International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 1
***Article Received:** 25 August 2023 **Revised:** 12 November 2023 **Accepted:** 30 December 2023*

Deep Stacked CNN-LSTM (DS-CNN-LSTM) based Spectrum Sensing in Cognitive Radio

Haribhau Shinde
Department of Electronics and Communication Engineering
Oriental University, Indore (M. P.), India
Corresponding Author Email : haribhau88@gmail.com

Dr. Sandeep Garg
Department of Electronics and Communication Engineering
Oriental University, Indore (M. P.), India
sandeepgargvlsi@gmail.com

Abstract—The multidimensionality of spectrum sensing, the intrinsic complexity of its dependence, and the unpredictability associated with spectrum data all contribute to the difficulty of the task. The network of cognitive radio (CR) is comprised of both primary and secondary users inside its network. The SUs that are part of the CR network are able to identify the spectrum band and access white space in an opportunistic manner. Enhancing spectrum efficiency may be accomplished by using white spaces. This study presents a Deep Stacked CNN-LSTM (DS-CNN-LSTM)-based spectrum sensing strategy that learns implicit features from spectrum data, such as temporal correlation. This approach is based on the research that we have conducted. The effectiveness of the recommended method is shown by a sufficient number of simulations, and the results of the simulations demonstrate that it outperforms the current state of the art in terms of detection probability and classification accuracy. A comparison is made between the most cutting-edge spectrum sensing approaches and the DS-CNN-LSTM method that has been recommended. The results of the experiments indicate that the proposed methods improve detection performance and classification accuracy even when the signal-to-noise ratio is low. As we can see, the improvement that was achieved comes at the price of a longer amount of time spent on training and a little increase in the amount of time spent on execution.

Keywords- Cognitive radio, spectrum sensing, long short-term memory, convolutional Neural networks, deep learning. primary users (PUs) , secondary users (SUs)

I. Introduction

The spectrum resources have become very valuable as a result of the fast growth of wireless communication technology and the deployment of the 5G paradigm. There is a major underutilization of spectrum resources, as shown by the spectrum occupancy campaign in [2], which states that the total use of spectrum bands ranges from 7% to 34%. One possible solution to the trade-off between the availability of spectrum and rapid expansion is the cognitive radio (CR) technology [3], which has emerged as a viable remedy. By reusing briefly unoccupied frequency bands, often known as spectrum gaps or white spaces, in an opportunistic way, it seeks to protect licensed users from interference while simultaneously preventing interference from occurring [4]. Primary users (PU) are the users who have been granted permission to access the CR network, whereas secondary users (SU) are the users who do not have permission to access the network. The usage of spectrum may be significantly increased by spectrum sensing. Actually, owing to the low signal strength and signal-to-noise ratio, it is difficult to determine whether or not the spectrum is occupied. This is because of the fact that the spectrum is occupied. In order to prevent interference with main users, CR makes an effort to allow secondary users to access spectrum bands that are not being used to their full potential. The use of a spectrum detecting device with a high level of effectiveness is necessary in order to remove interference from secondary users [5-8]. Detecting the spectrum and making use of the frequency band may be accomplished via the use of a variety of spectrum sensing methods [9] in [11].

In spite of the fact that these spectrum sensing technologies are utilized for the purpose of distributing spectrum bands, the capture of vital user actions in the presence of ambient noise continues to give rise to difficulties [12–15]. The conventional spectrum sensing systems normally make use of well-designed test statistics that are derived from the sensing signals that have been received. These statistics are then compared to a threshold that has been specified in order to determine the availability of the spectrum. Depending on the amount of prior information that is required about the PU's signal (such as the modulation type and grade, pulse shape, and frame format) and noise (such as the channel model and strength), spectrum sensing systems may be divided into three distinct categories: nonblind, semi blind, and entirely blind. The methods that constitute nonblind spectrum sensing are based on precise statistical models of the signal and noise produced by the PU. When using traditional spectrum sensing, the secondary user (SU) is only able to identify a single characteristic of the signal that is being sent by the main user (PU). Single feature detection, on the other hand, does not do a comprehensive analysis of PU data. In order to develop an efficient spectrum sensing model, it is essential to use a combination of Convolutional Neural Networks (CNN) and Long ShortTerm Memory (LSTM), as shown by the research that has been conducted up to this point [28]. On the other hand, these models are intended for use in contexts that are devoid of noise and do not include real-time spectral data. In this article, Hybrid Optimized LSTM Enabled Networks are discussed. These networks make use of Convolutional Neural Networks (CNN) and Optimized Long Short-Term Memory (LSTM) to perform better in noisy environments. The most important addition that this study makes is a novel hybrid combination of CNN and LSTM. In this novel hybrid combination, the hyperparameters of LSTM are optimized to achieve high sensing accuracy.

II. LITERATURE SURVEY

Despite the fact that the bulk of these models fail to take into account temporal dependencies in the signal, machine learning techniques are often used in the research literature for the purpose of spectrum sensing in cognitive radio communications. In order to take use of the temporal correlation that exists between spectrum data points, this article makes use of a Long Short-Term Memory (LSTM) network. A hierarchical cooperative long short-term memory (LSTM) network-based cooperative spectrum sensing (CSS) approach was developed by D. Janu and colleagues [17]. This technique makes use of both a convolutional neural network (CNN) and an LSTM network. The convolutional neural network (CNN) is able to extract spatial properties from the input covariance matrices (CMs) provided by the sensory data of each secondary user. A. Rojas and colleagues [18] offered eight different strategies for narrowband spectrum sensing, including three fuzzy logic algorithms and four deep learning-based approaches. There are a variety of implications and aggregation processes that are used by fuzzy logic systems. Triangular and Gaussian membership functions are also utilized. Convolutional neural networks (CNN), long short-term memory (LSTM), and fully connected (FC) layers are the three primary designs that are used in the construction of deep learning systems. The stacked autoencoder (SAE) and the bi-directional long short-term memory (Bi-LSTM) based spectrum prediction approach (SAEL-SP) are presented by G. Pan and colleagues [19]. In order to extract hidden properties (semantic coding) from spectrum data in an unsupervised manner, an SAE was created expressly for this purpose from the beginning. After then, the output of the SAE is sent into a predictor known as a Bi-LSTM, which is able to learn long-term predictions by discovering previously unknown traits. A novel cluster-based cooperative sensing-after-prediction strategy is presented by D. Nie et al. [20]. This approach is characterized by the collaboration of a learning cluster and a sensing cluster in order to achieve successful cooperative prediction and sensing. Chae and colleagues [21] introduced a novel spectrum sensing approach that is based on deep learning and makes use of a receiver with several antennas. The creation of a correlation matrix that incorporates not only auto correlation functions for each antenna but also cross-correlation functions across antennas is the essential notion that we are working with. Our Deep Spectrum Sensing with Multiple Antennas (DS2MA) model is able to swiftly train to recognize the presence of a principal user (PU) by using a rich informative matrix and a fundamental convolutional neural network (CNN) structure. This allows the model to quickly adapt to changing conditions. ConvLSTM-based spectrum sensing is a methodology that was proposed by Q. Wang and colleagues [22]. This method makes use of the ConvLSTM network to simultaneously extract the temporal and spatial components of the observed IQ signals. Subsequently, low-SNR spectrum sensing is performed based on the features that were produced from the ConvLSTM network.

The Time-Frequency-Fused Adjustable Deep Convolutional Neural Network (TFF_ADCNN) was proposed by X. Li and colleagues [23]. This network was trained to give a pre-trained base model with a single distribution. In the subsequent step, the author used the base model for transfer learning in order to accomplish the sensing job in the actual environment. This resulted in a newly trained sensing model that was rather quick to be created. A new compact optical fiber concentration sensing system that is based on machine learning was proposed by J. Xue and colleagues. A prediction-driven channel-switching scheduling system for multi-channel customer relationship networks (MC-CRNs) is presented by Chauhan et al. [25]. This system is based on machine learning. Because it strikes a balance between spectral efficiency and channel switching cost, the technique that has been presented makes the most of the utility of SUs. Initially, a network-based channel prediction approach that is based on Long-Short Term Memory (LSTM) is introduced. On the basis of their previous spectrum sensing experiences, individual SUs makes local predictions on the occupancy of PUs in future time slots. A novel method for spectrum sensing is presented by M. Liu and colleagues [26], which is based on the combination of multimodal fusion and convolutional neural networks (CNN). To begin, the signal that has been received from a number of antennas is preprocessed by using the generalized covariance matrix and the generalized Wigner-Ville distribution. This is done in order to define two distinct modes of the received signal, which are then then used as input to CNN. Afterwards, a CNN model that incorporates multimodal fusion is developed.

III. METHODOLOGY

Depending on whether or not the primary user signal is present, the sequence characteristics of the received signal might vary significantly. For the purpose of extracting the temporal attributes of each major user's signal sequence, LSTM is used. The fully connected layer is utilized for the purpose of fusing the features in the fusion center, and SoftMax is utilized for the purpose of classifying the merged features. Because of this, the current research in spectrum sensing has focused a significant amount of attention on deep learning, which does not make any assumptions about the models being used. It is important to note that in the field of deep learning, convolutional neural networks (CNNs) and long-short term memory (LSTM) networks have excellent capabilities for extracting spatial and temporal features of input, respectively. In this letter, we present a CNN-LSTM detector that first extracts energy-correlation features from the covariance matrices created by the sensing data, and then inputs a sequence of energy-correlation features corresponding to different sensing periods into the LSTM in order to learn the PU activity pattern. This algorithm is designed to facilitate the learning of the PU activity pattern.

3.1 System Model:

A multi-user scenario is regarded to exist in the cognitive radio setting. Transmission of primary user signals is carried out by a main user (PU) transmitter. It is captured and sampled that the primary signal consumers are monitored. In order to train and assess the suggested design, the sampled signals are employed. This enables the architecture to identify samples that are not known to exist inside the network infrastructure.
Consider,

$$X(k) = \{X\ x_1(k), x_2(k), x_3(k), \ldots, x_m(k)\} \quad [1]$$

where m represents the number of user and k denotes the received signals from m users. X (k) denotes the discrete time sample present at mth users. The paper uses the binary hypothesis testing process for spectrum sensing as mentioned,

$$H_1: x(k) = RN(k) + Y(k)$$
$$H_0: x(k) = Y(k)$$

Channel fading and route loss are two factors that might have an effect on the signal vector $RN(k)$. The separate noise vector that contains the aero mean is denoted by $Y(k)$. As a result, hypothesis H_1 implies that there is a significant user, but hypothesis H_0 says that there is no such user. In order to train and test the proposed architecture, these signal characteristics are separated into real and imaginary components. These components help to train and test the design.

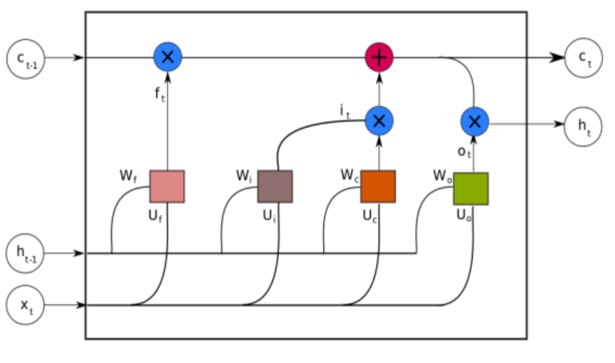

Fig 1: Memory Networks for Long-Term and Short-Term Storage

In order to accommodate the concept of memory, it is possible to alleviate the disadvantage that is associated with normal RNNs by including three gates into each network cell. In an effort to make things easier, this is done. A memory is first established and then updated inside the cell whenever it receives data. This happens each and every time.

LSTMs with four gate: forget (f), input (i), memory (c), and output gate (o).

If we have an old memory, C_{t-1}, we can calculate the new cell memory, C_t, as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad [2]$$

Forget Gate: chooses which information will be removed from the working memory.

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad [3]$$

Memory Gate: generates a new set of potential recollections.

$$\tilde{C}_t = \tanh\,(W_c x_t + U_c h_{t-1} + b_c) \quad [4]$$

Input Gate: This gate regulates the quantity of data that is sent from the candidate memory to the updated memory.

$$i_t = \sigma\,(W_i x_t + U_i h_{t-1} + b_i) \quad [5]$$

Output Gate: restricts the amount of data that may be retrieved from the memory of the unit.

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

The LSTM networks that make up ELMo are linked to one another in a back-to-back configuration. Word vectors that are ELMo are generated at the top end of a two-layer bidirectional word embedding, which is referred to as BiLSTM. Two levels are included in this template, with each layer being placed on top of the other. There are two passes ahead and two passes back. Its forward pass includes information on that word in addition to further phrases that have meanings that are comparable to that word right up to that point. The information from the backward pass includes additional information about the word as well as the context that comes after it. This is the final ELMo description, which is obtained by adding together all of the fundamental word predictions and the word indexes that are most likely to be accompanied by them. LSTM algorithms are enhanced by BiLSTM algorithms, which analyses data in two independent LSTM layers in both forward and backward orientations when applied to the data. On the other hand, the backward layer analyses the same data in reverse order, and the forward layer processes the input in the same manner that a normal LSTM does. In a neural network, the first layer is known as the input layer, and it also serves as the point of entry into the network.

During the training process, the Dropout Layer is responsible for introducing disorder into the network by regularly disrupting the number of connections that exist between neurons as they move from one layer to the next. The result is a reduction in overfitting, which makes it possible for models to generalize more successfully. Generally speaking, this results in an improvement in the accuracy of the model throughout the assessment.

- LSTM Layer: An implementation of a single LSTM layer that meets all of the forward and backward requirements for the creation process is carried out.
- A layer that enables RNN layers to construct models that look in both directions. In place of generating two distinct RNN layers for forward and reverse direction and adding the outputs, the bidirectional template element does all of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 12 Issue: 1
Article Received: 25 August 2023 Revised: 12 November 2023 Accepted: 30 December 2023

these tasks inside a single layer. The Dense Layer is comprised of a single completely linked vanilla artificial neural layer being used. In the Embedding Layer, positive integers are converted into floating point vectors. This layer is responsible for this transformation.

- A one-dimensional version of the convolutional neural network layer is referred to as the Conv1D Layer.
- A layer that performs maximum pooling in a single dimension is referred to as the MaxPooling1D Layer.

A mathematical representation of it may look something like this, for example:

$$x_j^l = f\left(\sum_{i \in M_j} x_i^{l-1} k_{ij}^l + b_j^l\right) \quad [7]$$

Where x_j^l is the output of the previous layer, x_i^{l-1} is the output of the current layer, k_{ij}^l is the kernel for the present layer, b_j^l is the bias for the current layer, and M_j represents a selection of input maps. Convoluting a text with several filters in different combinations may help with tasks like recognition, identification.

The second layer, known as the pooling layer, is utilized to reduce the number of parameters if the data are too large to process with the first layer alone. Spatial pooling, also known as sub - sampling or down sampling, reduces the number of dimensions in each map while preserving the crucial details. Pooling is a sampling-based approach of discretization. Its goal is to minimize the number of dimensions in an input sequence (such as an image or the output matrix of a hidden layer). The features contained in the sub-regions are thought to have been binned. Common types of pooling include maximum pooling and minimum pooling. As its primary function is down sampling, this layer is often described as the subsampling layer. This procedure may be described in a number of ways, including the one shown here:

$$x_j^l = f\left(\beta_j^l \, down\left(x_j^{l-1}\right) + b_j^l\right) \quad [8]$$

Each dot here denotes a different kind of subsampling strategy. This function typically picks either the average or the greatest value from all nxn blocks of the mappings from the layer below it. Maximum pooling, like minimum pooling, seeks to choose the greatest possible value from a given collection. Choosing the greatest possible value is the objective of max pooling. To build linear stacking for metaheuristic algorithms without limiting the program's generalization ability, we will use the interaction of two subcomponents as an instance. The approach may be easily adapted for use with many other kinds of subsystems.

Let's call the first result from the deep learning module as $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_i, \cdots, \boldsymbol{y}_N]$ in terms of the posterior probability at the level of frames of $\boldsymbol{C}$ classes and with a total of $\boldsymbol{N}$ frames in the data (test or training); that is, $\boldsymbol{Y} \in R^{C\times N}$. the same result is obtained for the second subsystem. $= [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_i, \cdots, \boldsymbol{z}_N] \in R^{C\times N}$.(Since the information we're sifting through is a representation of medical transcripts, our sample size of N = 10,000 and a confidence level of $C = 1000$. To generate the combined system's output at each frame, we use linear ensemble learning. $i = 1,2,3,4, \dots N$ to be.

$$\boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{W}\boldsymbol{z}_i \in R^C \quad [9]$$

a sequence of which, with $i = 1,2,3,4, \dots N$, $\mathrm{i} = 1, 2, \cdots, \mathrm{N}$, is passed to a different HMM to generate the test phoneme or word sequences. The two matrices, $\boldsymbol{V} \in R^{C\times C}$ and $\boldsymbol{W} \in R^{C\times C}$, are the ad hoc variables that may be adjusted as needed during training and which we'll go through below.

Parameter estimation:

To master $\boldsymbol{V}$ and $\boldsymbol{W}$, we turn to the supervised learning environment. For this setup, the pre-labeled category targets just at segment level of the data sets serve as the supervisory signal:

$$\boldsymbol{T} = [\boldsymbol{t}_1, \cdots, \boldsymbol{t}_i, \cdots, \boldsymbol{t}_N] \in R^{C\times N} \quad [10]$$

Possibilities based on information gained in retrospect $\boldsymbol{Y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_i, \cdots, \boldsymbol{y}_N]$ and $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_i, \cdots, \boldsymbol{z}_N]$,comprise the training data input. N is the total number of images used in the training process.

For this purpose, we shall use TSE as our loss function. The training objective function of is derived using $\boldsymbol{L}_2$ regularisation.

$$E = \frac{1}{2}\sum_i \parallel \boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{W}\boldsymbol{z}_i - \boldsymbol{t}_i \parallel^2 + \lambda_1 \parallel \boldsymbol{V} \parallel^2 + \lambda_2 \parallel \boldsymbol{W} \parallel^2, \quad [11]$$

where λ_1 and λ_2 are two experimental hyper-parameters, Lagrange multipliers, that we tune using both training and validation data. Making a few tweaks to (2) improves it.

$$\frac{\partial E}{\partial \boldsymbol{V}} = \boldsymbol{0} \text{ and } \frac{\partial E}{\partial \boldsymbol{W}} = \boldsymbol{0},$$

we acquire

$$\sum_i (\boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{W}\boldsymbol{z}_i - \boldsymbol{t}_i)\boldsymbol{y}_i^T + \lambda_1 \boldsymbol{V} = \boldsymbol{0} \quad [12]$$

$$\sum_i (\boldsymbol{V}\boldsymbol{y}_i + \boldsymbol{W}\boldsymbol{z}_i - \boldsymbol{t}_i)\boldsymbol{z}_i^T + \lambda_2 \boldsymbol{W} = \boldsymbol{0} \quad [13]$$

The equations in this set may be reduced to

$$\boldsymbol{V}(\boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}} + \lambda_1 \boldsymbol{I}) + \boldsymbol{W}(\boldsymbol{Z}\boldsymbol{Y}^{\boldsymbol{T}}) = \boldsymbol{T}\boldsymbol{Y}^{\boldsymbol{T}} \quad [14]$$

$$\boldsymbol{V}(\boldsymbol{Y}\boldsymbol{Z}^{\boldsymbol{T}}) + \boldsymbol{W}(\boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}} + \lambda_2 \boldsymbol{I}) = \boldsymbol{T}\boldsymbol{Z}^{\boldsymbol{T}} \quad [15]$$

The analytical solution to the learning dilemma:

$$[\boldsymbol{V}, \boldsymbol{W}] = [\,\boldsymbol{T}\boldsymbol{Y}^{\boldsymbol{T}}, \boldsymbol{T}\boldsymbol{Z}^{\boldsymbol{T}}] \begin{bmatrix} \boldsymbol{Y}\boldsymbol{Y}^{\boldsymbol{T}} + \lambda_1 \boldsymbol{I} & \boldsymbol{Z}\boldsymbol{Y}^{\boldsymbol{T}} \\ \boldsymbol{Y}\boldsymbol{Z}^{\boldsymbol{T}} & \boldsymbol{Z}\boldsymbol{Z}^{\boldsymbol{T}} + \lambda_2 \boldsymbol{I} \end{bmatrix}^{-1} \quad [16]$$

The pseudo-code for creating this design is shown in Algorithm 1. Employing high-level domain-specific deep learning tools like Keras, it is trivial to transform such pseudo code into actual implementation. Fine-tuning the model parameters is considerably more complex and time-consuming in practice.

Algorithm 1: Pseudo Code of the spectrum sensing detection
Define Model:
Model.add(Embedding)
Model.add(CNN)
Model.add(LSTM)
Model.add(Dropout)
Model.add(Dense)

$$\text{Model.add}((\text{Activation})$$
$$\text{Compile Model:}$$
$$\text{Model.compile}()$$

IV. EXPERMENT AND RESULT

In order to evaluate the dependability of the proposed system, we made use of the following criteria for suggested and well-known state-of-the-art methodologies. The well-known state of the art method is put into practice and evaluated using the exact same training and testing procedure on the same signal data set.

$$\boldsymbol{Accuracy} = \frac{True\ Positive\ SU\ signal + True\ Nagative\ SU\ signal}{Positive\ SU\ signal + Negative\ SU\ signal}$$

$$\boldsymbol{Sensitivity} = \frac{True\ Positive\ SU\ signal}{Positive\ SU\ signal}$$

$$\boldsymbol{Specificity} = \frac{True\ Nagative\ SU\ signal}{True\ Negative\ SU\ signal + False\ Positive\ SU\ signal}$$

$$\boldsymbol{Precision} = \frac{True\ Positive\ SU\ signal}{True\ Positive\ SU\ signal + False\ Positive\ SU\ signal}$$

For the most part, the evaluation of the performance of the machine learning model is based on accuracy. Precision is the factor that is responsible for defining the significance of the discoveries that are generated by predictions. It is the responsibility of the recall to quantify the number of tweets that have been correctly detected, and the F1-Score is a weighted combination of the recall and accuracy ratings. A great number of observations and conclusions may be drawn from the results of the implementation. The overall performance scores across all of the approaches have begun to show signs of improvement. Taking into consideration the findings, one might reach the following conclusions: With a precision of 96% and an accuracy of 98%, the DS-CNN-LSTM model that was proposed for spectrum identification was able to reach the highest possible results.

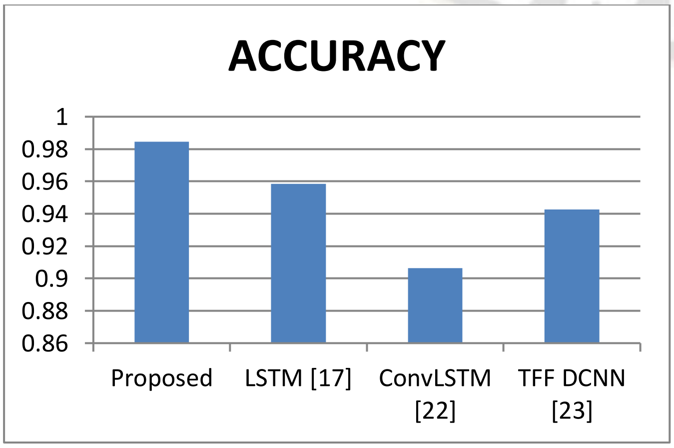

Fig. 2: Accuracy

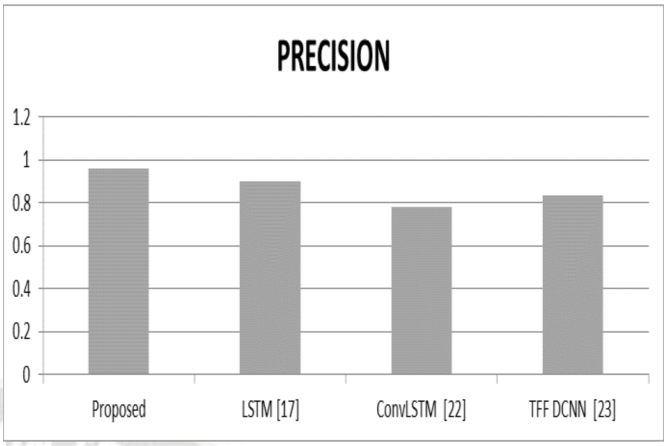

Fig. 3: Precision

V. CONCLUSION

When it comes to establishing ultra-dense and ultra-large-capacity intelligent connections of everything, intelligent wideband spectrum sensing technology is very necessary in the future of beyond 5G (B5G) and 6G. On the other hand, attaining high-precision and high-reconstruction-capability wideband spectrum sensing (WSS) with very low signal-to-noise ratio (SNR) continues to be a challenge over an extremely large frequency range. In the present study, we provide a spectrum sensing approach that is based on DS-CNN-LSTM. This method is capable of learning implicit properties from spectrum data, such as temporal correlation, which refers to the connection between the current timestamp and the timestamps that came before it.

REFERENCES

[1] J. Lunden, V. Koivunen, and H. V. Poor, ``Spectrum exploration and exploitation for cognitive radio: Recent advances," IEEE Signal Process. Mag., vol. 32, no. 3, pp. 123_140, May 2015.
[2] 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.
[3] J. Mitola and G. Q. Maguire, ``Cognitive radio: Making software radios more personal," IEEE Pers. Commun., vol. 6, no. 4, pp. 13_18, Aug. 1999.
[4] S. Haykin, ``Cognitive radio: Brain-empowered wireless communications," IEEE J. Sel. Areas Commun., vol. 23, no. 2, pp. 201_220, Feb. 2005.
[5] X. Chen, H. Zhang, A. B. MacKenzie and M. Matinmikko, "Predicting spectrum occupancies using a nonstationary hidden markov model," IEEE Wireless Communications Letters, vol. 3, no. 4, pp. 333–336, 2014.
[6] Y. Saleem and M. H. Rehmani, "Primary radio user activity models for cognitive radio networks: A survey," Journal of Network and Computer Applications, vol. 43, pp. 1–16, 2014.
[7] T. Nguyen, B. L. Mark and Y. Ephraim, "Spectrum sensing using a hidden bivariate markov model," IEEE Transactions on Wireless Communications, vol. 12, no. 9, pp. 4582–4591, 2013.
[8] H. Urkowitz, "Energy detection of unknown deterministic signals," Proceedings of the IEEE, vol. 55, no. 4, pp. 523–531, 1967.
[9] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 116–130, 2009.
[10] R. Tandra and A. Sahai, "SNR walls for signal detection," IEEE Journal of Selected Topics in Signal Processing, vol. 2, no. 1, pp. 4–17, 2008.
[11] H.Wang, E. H. Yang, Z. Zhao andW. Zhang, "Spectrum sensing in cognitive radio using goodness of fit testing," IEEE Transactions on Wireless Communications, vol. 8, no. 11, pp. 5427–5430, 2009.

[12] G. Zhang, X. Wang, Y. C. Liang and J. Liu, “Fast and robust spectrum sensing via kolmogorov–Smirnov test,”IEEE Transactions on Communications, vol. 58, no. 12, pp. 3410–3416, 2010.
[13] D. K. Patel and Y. N. Trivedi, “LRS-G 2 based non-parametric spectrum sensing for cognitive radio,” in Int. Conf. on Cognitive Radio Oriented Wireless Networks, Grenoble, France, pp. 330–341, 2016.
[14] D. K. Patel, B. Soni and M. López-Benítez, “Improved likelihood ratio statistic based cooperative spectrum sensing for cognitive radio,” IET Communications, vol. 14, no. 11, pp. 1675–1686, 2019.
[15] J. Tian, Y. Pei, Y. D. Huang and Y. C. Liang, “Modulation-constrained clustering approach to blind modulation classification for MIMO systems,” IEEE Transactions on Cognitive Communications and Networking, vol. 4, no. 4, pp. 894–907, 2018.
[16] K. Yang, Z. Huang, X.Wang and X. Li, “A blind spectrum sensing method based on deep learning,” Sensors, vol. 19, no. 10, pp. 1–17, 2019.
17] D. Janu, K. Singh, S. Kumar and S. Mandia, "Hierarchical Cooperative LSTM-Based Spectrum Sensing," in IEEE Communications Letters, vol. 27, no. 3, pp. 866-870, March 2023, doi: 10.1109/LCOMM.2023.3241664.
18] A. Rojas and G. J. Dolecek, "Narrowband Spectrum Sensing: Fuzzy Logic Versus Deep Learning Systems," 2023 27th International Conference Electronics, Palanga, Lithuania, 2023, pp. 1-6, doi: 10.1109/IEEECONF58372.2023.10177594.
19] G. Pan et al., "Deep Stacked Autoencoder-Based Long-Term Spectrum Prediction Using Real-World Data," in IEEE Transactions on Cognitive Communications and Networking, vol. 9, no. 3, pp. 534-548, June 2023, doi: 10.1109/TCCN.2023.3254524.
20] D. Nie, W. Yu, Q. Ni, H. Pervaiz and G. Min, "Cluster Control and Energy Consumption Minimization for Cooperative Prediction Based Spectrum Sensing in Cognitive Radio Networks," in IEEE Transactions on Communications, doi: 10.1109/TCOMM.2023.3287531.
21] K. Chae and Y. Kim, "DS2MA: A Deep Learning-Based Spectrum Sensing Scheme for a Multi-Antenna Receiver," in IEEE Wireless Communications Letters, vol. 12, no. 6, pp. 952-956, June 2023, doi: 10.1109/LWC.2023.3250257.
22] Q. Wang, B. Su, C. Wang, L. P. Qian, Y. Wu and X. Yang, "ConvLSTM-Based Spectrum Sensing at Very Low SNR," in IEEE Wireless Communications Letters, vol. 12, no. 6, pp. 967-971, June 2023, doi: 10.1109/LWC.2023.3254048.
23] X. Li, Z. Hu, C. Shen, H. Wu and Y. Zhao, "TFF_aDCNN: A Pre-Trained Base Model for Intelligent Wideband Spectrum Sensing," in IEEE Transactions on Vehicular Technology, doi: 10.1109/TVT.2023.3271970.
24] J. Xue, L. Zhao, T. Wang, L. Zhao and F. Cui, "A Novel Compact Fiber Optic Concentration Sensing System Based on Machine Learning Demodulation," in IEEE Photonics Journal, vol. 15, no. 4, pp. 1-6, Aug. 2023, Art no. 7100906, doi: 10.1109/JPHOT.2023.3290984.
25] Prakash Chauhan, Nityananda Sarma, Sanjib K. Deka, “LSTM-enabled prediction-based channel switching scheduling for multi-channel cognitive radio networks”, Physical Communication, 2023, 102136, ISSN 1874-4907, https://doi.org/10.1016/j.phycom.2023.102136.
26] Mingqian Liu, Xiaobo Zhang, Yunfei Chen, Hui Tan, Multi-antenna intelligent spectrum sensing in the presence of non-Gaussian interference, Digital Signal Processing, Volume 140, 2023, 104135, ISSN 1051-2004, https://doi.org/10.1016/j.dsp.2023.104135.
27] B. Soni, D. K. Patel and M. López-Benítez, “Long short-term memory-based spectrum sensing scheme for cognitive radio using primary activity statistics,” IEEE Access, vol. 8, pp. 97437–97451, 2020.
28] K. Yang, Z. Huang, X.Wang and X. Li, “A blind spectrum sensing method based on deep learning,” Sensors, vol. 19, no. 10, pp. 1–17, 2019