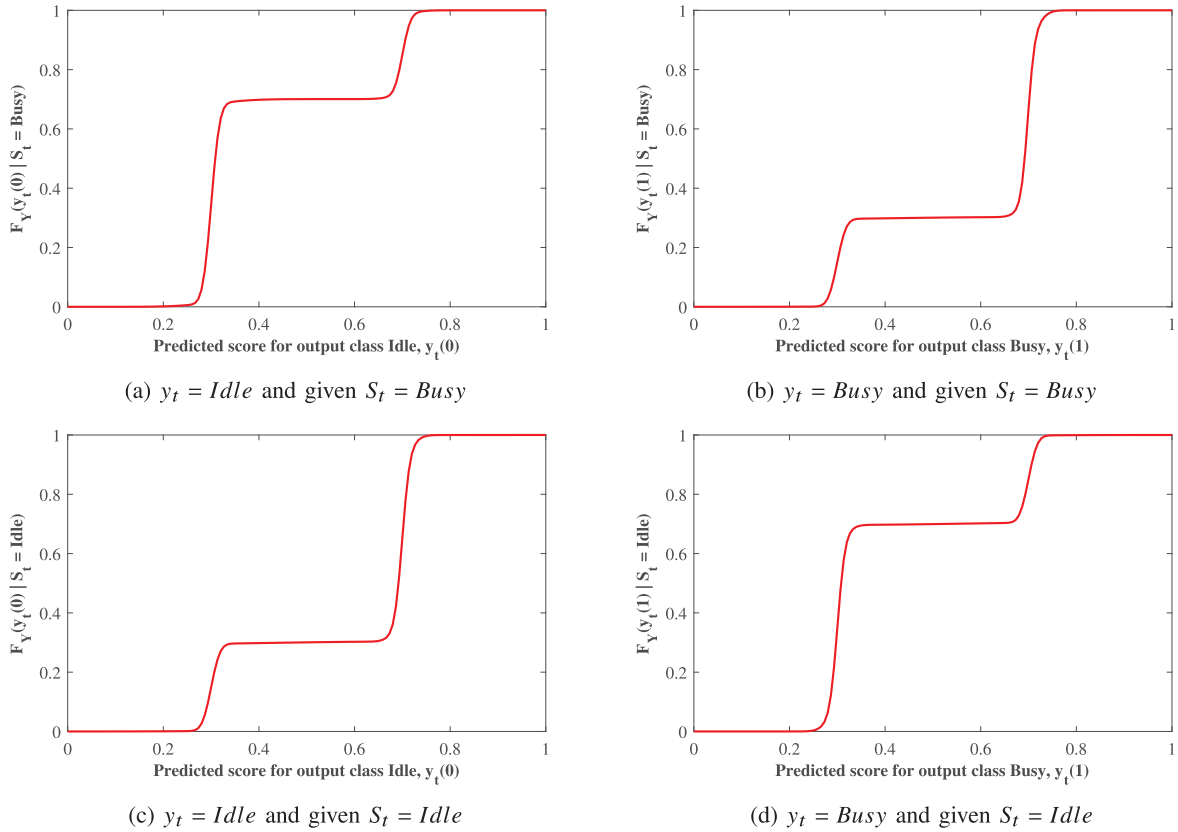


FIGURE 6. Comparison of CDFs of proposed models of predicted scores for $\alpha = 0.7$.

TABLE 1. Proportions and model parameters for the distribution of busy output class, given true label = 1.

α	π_{11}	μ_{11}	σ_{11}	π_{12}	μ_{12}	σ_{12}	π_{13}	μ_{13}	σ_{13}	π_{14}	μ_{14}	σ_{14}
0.5	0.296713	0.5029	1.0202e-04	0.155543	0.4986	3.4298e-04	0.386135	0.4968	9.4063e-05	0.161610	0.5030	1.2824e-04
0.55	0.164746	0.5516	2.3971e-04	0.160343	0.4507	2.7276e-04	0.384385	0.5492	1.1748e-04	0.290526	0.4506	1.0626e-04
0.6	0.393752	0.4003	1.6683e-04	0.198638	0.6038	2.5645e-04	0.010480	0.4763	0.0060	0.397130	0.5970	1.2720e-04
0.65	0.200501	0.6511	3.7957e-04	0.340938	0.3505	1.9101e-04	0.017910	0.4792	0.0141	0.440651	0.6489	1.2293e-04
0.7	0.007877	0.4317	0.0184	0.295086	0.2997	2.2236e-04	0.226594	0.7037	5.1653e-04	0.470443	0.6983	1.1120e-04
0.75	0.515135	0.7492	1.0263e-04	0.016245	0.5010	0.0307	0.226090	0.7520	6.5354e-04	0.242530	0.2494	2.1614e-04
0.8	0.605831	0.8003	1.1071e-04	0.188386	0.1994	1.8323e-04	0.026115	0.4808	0.0437	0.179669	0.7953	9.8683e-04
0.85	0.655387	0.8506	9.1656e-05	0.136572	0.1501	1.4230e-04	0.033718	0.4990	0.0576	0.174323	0.8318	0.0011
0.9	0.626430	0.9011	4.8449e-05	0.243394	0.8782	8.5855e-04	0.088240	0.1008	1.2340e-04	0.041936	0.5559	0.0674
0.95	0.673859	0.9504	1.9319e-05	0.039678	0.0511	4.7098e-05	0.040032	0.5530	0.0844	0.246431	0.9116	9.4396e-04

the distribution of states after k steps is given by,

$$\pi^{(k)} = \pi^{(0)} P^k \tag{42}$$

Applying (42) in (40) and making relevant substitutions, we get,

$$R_{xx}(d) = \frac{1}{2} \left(\frac{(\alpha - 1) - (\alpha + \beta - 1)^d (1 - \beta)}{(\alpha + \beta - 2)} \right) \tag{43}$$

For simplicity, we consider the case where $\alpha = \beta$. Then, (43) simplifies as,

$$R_{xx}(d) = \frac{1}{4} (1 + (2\alpha - 1)^d) \tag{44}$$

From (44), the autocovariance function (ACV) for the Markovian input can be derived as,

$$\sigma_{xx}(d) = R_{xx}(d) - (\mu_x)^2 = \frac{1}{4} (1 + (2\alpha - 1)^d) - \frac{1}{4} \tag{45}$$

$$P^k = \begin{bmatrix} \frac{(\alpha - 1)(\alpha + \beta - 1)^k - (1 - \beta)}{(\alpha + \beta - 2)} & \frac{(1 - \alpha)((\alpha + \beta - 1)^k - 1)}{(\alpha + \beta - 2)} \\ \frac{(\beta - 1)(1 - (\alpha + \beta - 1)^k)}{(\alpha + \beta - 2)} & \frac{(\alpha - 1) - (\alpha + \beta - 1)^k (1 - \beta)}{(\alpha + \beta - 2)} \end{bmatrix} \tag{39}$$

TABLE 2. Proportions and model parameters for the distribution of Idle output class, given true label = 0.

α	θ_{01}	γ_{01}	δ_{01}	θ_{02}	γ_{02}	δ_{02}	θ_{03}	γ_{03}	δ_{03}	θ_{04}	γ_{04}	δ_{04}
0.5	0.347906	0.4938	9.2266e-05	0.154622	0.5030	2.3691e-04	0.034695	0.5001	5.8467e-04	0.462777	0.5040	8.1807e-05
0.55	0.445641	0.4500	1.4989e-04	0.423134	0.5492	1.2184e-04	0.123656	0.5505	2.8133e-04	0.007568	0.4962	0.0021
0.6	0.356210	0.5988	1.1202e-04	0.395287	0.4008	1.7509e-04	0.010960	0.4967	0.0068	0.237543	0.6012	2.6275e-04
0.65	0.196003	0.6529	3.8106e-04	0.342877	0.3504	1.9265e-04	0.017365	0.5043	0.0126	0.443755	0.6483	1.2752e-04
0.7	0.489565	0.6996	1.3873e-04	0.294062	0.3001	2.1403e-04	0.009839	0.4283	0.0230	0.206534	0.7018	5.0474e-04
0.75	0.604645	0.7510	1.4413e-04	0.241234	0.2497	2.1322e-04	0.136469	0.7466	7.8702e-04	0.017652	0.4592	0.0314
0.8	0.147625	0.7902	0.0010	0.189938	0.1999	2.0636e-04	0.634534	0.8017	1.1444e-04	0.027903	0.5224	0.0442
0.85	0.664880	0.8506	8.5372e-05	0.139324	0.1511	1.7418e-04	0.037764	0.5500	0.0504	0.158032	0.8372	0.0013
0.9	0.616630	0.9001	4.6025e-05	0.090576	0.1021	1.6686e-04	0.043915	0.5870	0.0549	0.248878	0.8870	0.0011
0.95	0.647289	0.9499	2.2264e-05	0.037970	0.0500	2.6157e-05	0.042527	0.5322	0.0888	0.272213	0.9182	0.0011

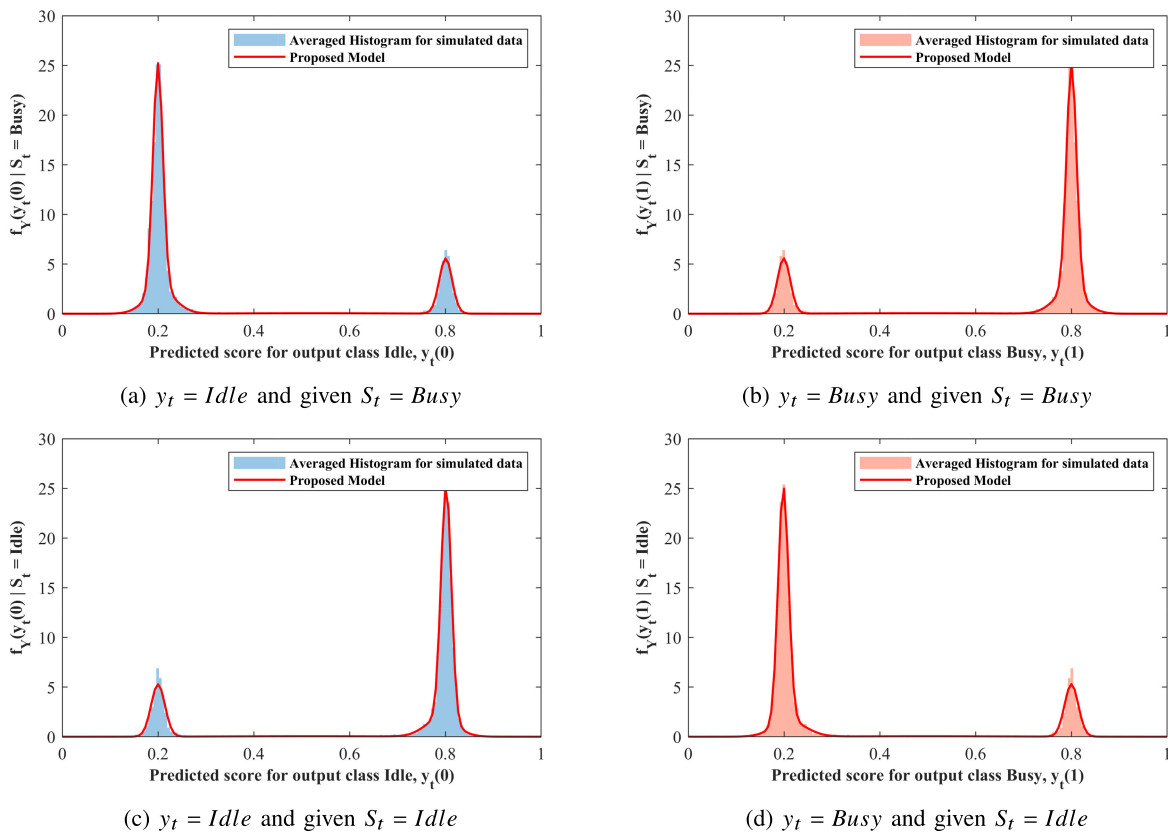


FIGURE 7. Comparison of theoretical model and averaged histograms of predicted scores for simulated data with $\alpha = 0.8$.

Figure 8 compares the ACV coefficients for lag values, $d = 1, 2$ using (44) and using simulations. The ACV coefficients for the predicted output is also shown in fig. 8.

C. ROOT MEAN SQUARE ERROR (RMSE)

In this section, we present the theoretical form of the RMSE of the LSTM model shown in fig. 1. We define the observed prediction error as,

$$P_e = \frac{N_e}{N} \tag{46}$$

where, N_e is the number of erroneous predictions, and N is the total number of predictions made in a single test run. We also define the RMSE for each test run as,

$$RMSE = \sqrt{\frac{\sum_n (\hat{y}_n - S_n)^2}{N}} \tag{47}$$

where, $n = 1, 2, \dots, N$ correspond to each prediction in a test run, \hat{y}_n represents the binary equivalent of the categorical predicted output, \hat{y}_t and S_n represents the actual binary occupancy value for the n^{th} prediction of the test run.

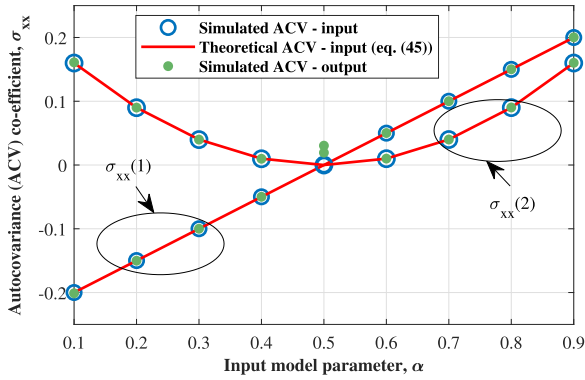


FIGURE 8. Comparison of Autocovariance coefficient, σ_{xx} for input and predicted output sequences.

Further, equation (47) can be expanded as,

$$RMSE = \sqrt{\frac{\sum_n (\hat{y}_n - S_n)^2}{N}} = \sqrt{E(\hat{y}_n^2) - 2E(\hat{y}_n S_n) + E(S_n^2)} \quad (48)$$

In (48), $E(\cdot)$ represents the expectation operator. Each term in (48) can be simplified using the definition of $E(\cdot)$ as,

$$E(\hat{y}_n^2) = 0^2 P(\hat{y}_n = 0) + 1^2 P(\hat{y}_n = 1) = P(\hat{y}_n = 1) \quad (49)$$

where,

$$P(\hat{y}_n = 1) = P(\hat{y}_n = 1 | S_n = 0) P(S_n = 0) + P(\hat{y}_n = 1 | S_n = 1) P(S_n = 1) \quad (50)$$

Similarly,

$$E(\hat{y}_n S_n) = P(\hat{y}_n = 1, S_n = 1) = P(\hat{y}_n = 1 | S_n = 1) * P(S_n = 1) \quad (51)$$

We also get,

$$E(S_n^2) = P(S_n = 1) \quad (52)$$

Substituting (49)-(52) in (48) and simplifying, we get,

$$RMSE = (P(\hat{y}_n = 1 | S_n = 0) * P(S_n = 0) + P(\hat{y}_n = 0 | S_n = 1) * P(S_n = 1))^{1/2} \quad (53)$$

Using (27), (53) can be simplified as,

$$RMSE = \sqrt{P_e} \quad (54)$$

D. MEAN ABSOLUTE ERROR (MAE)

In this section, we present the theoretical form of the Mean Absolute Error (MAE) of the LSTM model shown in fig. 1. We define the observed or simulated prediction error, P_e , as in (46). We also define the MAE for each test run as,

$$MAE = \frac{\sum_n |\hat{y}_n - S_n|}{N} \quad (55)$$

Here, \hat{y}_n and S_n are as defined in section IV-C. Further, the definition in (55) can be written as,

$$MAE = \frac{\sum_n |\hat{y}_n - S_n|}{N} = E(|\hat{y}_n - S_n|) \quad (56)$$

Further expanding (56) using the definition of $E(\cdot)$, we get,

$$MAE = \sum_n |\hat{y}_n - S_n| P(\hat{y}_n, S_n) \quad (57)$$

Splitting (57) into two sum terms corresponding to $\hat{y}_n = S_n$ and $\hat{y}_n \neq S_n$, we get,

$$MAE = \sum_{\hat{y}_n=S_n} |\hat{y}_n - S_n| P(\hat{y}_n, S_n) + \sum_{\hat{y}_n \neq S_n} |\hat{y}_n - S_n| P(\hat{y}_n, S_n) \quad (58)$$

As $\hat{y}_n = S_n \implies |\hat{y}_n - S_n| = 0$ and $\hat{y}_n \neq S_n \implies |\hat{y}_n - S_n| = 1$, we can simplify (58) as,

$$MAE = \sum_{\hat{y}_n \neq S_n} P(\hat{y}_n, S_n) = P(\hat{y}_n = 1, S_n = 0) + P(\hat{y}_n = 0, S_n = 1) \quad (59)$$

As the RHS of (59) is equivalent to the calculation of P_e , we have,

$$MAE = P_e \quad (60)$$

V. MODEL VERIFICATION AND NUMERICAL RESULTS

In this section, we verify the proposed analytical form of P_e discussed in section IV-A, for simulated and measured datasets. We also present the prediction performance results in terms of the mean observed prediction error, RMSE, and MAE, for different values of the parameter, α , over the simulated datasets.

A. VERIFICATION ON SIMULATED DATA

For the input model parameter α taking values as 0.5, 0.55, ..., 0.95, the mean observed prediction error is estimated over 1000 independent simulated test runs, as described in section III-A. The probability of error, P_e is also calculated as per (33), for the corresponding data. The theoretical and observed P_e values are as shown in fig. 9. As can be seen from fig. 9, the probability of error reduces with increasing values of α and the graph is symmetric with respect to $\alpha = 0.5$. Moreover, there is a linear relationship between P_e and α , in addition to the fact that the modelling based approach gives results that match with the simulated P_e averaged over multiple independent test runs.

B. VERIFICATION ON SPECTRUM MEASUREMENTS DATA

The probability of error of the LSTM based spectrum prediction model is estimated on the spectrum measurements data published by the RWTH Aachen University [32], [33]. The data used for verification is captured indoors in an

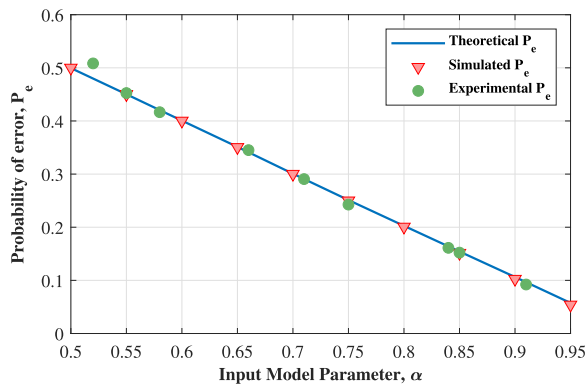


FIGURE 9. Comparison of probability of error.

office environment. The dataset comprises PSD measurements for 6000 sweeps and 8192 trace points, for the sub-band I, a centre frequency of 770 MHz and over a frequency range of 20 MHz to 1.52 GHz. The PSD data is converted to a binary spectrum occupancy data, Ω , as per (61),

$$\Omega_{t,i} = \begin{cases} 0 & \text{if } PSD_{rx,t,i} \text{ is } < \gamma \\ 1 & \text{if } PSD_{rx,t,i} \text{ is } \geq \gamma \end{cases} \quad (61)$$

where, $PSD_{rx,t,i}$ is the received PSD for time t , channel, i , and γ is the energy detection threshold, taken as -107 dBm/200kHz. The dataset, Ω , is assumed to consist of channel data following a two-state DTMC process. For this study, the channel data with approximately equal values of α and β are selected for testing the trained LSTM models. For different values of α , the observed prediction error is calculated as per (46) and compared with the probability of error calculated using (33). The comparison results are as shown in fig. 9. These results indicate that theoretical expression for P_e as in (33) applies well for LSTM based spectrum prediction over experimental data, for different cases of the input model parameter, α .

C. PREDICTION PERFORMANCE RESULTS

To analyze the prediction performance of the LSTM based model for a Markovian spectrum occupancy input in a statistical sense, 1000 independent test runs are carried out for a given value of α . The prediction error, P_e is calculated for each test run using (46) and averaged over all test runs. Figure 10 shows a comparison between the mean prediction errors of the LSTM based prediction model and that of an HMM-based spectrum prediction model, which is a Bayesian-based technique adopted in [27]. The prediction errors are plotted for different values of input autocovariance, $\sigma_{xx}(1)$, where, $\sigma_{xx}(1) = 0$ indicates the equally likely case of $\alpha = 0.5$. From fig. 10, we observe that the mean prediction error is symmetric and linearly decreasing on both sides of the peak error at $\sigma_{xx}(1) = 0$.

Figure 11 shows the comparison of the RMSE calculated theoretically and in simulations for different values of α . For a given α , the RMSE is calculated for each simulation test

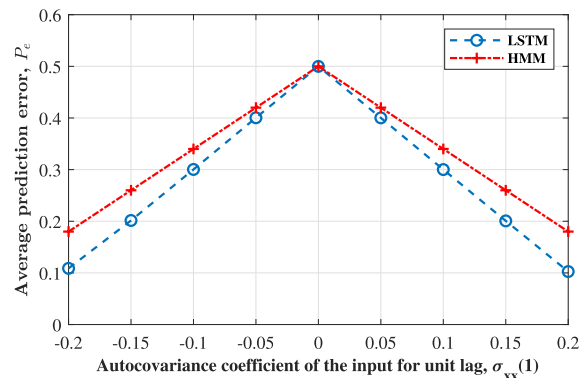


FIGURE 10. Comparison of probability of error of LSTM and HMM models [27].

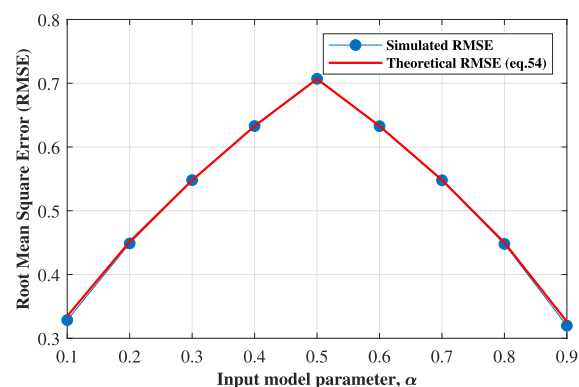


FIGURE 11. Comparison of simulated and theoretical RMSE for different values of input model parameter, α .

run using (47) and averaged over all test runs. Additionally, the theoretical value of RMSE is calculated using (54) and the model derived for P_e in (33). From fig. 11, we observe that the theoretically calculated RMSE closely matches the RMSE obtained from simulations. Figure 11 also shows a symmetric reducing trend on both sides of the maximum case corresponding to $\sigma_{xx}(1) = 0$. Therefore, the results indicate that the model exhibits the worst prediction performance for the truly random or equally likely case of the transition probability, α . The prediction performance improves steadily as the input occupancy data deviates from the equally likely case.

Figure 12 shows the comparison of the MAE calculated theoretically and in simulations for different values of α . For a given α , the MAE is calculated for each simulation test run using (55) and averaged over all test runs. Additionally, the theoretical value of MAE is calculated using (60) and the model derived for P_e in (33). From fig. 12, we observe that the theoretically calculated MAE closely matches the MAE calculated from simulations. Figure 12 also shows a symmetric reducing trend on both sides of the maximum case of $\alpha = 0.5$, corresponding to $\sigma_{xx}(1) = 0$. Therefore, the results indicate that the model exhibits the worst prediction performance for the truly random case of α and improves

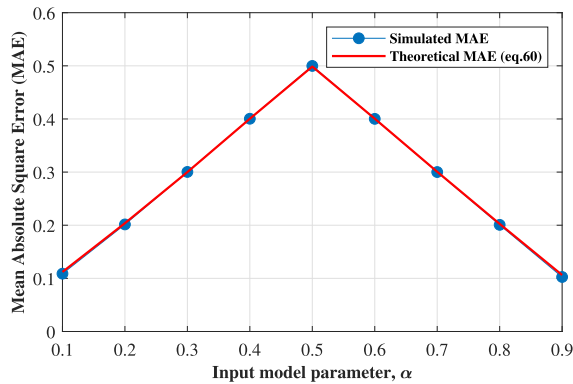


FIGURE 12. Comparison of simulated and theoretical MAE for different values of input model parameter, α .

steadily in case of higher correlation in the input occupancy data.

VI. CONCLUSION

In this article, we analyzed and modelled the prediction performance of an LSTM based system model for spectrum prediction. We assumed the spectrum occupancy data to follow a two-state discrete-time Markov model. We then tested a trained LSTM system model with simulated and measured spectrum datasets and analyzed the model's prediction scores for each output class. Based on our results, we verified that the predicted scores of the model could be represented using mixtures of truncated Gaussian distributions. Further, we arrived at the probability of error of the prediction model based on the proposed mixture model and compared the results with the observed prediction error over simulated and measured data. We also presented the comparison of simulated and theoretically calculated RMSE and MAE for the LSTM system model. In the future, we aim to characterize the prediction performance of multi-step ahead spectrum prediction using LSTM models.

REFERENCES

- [1] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management toward 5G radio access networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1656–1686, 3rd Quart., 2016.
- [2] X. Xing, T. Jing, W. Cheng, Y. Huo, and X. Cheng, "Spectrum prediction in cognitive radio networks," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 90–96, Apr. 2013.
- [3] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "A survey on spectrum management in cognitive radio networks," *IEEE Commun. Mag.*, vol. 46, no. 4, pp. 40–48, Apr. 2008.
- [4] H. Eltom, S. Kandeepan, R. J. Evans, Y. C. Liang, and B. Ristic, "Statistical spectrum occupancy prediction for dynamic spectrum access: A classification," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, p. 29, Feb. 2018.
- [5] B. S. Shawel, D. H. Woldegebreal, and S. Pollin, "Convolutional LSTM-based long-term spectrum prediction for dynamic spectrum access," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2019, pp. 1–5.
- [6] 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 Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 150–182, 1st Quart., 2018.
- [7] I. Alkhouri, M. Joneidi, F. Hejazi, and N. Rahnavard, "Large-scale spectrum occupancy learning via tensor decomposition and LSTM networks," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Apr. 2020, pp. 677–682.
- [8] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Machine learning for wireless communications in the Internet of Things: A comprehensive survey," *Ad Hoc Netw.*, vol. 93, Oct. 2019, Art. no. 101913. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1570870519300812>
- [9] H. D. Trinh, L. Giupponi, and P. Dini, "Mobile traffic prediction from raw data using LSTM networks," in *Proc. IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1827–1832.
- [10] T. Senevirathna, B. Thennakoon, T. Sankalpa, C. Seneviratne, S. Ali, and N. Rajatheva, "Event-driven source prediction in machine-type communications using LSTM networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
- [11] L. Yu, J. Chen, G. Ding, Y. Tu, J. Yang, and J. Sun, "Spectrum prediction based on Taguchi method in deep learning with long short-term memory," *IEEE Access*, vol. 6, pp. 45923–45933, 2018.
- [12] M. A. Aygul, M. Nazzal, A. R. Ekti, A. Gorcin, D. B. da Costa, H. F. Ates, and H. Arslan, "Spectrum occupancy prediction exploiting time and frequency correlations through 2D-LSTM," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.
- [13] J. Sun, J. Wang, J. Chen, G. Ding, and F. Lin, "Clustering analysis for internet of spectrum devices: Real-world data analytics and applications," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4485–4496, May 2020.
- [14] L. Yu, J. Chen, and G. Ding, "Spectrum prediction via long short term memory," in *Proc. 3rd IEEE Int. Conf. Comput. Commun. (ICCC)*, Dec. 2017, pp. 643–647.
- [15] L. Yu, Q. Wang, Y. Guo, and P. Li, "Spectrum availability prediction in cognitive aerospace communications: A deep learning perspective," in *Proc. Cognit. Commun. Aerosp. Appl. Workshop (CCAA)*, Jun. 2017, pp. 1–4.
- [16] J. Joo, M. C. Park, D. S. Han, and V. Pejovic, "Deep learning-based channel prediction in realistic vehicular communications," *IEEE Access*, vol. 7, pp. 27846–27858, 2019.
- [17] P. Zuo, X. Wang, W. Linghu, R. Sun, T. Peng, and W. Wang, "Prediction-based spectrum access optimization in cognitive radio networks," in *Proc. IEEE 29th Annu. Int. Symp. Personal, Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 1–7.
- [18] P. Zuo, T. Peng, X. Wang, K. You, H. Jing, W. Guo, and W. Wang, "Spectrum prediction for frequency bands with high burstiness: Analysis and method," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–7.
- [19] X. Li, Z. Liu, G. Chen, Y. Xu, and T. Song, "Deep learning for spectrum prediction from spatial-temporal-spectral data," *IEEE Commun. Lett.*, vol. 25, no. 4, pp. 1216–1220, Apr. 2021.
- [20] B. S. Shawel, D. H. Woldegebreal, and S. Pollin, "Deep-learning based cooperative spectrum prediction for cognitive networks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 133–137.
- [21] O. Omotere, J. Fuller, L. Qian, and Z. Han, "Spectrum occupancy prediction in coexisting wireless systems using deep learning," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–7.
- [22] Siddhartha, Y. H. Lee, D. J. M. Moss, J. Faraone, P. Blackmore, D. Salmond, D. Boland, and P. H. W. Leong, "Long short-term memory for radio frequency spectral prediction and its real-time FPGA implementation," in *Proc. IEEE Mil. Commun. Conf. (MILCOM)*, Oct. 2018, pp. 1–9.
- [23] B. Soni, D. K. Patel, and M. Lopez-Benitez, "Long short-term memory based spectrum sensing scheme for cognitive radio using primary activity statistics," *IEEE Access*, vol. 8, pp. 97437–97451, 2020.
- [24] N. Radhakrishnan and S. Kandeepan, "An improved initialization method for fast learning in long short-term memory-based Markovian spectrum prediction," *IEEE Trans. Cognit. Commun. Netw.*, vol. 7, no. 3, pp. 729–738, Sep. 2021.
- [25] X. Li, Z. Liu, Y. Xu, X. Wang, and T. Song, "Recovering missing values from corrupted historical spectrum observations for dependable spectrum prediction," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Aug. 2020, pp. 941–946.
- [26] Y. Saleem and M. H. Rehmani, "Primary radio user activity models for cognitive radio networks: A survey," *J. Netw. Comput. Appl.*, vol. 43, pp. 1–16, Aug. 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804514000848>
- [27] H. Eltom, S. Kandeepan, B. Moran, and R. J. Evans, "Spectrum occupancy prediction using a hidden Markov model," in *Proc. IEEE Int. Conf. Signal Process. Commun. Syst. (ICSPCS)*, Dec. 2015, pp. 1–8.
- [28] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[29] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," 2013, *arXiv:1312.6120*.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13rd Int. Conf. Artif. Intell. Statist. (AISTATS)*, 2010, pp. 249–256.

[31] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., B Methodol.*, vol. 39, no. 1, pp. 1–38, 1977.

[32] M. Wellens and P. Mähönen, "Lessons learned from an extensive spectrum occupancy measurement campaign and a stochastic duty cycle model," *Mobile Netw. Appl.*, vol. 15, no. 3, pp. 461–474, Jun. 2010.

[33] M. Wellens, J. Riihijärvi, and P. Mähönen, "Empirical time and frequency domain models of spectrum use," *Phys. Commun.*, vol. 2, nos. 1–2, pp. 10–32, Mar. 2009.



NIRANJANA RADHAKRISHNAN received the B.Tech. degree (Hons.) in electronics and communication engineering from the University of Calicut, in 2014, and the M.Tech. degree in electronic systems engineering from the Indian Institute of Science, Bangalore, in 2017. She is currently pursuing the Ph.D. degree in electrical and electronics engineering with the School of Engineering, RMIT University, Melbourne, Australia. Her research interests include applying

machine learning to optimize communication systems, recurrent neural networks, and cognitive radios. She was among the top ten rank holders from the University of Calicut and secured the Best Student Project Award by Tata Consultancy Services Ltd., in recognition of academic excellence, in 2014.



SITHAMPARANATHAN KANDEEPAN (Senior Member, IEEE) is currently the Discipline Leader of telecommunications and photonics with the School of Engineering, RMIT University. His research interests include communication engineering, signal processing and networks applied in wireless, satellite, and mobile systems. He has extensive experience in working with the industry for bringing research to practice and conducting fundamental research. He currently works on several

research projects, such as defence science technology on intelligent spectrum management communications, satellite-aerial terrestrial communication systems, cognitive (satellite) radios, and the IoT communications. He has authored a scholarly book *Cognitive Radio Techniques: Spectrum Sensing, Interference Mitigation and Localization* (New York: Artech House, 2012). He has published over 135 peer reviewed scientific journals and conference papers. He had received the RMIT Research Excellence Award, in 2019, for his research work with the industry. He currently engaged with the ITU as an Academic Member, the Chair of IEEE Comsoc VIC chapter, and the Vice Chair of the IEEE Technical Committee on Cognitive Networks.



XINGHUO YU (Fellow, IEEE) received the B.Eng. and M.Eng. degrees in electrical and electronic engineering from the University of Science and Technology of China, Hefei, China, in 1982 and 1984, respectively, and the Ph.D. degree in control science and engineering from Southeast University, Nanjing, China, in 1988. He is currently an Associate Deputy Vice-Chancellor and a Distinguished Professor with the Royal Melbourne Institute of Technology (RMIT University), Melbourne, Australia. He is also the Junior Past President of the IEEE Industrial Electronics Society, from 2020 to 2021. His research interests include control systems, complex and intelligent systems, and smart energy systems. He received a number of awards and honors for his contributions, including the 2013 Dr.-Ing. Eugene Mittelmann Achievement Award of IEEE Industrial Electronics Society, the 2018 MA Sargent Medal from Engineers Australia, and the 2018 Australasian AI Distinguished Research Contribution Award from Australian Computer Society. He has served as an Associate Editor for *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS-I: REGULAR PAPERS*, *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS*, and *IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS*.



GIANMARCO BALDINI (Senior Member, IEEE) received the Laurea degree in electronic engineering and the Ph.D. degree from the University of Rome, Rome, Italy, in 1993 and 2019, respectively. He has been a Scientific Project Manager with the Joint Research Centre (JRC), European Commission, Ispra, Italy, since 2007.

...