# Accuracy Enhancement in 5G, 4G, Radar, and Gaussian Noise Spectrum Sensing with Deep Learning Approaches

Dainty S. Fariscal [1,]* and Lawrence Materum [1,2]

[1] Department of Electronics and Computer Engineering, De La Salle University, Manila 1004, Philippines
[2] International Centre, Tokyo City University, Tokyo, 158-8557, Japan
Email: dainty_fariscal@dlsu.edu.ph (D.S.F.); materuml@dlsu.edu.ph (L.M.)
*Corresponding author

***Abstract*—Wireless communication is continuously evolving. Signals compete in the spectrum as there is a demand for data transfer. There is also a growing demand for radio spectrum, resulting in congestion. With the advancement of deep learning, there are numerous applications in which it could be utilized; this could be used as a solution to the current issue. Given the state of wireless communication, this research aims to achieve a higher accuracy rate regarding spectrum sensing of noise, Long-Term Evolution (LTE), New Radio (NR), and radar through deep learning. The following methods are used: a) different Cellular Neural Network (CNN) models such as resnet18, resnet50, mobilenetv2, etc., b) modifying segments of code, c) manipulation of the dataset. For the results, the modified normalized confusion matrix values significantly increased compared to the pre-modified version. For the spectrogram sensing, the received spectrogram, true signal labels, and estimated signals were dominated mainly by noise. Based on the graph, the frame mean Intersection over Union (IoU) has noticeably increased; however, based on data metrics, the mean IoU decreased. There is an increase in global accuracy, mean accuracy, Weighted IoU, and Mean BF Score. In conclusion, the objective of increasing the accuracy rate was met through various modifications, such as using different CNN architectures, parameter adjustments, and manipulating the given data set. However, the decrease in the mean IoU should also be considered, as it is also part of the accuracy measurement.**

***Keywords*—5G, deep learning, radar, signals, spectrum sensing**

## I. Introduction

Wireless communications has become widespread over the years of modern technology. According to the study of Barb and Otesteanu [1], wireless communication systems such as Long Term Evolution (LTE/4G) and New Radio (5G/NR) are continuously evolving. With this trend in the community, there is a higher demand for data transfer. Due to the intense demand, there is competition for the bandwidth and latency in the spectrum. In addition, aside from the existing wireless communication, there are on-going deployments for other innovative networks such as the 6th Generation Mobile System (6G). Furthermore, with wireless communication such as LTE and 5G, there is also a growing demand for radio spectrum for radar, resulting in congestion [2].

Given the state of wireless communication, this research aims to achieve a higher accuracy rate in terms of spectrum sensing of airport surveillance radar (ASR/Radar), LTE, 5th generation technology (5G/NR), and Gaussian noise (Gauss). This paper is divided into several sections, including the topic introduction, the methodology used, the results acquired, the discussion about the comparison of the existing related studies, and the overall conclusion for this paper.

## II. Literature Review

### A. Integrated Sensing and Communication

The concept of integrated sensing and communications refers to the combination of both wireless and radar sensing. It acts as a single system wherein it senses, collects, and communicates the information in a single go, compared to the previous architecture/s, where both signals are treated differently. As per the study of Liu *et al.* [3], ISAC is expected to considerably improve spectral and energy efficiencies. ISAC is a design methodology that integrates sensing and communication functionalities to achieve efficient usage and mutually benefit one another. In the case of implementing additive white Gaussian noise for both radar and wireless connections, one might think that these noises have only detrimental effects on the system; however, in a real scenario, this noise plays a crucial role in the ISAC model as it creates a realistic channel, ensuring that the system can handle real-life conditions.

### B. Deep Learning in Radar and Wireless Communication

With the ongoing state of deep learning, there are numerous applications in which it could be utilized. In the context of radar and wireless communication, deep learning could be used as a solution to the current spectrum

Manuscript received July 29, 2025; revised September 24, 2025, accepted October 5, 2025; published January 28, 2026.

doi:10.12720/jcm.21.1.49-56

traffic problem. As Nasser *et al.* [4] stated, machine learning and deep learning have been employed in wireless communication to manage the spectrum efficiently. In cognitive radio systems, machine learning and deep learning were also used to enhance spectrum sensing performance. There are two phases involved in learning techniques, namely: learning and prediction. In traditional spectrum sensing, secondary users must determine the threshold for the test statistic before deciding if primary users are present. Various parameters, such as noise, channel, and primary users, should be determined beforehand, as the threshold can be based on false alarm and detection rates. Applying machine learning and deep learning eliminates the need for the parameters.

### *C. The Current Challenges in Spectrum Sensing*

In spectrum sharing, signals share the vacant bands to avoid collisions, including three major spectrum access, selection of channels, and spectrum allocation [5]. Spectrum Decision deals with the analysis of the band and is responsible for decision-making [2]. The efficient implementation of cognitive radios into wireless networks gives rise to a tremendous number of optimization problems with multiple objectives [6]. Another issue related to radar is the increase in bandwidth requirement. However, it was addressed by Reed *et al.* [7] through fixed frequency bands; it is still seen as an issue, as bandwidth plays a key role in performing operations. The Federal Communications Commission indicates that fewer radio frequencies are conducive to operating cost-effective wireless cellular and mobile broadband [8]. Several Cognitive Radios working in the same frequency band of primary users can severely affect the spectrum sensing process [9]. Given the state of wireless communication, this study aims to achieve a higher accuracy rate regarding spectrum sensing of noise, LTE, NR, and Radar through deep learning.

## III. Materials and Methods

### *A. Spectrum Sensing Flowchart using Deep Learning*

Fig. 1 provides a detailed flowchart describing the code simulation's overall structure in MATLAB. This chart is a visual guide to understand the data processing workflow and the interactions between each sub-category. The latter part provided a detailed explanation involving the sequence of data processing throughout the code.

#### *1) Setup of the waveform*

The first step was to set up the waveform used in radar or wireless systems. The signal's center frequency is 2.8 GHz with a sampling frequency of 62.44 MHz. It then calculated the pulse repetition by almost 1050 per second with a pulse width of 1 microsecond. Finally, a RectangularWaveform function was used.

#### *2) Setup of the antenna*

The following step is to set up the antenna; this segment defines the parabolic parameters of an antenna: the radius and focal length of 2.5 meters. The horn exciter and the reflector were tilted 90°. Finally, the conformal array function was used for the shape of the antenna.

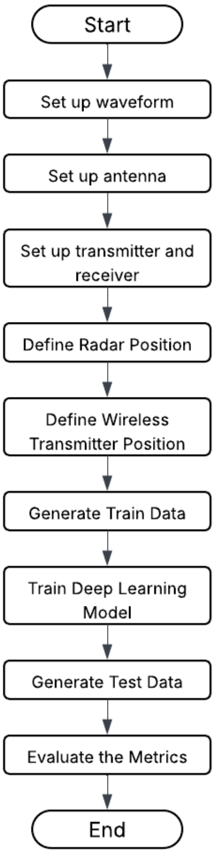

Fig. 1. Flowchart illustrating the structured flow from setting up the variables, generating the data, and evaluating the metrics in MATLAB.

#### *3) Setup of the transmitter and receiver*

The next step focused on the setup for the transmitter and receiver. The transmit power is set to 25 kW while the transmit gain is 32.8% which is defined through the use of phased.Transmitter function. In addition, the rxpos_horiz and rxpos_vert are the ranges for the receiver.

#### *4) Define radar position*

For the radar position, the radar transmitter's position was set at (0,0,5) meters, and the Scattering MIMO Channel function was used to create a radar environment. Under the ConformalArray function, parameters such as CarrierFrequency, SpecifyFrequency, SpecifyAtmosphere, etc, were used to simulate the environment.

#### *5) Definition of the transmitter position*

In defining the wireless transmitter position, this segment positioned the transmitter at (200.0.450) using a taper type condition under the ConformalArray function. Similar to the previous code segments, this section also created its environment.

#### *6) Generation of the training data*

A zip file from an online source, Ref. [10], was extracted to generate the train data, containing both the training and testing datasets. This part was intended to set up the machine learning model to recognize the signals, LTE, 5G, Radar, and Noise. The training image was set to 50 while the image size was [480 640 3]. The images were then organized according to their labels and split into training and validation sets. The next part focused on the pixel-level classifier and the deep learning model.

*7) Training of the deep learning model*

As for the deep learning model, this is the primary focus of spectrum sensing. The code utilized DeepLabV3+ with a backbone of MobileNetV2. Under the training Options function, minibatch size, maxepochs, learn rate schedule, etc., were defined. Additional data augmentation was also incorporated, including Rand Rotation, Rand X Reflection, Rand Y Reflection, etc. After the augmentations, the data was trained. Since the train Now is set to false, the pre-trained model was loaded.

*8) Generation of the testing data*

To generate a test data segment, the code generated 120 images for evaluating the trained data. It then used the helper Generate Radar Comm Data to simulate the radar and communication data. After the test data were generated, it was loaded using image Data Store; the semantics function was used to apply the testing data to the training data, in which image segments are classified by predicting the type of signal for each pixel.

*9) Evaluation of the performance metrics*

In the last segment of the code, the pixel Label Datastore function was used to identify the truth labels of the image by reading the actual class label from PDF files. The true and predicted labels were compared, and the evaluate Semantic Segmentation function was used to calculate data metrics. The expected visual outputs are the Confusion Matrix, Received Spectrogram, True Signal Labels, Estimated Signal Labels, and the Mean IoU, which showcases the accuracy values of the given data set within the created simulation.

*D. Modifications for Increasing the Accuracy of the Spectrum Sensing Model*

With the given flowchart of the code, a series of alterations within the parameters were made to increase the accuracy of spectrum sensing. This modification includes the trial and error in using different models for the convolutional neural network, specific values for each function in the given code, and the modification of the data sets between the train and test folders. These steps were necessary to determine slight differences that could significantly influence the resulting output values.

*1) Convolutional neural network in MATLAB*

A convolutional neural network type is one of the deep learning methods used in the study, in which various models were used to find the best-fitting model that could yield a significantly higher accuracy. In addition, according to a study by Qin *et al.* [11], MobileNetV2 has faster identification speed. It requires fewer computational resources than traditional deep CNNs. Before encountering MobileNetV2, several models were also tested, such as ResNet18 [12], ResNet50, MobileNetV2, Xception, InceptionResNetV2, EfficientNetb0, and DenseNet201; however, all models failed to improve the accuracy slightly. From the pre-modified version, the base network is ResNet50. Upon testing the different models, the MobileNetv2, while less complex, is greater than ResNet50.

*2) Code modifications*

Various modifications are made in the original code on spectrum sensing with deep learning for radar and wireless communications. Table I shows the different modifications in terms of the code given by MATLAB.

TABLE I. CODE DIFFERENCE BETWEEN PRE-MODIFIED AND MODIFIED

| **Pre-modified** | **Modified** | **Used Values** |
|---|---|---|
| Num Training Data=500 | Num Training Data = 450 | 500,450 |
| ng | opts = training Options (“adam”) | sgdm, adam |
| Mini Batch Size = 40 | Mini Batch Size = 20 | 40,80,100,20 |
| Max Epochs = 40 | Max Epochs = 20 | 40,80,100,20 |
| Initial Learn Rate = 0.02 | Initial Learn Rate = 0.0001 | 0.02,0.01,0.0001 |
| Learn Rate Drop Period = 10 | Learn Rate Drop Period = 50 | 10,30,50 |
| Learn Rate Drop Factor = 0.01 | Learn Rate Drop Factor = 0.05 | 0.01,0.05 |
| Validation Patience = 5 | Validation Patience = 30 | 5,15,30 |
| N/A | rxpos_horiz_minimax = [−1000 1000] | N/A |
| N/A | rxpos_vert_minimax = [0 2000] | N/A |

*3) Radar communication sensing spectrum data*

The original dataset [10] has been modified for this study. The folder includes two subfolders: RadarCommTrainData with a total of 8802 various files and RadarCommTestData with 1765 files. The ASR, LTE, 5G, and Gauss are the signals that can be seen on the spectrum. The researcher did an 80/20% file hatching in which 20% of a specific data type from the folder RadarCommTrainData was moved to the RadarCommTestData folder; the summary of the original number of files of both folders, RadarCommTestData and RadarCommTrainData, can be seen in Table II. For example, the data type label_ASR_5G_gauss*.hdf has a total of 500 files, 20% of which are the files from 401 to 500, which were moved to the other folder. For other files with a total of 467, such as data_ASR_5G_LTE_gauss.png, only the remaining 67 were transferred.

The images shown in Fig. 2 are the sample files from the folder RadarCommTrainData. The image on the left, labeled as data_asr_lte1.png, shows strokes of mostly blue signals. However, notice a small band mixture of red and yellow in the middle. This band signifies signals, specifically ASR and LTE, on this data image. On the other hand, the image on the right, labeled data_asr_lte_gauss1.png, is a fuzzy image showing scattered patches of yellow, green, and blue, which can suggest that the signals detected in this image are mostly noise. Both images have the labels for ASR and LTE; however, when the Gaussian noise was added, it overlaid most of the signals. This addition might affect the detection of signals on the spectrum due to the noise. However, note that there are 400 images to be trained for each data type.

TABLE II. ORIGINAL NUMBER OF FILES OF RADAR COMM TRAIN DATA AND RADAR COMM TEST DATA FOLDERS

| File Name | Number of Files (Radar Comm Train Data) | Number of Files (Radar Comm Test Data) |
|---|---|---|
| label_ASR_5G.hdf | 500 | 100 |
| label_ASR_5G_LTE_gauss.hdf | 467 | 94 |
| label_ASR_5G_LTE.hdf | 467 | 94 |
| label_ASR_5G.hdf | 500 | 100 |
| label_ASR_LTE_gauss.hdf | 500 | 100 |
| label_ASR_LTE.hdf | 500 | 100 |
| data_ASR_5G_gauss.png | 500 | 100 |
| data_ASR_5G_LTE_gauss.png | 467 | 94 |
| data_ASR_5G_LTE.png | 467 | 94 |
| data_ASR_5G.png | 500 | 100 |
| data_ASR_LTE_gauss.png | 500 | 100 |
| data_ASR_LTE.png | 500 | 100 |
| data_ASR_5G_gauss.png_original | 500 | 100 |
| data_ASR_5G_LTE_gauss.png_original | 467 | 94 |
| data_ASR_5G_LTE.png_original | 467 | 94 |
| data_ASR_5G.png_original | 500 | 100 |
| data_ASR_LTE_gauss.png_original | 500 | 100 |
| data_ASR_LTE.png_original | 500 | 100 |
| label_ASR_5G.hdf | 500 | 100 |
| label_ASR_5G_LTE_gauss.hdf | 467 | 94 |
| Total | *8802* | *1765* |

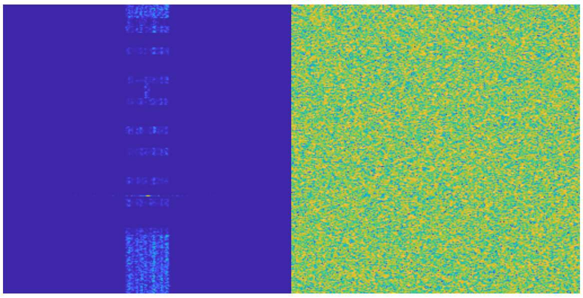

Fig. 2. Comparison of data signals from the radar comm train data folder.

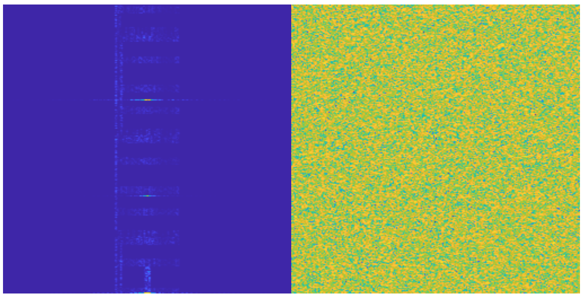

Fig. 3. Comparison of data signals from the radar comm test data folder.

Fig. 3 shows the sample images from the Radar Comm Test Data folder. On the left is the data_asr_lte1.png, where distinct strips of colored lines are clearly visible. These strips indicate that ASR and LTE can be detected on the image. On the right, the data_asr_lte_gauss1.png similarly shows the same pattern as the previous figure, wherein the overall image became vague with the added Gaussian noise, making it difficult to detect signals. Notice that both images exhibit the same structure (fuzzy image with Gaussian noise added). It is important to note that both are from different datasets, one is for training and the other is for testing. To increase the accuracy of the created deep learning approach, some of the images from the training images were transferred to the folder of the testing dataset folder.

### *E. ISAC Receiver Design*

The concept for this equation was based on an ISAC receiver that received communication signals from a signal while experiencing interference from a sensing system [3]. The general formula of the received code was expressed as:

$$\mathcal{Y}(t) = \mathcal{Y}_C(t) + \mathcal{Y}_R(t) + z(t)$$

$$\mathcal{Y}(t) = \underbrace{\textstyle\sum_{n=0}^{N-1} \mathcal{X}(n)\,\mathcal{E}(t - nT)}_{\text{Comms Signal}} + \underbrace{\sum_{j=1}^{J}\sum_{n=0}^{N-1} c_j g_j(n)\,\mathcal{E}\left(t - nT - \tau_j\right)}_{\text{Radar Interference}} + \underbrace{z(t)}_{\text{Noise}} \tag{1}$$

Eq. (1) signifies the sum of communication signals, coded radar signal, and noise in the time domain. $\mathcal{Y}(t)$ stands for the total received signal at time $(t)$; $\mathcal{Y}_C(t)$ is the signal generated from the communication, $\mathcal{Y}_R(t)$ stands for radar, and $z(t)$ is for the noise. The $\varepsilon(t)$ is the basic pulse for the Nyquist criterion, for a duration of $T$. $N$ is the code length for both radar and communication. The $\tau_j$ and

$c_j$ is for the delay and complex channel between the $j^{\text{th}}$ radar signal. The $g_j(n)$ is the nth code for the $j^{\text{th}}$ radar signal; and lastly, $\mathcal{X}(n)$ is the n$^{\text{th}}$ communication signal sample [3].

In this design, the system receives a mix of communication and radar signals. The concept of sparsity, wherein only a few radar signals are active, allows the receiver to isolate and retrieve both the radar and communication signals. Through this, decoding the signal is possible without the use of specific software.

## IV. Results and Discussions

### A. Normalized Confusion Matrix

Figs. 4−5 highlight the normalized confusion matrices created using MATLAB. Fig. 4 illustrates the created accuracy rate for LTE, which is 93.8%, 80.5% for 5G New Radio, 92.5% for noise, and 99.2% for radar. Following the reconfigurations, the new matrix on Fig. 5 displays a new set of accuracy rate for LTE which is 94.6 % (increase rate of 0.8%); for 5G, the rate decreased by 1.0% resulting into a 79.5% accuracy rate; for noise, the increase rate was 0.3% resulting to 92.8%; and lastly, for radar, an increase of 0.3% was created resulting into 99.5% accuracy rate.

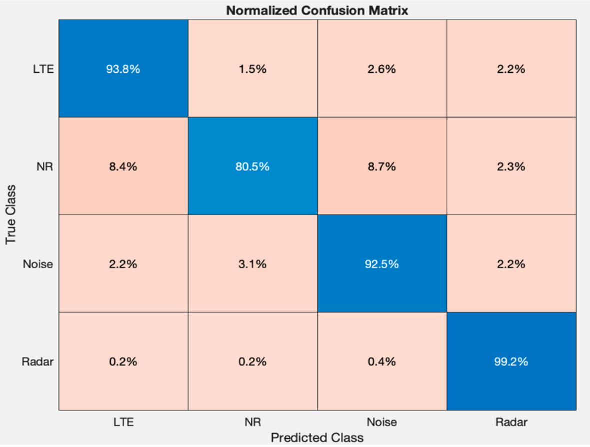

Fig. 4. Pre-modified normalized confusion matrix.

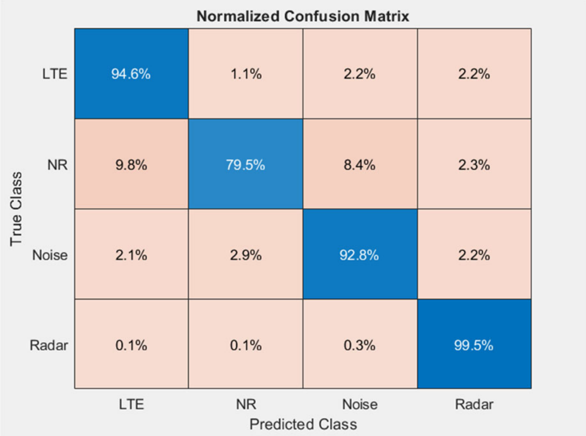

Fig. 5. Modified normalized confusion matrix.

### B. Spectrogram Sensing

In Figs. 6 and 7, a received spectrogram was analyzed using the spectrogram function. Prior to reconfiguration, the latter portion displays an increased portion of bright blue regions, which is most likely noise. As for the true signal labels, the 5G signal dominated the pre-modifications spectrogram compared to the latter version, in which the noise occupied the spectrogram mostly. Similarly, for the estimated signal labels, the original spectrogram shows 5G as the prevailing signal, compared to the revised version in which the noise becomes the dominant classification. Notably, the radar lines on the latter version appeared fragmented compared to before, where the lines were continuously connected.

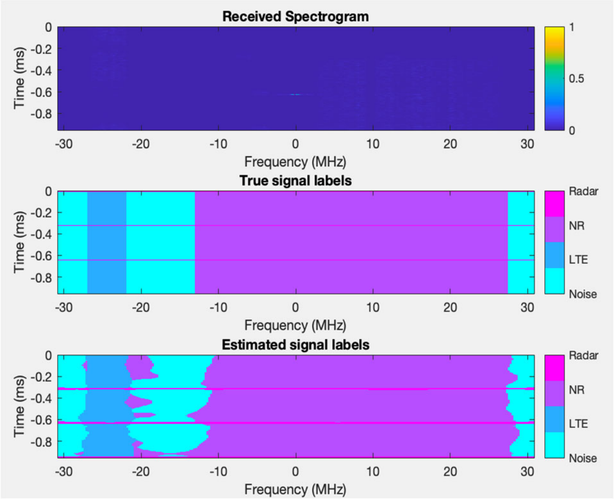

Fig. 6. Pre-modified spectrum sensing including received spectrogram, true signal labels, and estimated signal labels.

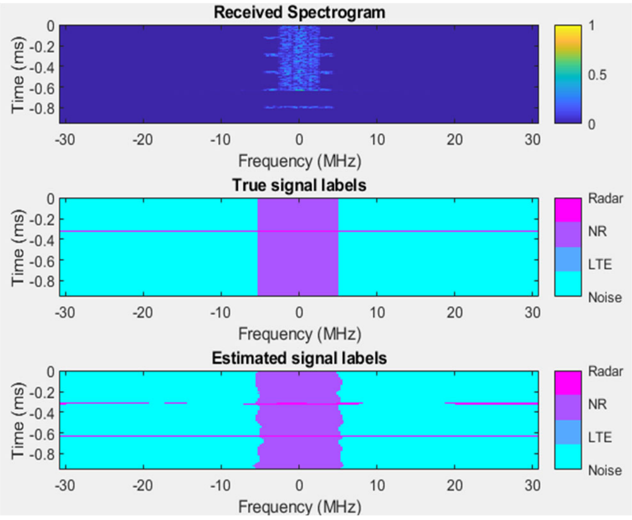

Fig. 7. Modified spectrum sensing including received spectrogram, true signal labels, and estimated signal labels.

### C. Frame Mean IoU

Figs. 8−9 present the Frame Mean IoU, which compares the predicted and the true labels across the number of frames. In these models, it is notably visible that there is a difference after the modifications. The following are the discrepancies between sets of given IoU and number of frames: for IoU values between 0.2−0.3, it can be noticed that on the right model, the bar graph is slightly above the 20 point mark of number of frames; for 0.3−0.4, the number of frames exceeded the 100 point; for 0.5−0.6, though it exceeded the 50 point, there is still a difference between the graphs; for both sectors 0.7−0.8 and 0.8−0.9, both points were increased exceeding the 100 point frame mark.

TABLE III. DATA METRICS BETWEEN PRE-MODIFIED AND MODIFIED MODEL

| Data Version | Global Accuracy | Mean Accuracy | Mean IoU | Weighted IoU | Mean BF Score |
|---|---|---|---|---|---|
| Pre-modified | 0.89313 | 0.91483 | 0.70508 | 0.81645 | 0.83124 |
| Modified | 0.89442 | 0.91588 | 0.70277 | 0.81945 | 0.84242 |

### D. *Data Set Metrics*

Table III presents the overall data metrics obtained using semantic segmentations between noise, 4G, 5G, and Radar signals. The global accuracy shows an increase of 0.14% increase; for mean accuracy, the increase is equivalent to 0.11%; mean IoU experienced a decrease of about 0.32%; for weighted IoU, there is an increase of 0.36%; lastly, the mean BF Score improved by 1.34%.

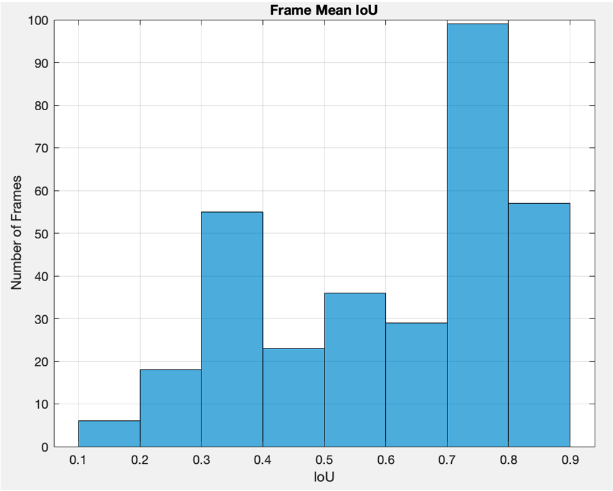

Fig. 8. Pre-modified frame means IoU.

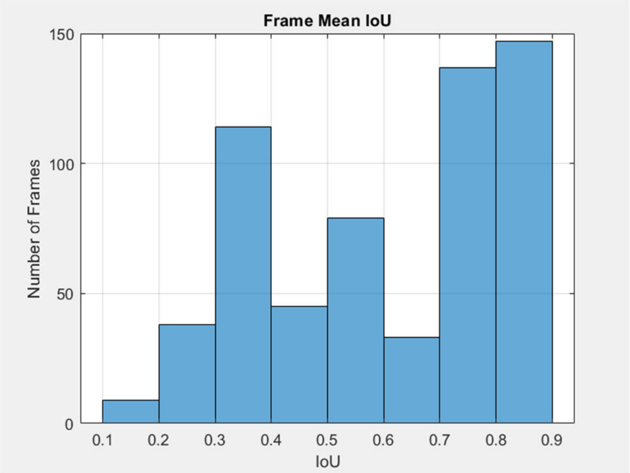

Fig. 9. Modified frame mean IoU.

### E. *Accuracy in Using Spectrum Sensing with Deep Learning*

Based on the acquired data, an increase in the diagonal values of the normalized confusion matrix was observed. The increase in the diagonal cells indicates a higher accuracy [13]. This outcome also shows that the code and data modifications create a better prediction at identifying the signals. However, despite the improvement in correct predictions, there are still misclassifications, as changes in the off-diagonal values were evident. Jayapalan and Anuar [14] indicated that the diagonal cells relate to accurately classified observations. In contrast, the off-diagonal cells correspond to observations that were inaccurately classified. The model demonstrated an improved accuracy in predicting LTE, noise, and radar, except for NR, since the diagonal value decreased.

As for the spectrogram sensing function, based on the received spectrogram, true signal labels, and estimated signal labels, the presence of noise became prevalent among the spectrograms. In addition, according to Ref. [15], the background noise in spectrograms may hide the foreground pixels (signals) in the noise. There could be three interpretations within these results: a) the given dataset has a higher segment of noise compared to the other signals, b) the model might be weakly labeling the other signals as noise, and c) the model used might be too sensitive in a way that any uncertain points were identified as noise.

Proceeding to the IoU, based on the bar graph, it can be noticed that there is a slight “increase” within each segment; however, based on the Mean IoU of Table 3, a decrease was observed. This result could mean that the predicted regions overlap the true labels, creating misclassifications. However, in terms of the increase of global accuracy, mean accuracy, Weighted IoU, and Mean BF Score, it was suggested that a better model with improved performance was used.

#### *1) Accuracy comparison between machine learning models*

To compare the accuracy result, related literature in spectrum sensing was used. However, these past studies were primarily focused on identifying 5G and LTE signals only.  A study by Huynh *et al.* [18] found that through using the model of a Convolutional Network (ConvNet), the simulation achieved a 95% mean accuracy and 91% mean IoU at a medium SNR level. On the other hand, a study from Nguyen *et al.* [19], with the same medium SNR level, the improved DeepLabV3+ model achieved a global accuracy of 75% while the mean IoU is at 59% which is mainly affected by the components’ Adaptive Dilation Rate (ADR) and Attention Mechanism (ATM). The same study also used MobileNetV2, ResNet18, and ResNet50 backbones. In comparison with the current study, the value for MobileNetV2 is 74.44% with the use of both ATM and ADR, compared to the global accuracy of the current study, which is 89.44%. However, in the case of the study of Nguyen *et al.* [20], it was noted that ConvNet and DeeplabV3+ exhibit lower accuracy, with rates of 39.69%

and 46.26%, respectively, negatively impacting their predictive performance. It can be reasonably assumed that various factors, such as the data sets used and the methodology employed during the simulation with these studies, might have influenced the resulting values.

### *F. T-Test Statistical Analysis for Pre-modified and Modified Model*

Table IV depicts the use of the t-test analysis to test if the data acquired have a significant difference. The data used for the t-test are the raw percentages from the normalized confusion matrices of both the pre-modified and modified models. The test employed a one-tailed distribution with a paired type design. The obtained *p*-value from the test is 0.40596806. Since the *p*-value is higher than the usual alpha level (0.05), it can be interpreted that there is no significant difference between the two sets of data.

TABLE IV. T-TEST ANALYSIS FROM NORMALIZED CONFUSION MATRIX DATA

| Data Type | Pre-modified | Modified |
|---|---|---|
| LTE | 93.80% | 94.60% |
| NR | 80.50% | 79.50% |
| Noise | 92.50% | 92.80% |
| Radar | 99.20% | 99.50% |
| | | *p*-value = 0.40596806 |

## V. CONCLUSION

In conclusion, the research objective was met by increasing the accuracy rate through utilizing different CNN architectures, manipulation of the data set, and adjusting the parameters of the code. However, even though the accuracy rate increased, the decrease in the mean IoU should also be considered, as it is also part of the accuracy measurement. What can be inferred from this study is that the manipulation of the dataset created a significant difference. The model used can also be considered in a way that alters the accuracy rate based on its complexity compared to the previous models used. Although different CNN models were used, dataset manipulation proved most significant, yielding improvements when comparing the pre-modified and modified normalized confusion matrix. This outcome also indicates that the study increased the accuracy; however, statistical analysis determined no significant difference in comparing the results of pre-modified and modified models.

For recommendation, it is suggested that future researchers pursuing similar studies select a compatible model (not a generic nor too complex) for the data to be accurate and aligned with one another. Another recommendation is to use another proportion (60/40 or 50/50) for the data set, which will also determine the accuracy. Lastly, filtering, noise level, or thresholding techniques can reduce noise factors and improve classifications.

## CONFLICT OF INTEREST

The authors declare no conflict of interest.

## AUTHOR CONTRIBUTIONS

D. Fariscal conducted the methodology, software implementation, simulations, machine learning analysis, data curation, and wrote the original draft; L. Materum supervised the project and contributed to conceptualization, data curation, and critical review; both authors reviewed, edited, and approved the final manuscript.

## FUNDING

De La Salle University and Engineering Research and Development for Technology (ERDT) funded this research.

## ACKNOWLEDGMENT

The authors acknowledge De La Salle University for its support.

## REFERENCES

[1] G. Barb and M. Otesteanu, “4G/5G: A comparative study and overview on what to expect from 5G,” in *Proc. 2020 43rd International Conference on Telecommunications and Signal Processing (TSP)*, 2020, pp. 37–40.

[2] S. K. Agrawal, A. Samant, and S. K. Yadav, “Spectrum sensing in cognitive radio networks and metacognition for dynamic spectrum sharing between radar and communication system: A review,” *Physical Communication*, vol. 52, 101673, 2022.

[3] F. Liu *et al.*, “Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond,” *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 6, pp. 1728–1767, 2022.

[4] A. Nasser, H. A. H. Hassan, J. A. Chaaya, A. Mansour, and K. C. Yao, “Spectrum sensing for cognitive radio: Recent advances and future challenge,” *Sensors*, vol. 21, no. 7, 2408, 2021.

[5] M. E. Bayrakdar, “Cooperative communication based access technique for sensor networks,” *International Journal of Electronics*, vol. 107, no. 2, pp. 212–225, Feb. 2020.

[6] M. R. Ramzan, N. Nawaz, A. Ahmed, M. Naeem, M. Iqbal, and A. Anpalagan, “Multi-objective optimization for spectrum sharing in cognitive radio networks: A review,” *Pervasive and Mobile Computing*, vol. 41, pp. 106–131, 2017.

[7] J. H. Reed *et al.*, “On the co-existence of TD-LTE and radar over 3.5 GHz band: An experimental study,” *IEEE Wireless Communications Letters*, vol. 5, no. 4, pp. 368–371, 2016.

[8] H. Wang, J. Johnson, C. Baker, L. Ye, and C. Zhang, “On spectrum sharing between communications and air traffic control radar systems,” in *Proc. 2015 IEEE Radar Conference (RadarCon)*, May 2015, pp. 1545–1550.

[9] K. Arora, “Spectrum sensing issues and techniques of cognitive radio systems: A review,” *IJST*, vol. 9, no. 1, pp. 1–4, Jan. 2016.

[10] Radar comm spectrum sensing data. [Online]. Available: https://ssd.mathworks.com/supportfiles/phased/data/RadarCommSpectrumSensingData.zip

[11] Y. Qin, Q. Tang, J. Xin, C. Yang, Z. Zhang, and X. Yang, “A rapid identification technique of moving loads based on mobilenetv2 and transfer learning,” *Buildings*, vol. 13, no. 2, p. 572, 2023.

[12] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[13] Y. Liu, H. Pu, and D. W. Sun, “Efficient extraction of deep image features using Convolutional Neural Network (CNN) for applications in detecting and analysing complex food matrices,” *Trends in Food Science & Technology*, vol. 113, pp. 193–204, 2021.

[14] S. K. Jayapalan and S. Anuar, “Transfer learning based network performance comparison of the pre-trained deep neural networks using MATLAB,” *Open International Journal of Informatics*, vol. 10, pp. 27–40, 2022.

[15] A. Rojas, G. J. Dolecek, J. M. D. L. Rosa, and G. L. Cembrano, “Increasing the accuracy of spectrogram-based spectrum sensing

trained by a deep learning network using a resnet-18 model," in *Proc. 2024 39th Conference on Design of Circuits and Integrated Systems (DCIS)*, 2024, pp. 1–4.

[16] T. H. The, Q. V. Pham, T. H. Vu, D. B. da Costa, and V. P. Hoang, "Intelligent spectrum sensing with convnet for 5G and LTE signals identification," in *Proc. 2023 IEEE Statistical Signal Processing Workshop (SSP)*, 2023, pp. 140–144.

[17] G. V. Nguyen, C. V. Phan, and T. H. The, "Accurate spectrum sensing with improved DeepLabV3+ for 5G-LTE signals identification," in *Proc. 12th International Symposium on Information and Communication Technology*, 2023, pp. 221–227.

[18] D. H. Nguyen, T. V. Nguyen, and T. H.The, "Enhancing spectrum sensing for 5G and LTE with improved U-Net architecture," in *Porc. 2024 International Conference on Advanced Technologies for Communications (ATC)*, 2024, pp. 167–172.

[19] A. A. Elngar *et al.*, "Image classification based on CNN: A survey," Journal *of Cybersecurity and Information Management*, vol. 6, no. 1, pp. 18–50, 2021.

[20] J. A. Pandian, G. Geetharamani, and B. Annette, "Data augmentation on plant leaf disease image dataset using image manipulation and deep learning techniques," in *Proc. 2019 IEEE 9th International Conference on Advanced Computing (IACC)*, IEEE, 2019, pp. 199–204.