

Machine learning model to classify modulation techniques using robust convolution neural network

Nadakuditi Durga Indira, Matcha Venu Gopala Rao

Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundtion (KLEF), Vaddeswaram, India

Article Info

Article history:

Received Jun 24, 2021

Revised May 30, 2022

Accepted Jun 9, 2022

Keywords:

Deep learning

Machine learning

Neural network architecture

Batch normalization

Robust CNN

ABSTRACT

In wireless communications receiver plays a main role to recognize modulation techniques which were used at the transmitter. While transferring information from transmitter to receiver, the receiver must retrieve original information. In order to achieve this goal we introduced a neural network architecture that recognizes the types of modulation techniques. The applications of deep learning can be categorized into classification and detection. The CNN architecture is used to perform feature extraction based on the layers to build a model which classifies the input data. A model that classifies the radio communication signals using deep learning method. The robust c (RCNN) is used to train the modulated signals; the transformations are used to help the neural network which estimate the signal to noise ratio of each signal ranges from -20dB to 18dB with loss and accuracy of 89.57% at SNR 0dB.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Nadakuditi Durga Indira

Department of Electronics and Communication Engineering,

Koneru Lakshmaiah Education Foundtion (KLEF)

Green fields, Vaddeswaram, Guntur, Andhra Pradesh, 522502, India

Email: durga.indira@gmail.com

1. INTRODUCTION

Modulation identification is an important basic function of receiver. It has various applications in cognitive radar, software define radio (SDR), and spectrum management, to identify communications and radar waveforms. It's necessary to classify or identify them by the type of modulation [1]–[4]. In this paper model describes the generation of dataset in GNU Radio using GNU Radio channel model blocks and then slice each time series signal up into a test and training set using based on signal to noise ratio (SNR) values ranges from -20dB to 18dB.

The dataset consisting of 11 modulation types: 8 digital and 3 analog modulations [5]–[7]. All of these are widely used in wireless communication systems. These consists of binary phase shift keying (BPSK), quadrature amplitude modulation (QPSK), phase shift keying (8PSK), quadrature amplitude modulation (16QAM), binary frequency shift keying (BFSK), CPFSK, and PAM4 for digital modulations, and WBFM, AM-SSB, and AM-DSB for analog modulations [8]–[10]. Figure 1. Shows the Constellation diagram of modulation techniques. Traditional modulation identification methods require basic knowledge of signals and channel parameters [11]. These can be inaccurate and might need frequent changes. Since the environment changes, this leads to a new modulation identification methods using deep neural networks (DNN) [12], [13]. The total dataset is stored as a python pickle file. This data set is available as a pickled Python format at <http://radioml.com> [14] which consists of time windowed examples, corresponding modulation class and SNR labels.

Deep neural networks (DNN) play a significant role in the domain of video, speech and image processing. Recently, deep learning has been introduced in the area of communications by applying convolutional neural networks (CNN) in radio modulation classification/identification [9], [15]–[17]. The CNN has been identified in image and voice signal processing. Based on its performance in feature extraction, a simple architecture of CNN was used in distinguishing 11/10 different modulation types. Residual networks (ResNet) and densely connected networks (DenseNet) has been introduced to strengthen feature propagation in the neural networks, in the past years. Lately, a convolutional long short-term deep neural network (CLDNN) has been used in, where it combines both architectures of CNN and long short-term memory (LSTM) into a recurrent neural network (RNN).

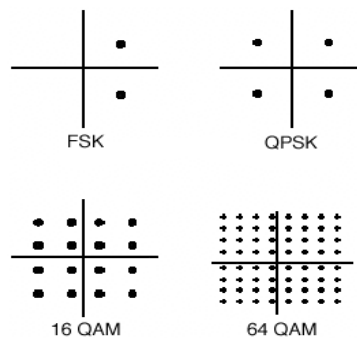


Figure 1. Constellation diagram of modulation techniques

In traditional automatic modulation classification, classifications were done from raw input signals like zero crossing locations, square law classifiers, and statistical moment classifiers. The traditional methods are likelihood based (LB), feature based (FB) and artificial neural network (ANN) based models. These models were used for specific modulation techniques and SNR levels. In recent advancements in the deep learning (DL), model architectures access to open-source software libraries like PyTorch, TensorFlow etc., graphics processing unit (GPUs) and tensor processing units (TPUs) have made DL to solve complex problems in better way. Hence the researchers in the wireless community apply deep learning in wireless communications [12], [18]–[21]. The recent works have applied deep learning for modulation classification. The deep learning deals with the convolutional neural networks (CNNs) is a process of applying matrix over the image which is used to obtain the features by providing the depth-wise seperable of each layer in an image due to its layering system each and every feature is identified easily to achieve better performance and accurate results [22], [23]. ResNet and convolutional long short-term memory networks (CLDNNs) recently introduced to improve the feature propagation in neural networks.

2. METHOD

2.1. Data preprocessing

The dataset contains IQ samples of 11 modulation classes (8 digital and 3 analog) over 20 SNR values ranging from -20 dB to 18 dB (RADIOML 2016.10A, RADIOML 2016.10B). The total 11 modulation classes are: 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM. In general the larger dataset doesn't have AM-SSB class. The smaller dataset and bigger dataset are stored in a dictionary with keys representing tuples of (modulation class, SNR value) and values providing the corresponding IQ samples. First step includes data preprocessing, arranging the dataset into a dictionary composed of a single key representing the entire dataset. All labels that are modulation classes are digitized i.e., a number was assigned to every individual class. Now the labels are transformed into a one-hot encoding vector [24]–[27].

2.2. Proposed Model

2.2.1 Robust CNN

The framework for creating the classifiers which are used to recognize or detect objects from the input image is the neural network. This architecture consists of input layer, hidden layers and output layer. The tensors are the nodes which has weights to perform feed forward and feed backward to get the transformations for each layer [28]–[33]. The activation function takes RELU and softmax for completely

connected layers as shown in Figure 2. Input layer takes the image as argument and apply convolutions over the entire image to extract the features by using the train data train data contains the features of Radio ML, the model will be trained and tested on the remaining dataset splitted for testing by test train split (70%-30%). Hidden Layers like convolution layers are used to scale the features on data, followed by pooling layers ensures to reduce the minima and the softmax layers are combining all the features that are used to predict and compare the actual values with the predicted values.

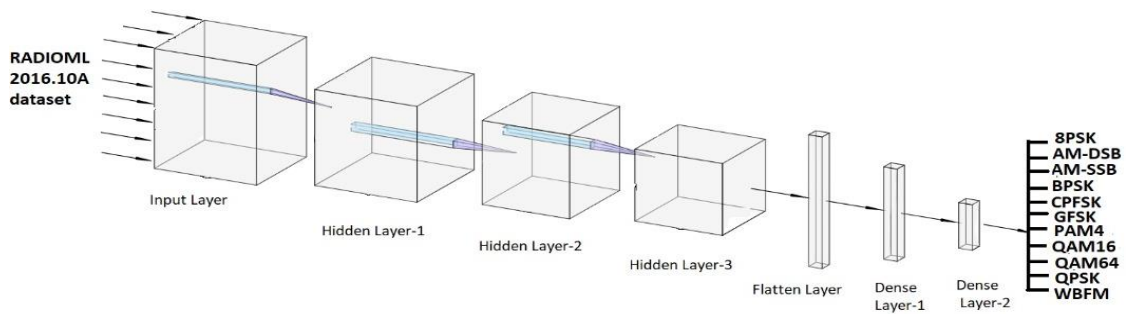


Figure 2. Architecture of robust CNN

The input layer is the input layer it is like a preprocessing layer it accepts the data and passes it to the next layers where each and every data component is connected to neural network. The hidden layer is the intermediate layer between the input and the output layers and it accepts the data and prepare connections on its weights through activation function with which the patterns will be identified and the generated model uses to recognize. The max pooling is purely focus on the brighter pixel of the input data and it is also used to downsample or dimensionality reduction by avoiding darker pixels. The Flatten layer is the output layer to create feature vector for classifying and detecting the input.

This paper shows the architecture and the neural network graph of a robust CNN which recognizes the different modulations. Here, the proposed model was trained with a hyper parameter tuning with dropout rate, learning rate, epochs, batch size, loss function and metrics. The tuning of parameters improves accuracy of the model. The performance was evaluated and the proposed model was trained [34]–[38].

Table 1. Robust CNN neural network model

Layer	Input shape	Output shape
Conv2D_4: input layer	2x128x1	2x128x1
Conv2D_4: hidden layer	2x128x1	2x128x256
Batch normalization_5	2x128x256	2x128x256
Max_pooling2D_4	2x128x256	2x64x256
Dropout_4	2x64x256	2x64x256
Conv2D_5: hidden layer	2x64x256	2x64x128
Batch normalization_6	2x64x128	2x64x128
Max_pooling2D_5	2x64x128	2x32x128
Dropout_5	2x32x128	2x32x128
Conv2D_6: hidden layer	2x32x128	2x32x64
Batch normalization_7	2x32x64	2x32x64
Max_pooling2D_6	2x32x64	2x16x64
Dropout_6	2x16x64	2x16x64
Conv2D_7: hidden layer	2x16x64	2x16x64
Batch normalization_8	2x16x64	2x16x64
Max_pooling2D_7	2x16x64	2x8x64
Dropout_7	2x8x64	2x8x64
Flatten_1	2x8x64	1024
Dense_2	1024	128
Dense_3	128	11

Here, the batch normalization layer was added after each convolution layer. The conv2d_4_input is the convolution layer whose input is the 2x128x1 which indicates the height, width and depth of the input image which is going to be recognized by RGB pallets, whose output is given to the conv2d_4 layer which is the first hidden layer. The hidden layer performs dimensionality reduction by using the layers of batch_

normalization, max_pooling2d_4, dropout_4 which gives the output as $2 \times 64 \times 256$ and then the conv2d_5, conv2d_6 and conv2d_7 with intermittent layers which gives $2 \times 8 \times 64$. The flatten_1 is the flatten layer which multiplies all the values to get 1024 and then the dense_2 is transformed into a shape 128 and the final layer gives 11 as the objects which we are recognizing (8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM). The Neural Network model is shown in Table 1.

2.3. Training

The training procedure contains two steps, in the first step the data is divided into three separate sets. We assigned 80% of the data for training the dataset and the other 20% percent assigned for testing the dataset. From the training dataset, 10% was extracted for validating the dataset and to evaluate the performance of the proposed model at every epoch for overfitting verification during the training process. The trained and test processed data were used to evaluate the proposed model performance in terms of accuracy [15]. After this process two parameters are considered from the training procedure: they are the loss function and the optimizer. In this work, we assured to consider the categorical cross-entropy as loss function and the ADAM optimizer to estimate the parameters of the proposed model with a learning rate of 0.0001. The model was trained for 100 epochs with a batch size of 128. We used Keras Tensor flow for training the model and the SNR values (-20 dB to 18 dB) are stored in the pickle files.

3. RESULTS AND DISCUSSION

In the robust CNN model, the first step was to train the proposed model with a dropout rate 0.3. In general, higher dropout rates reduce the model complexity by neglecting parameters. The training process took about (~1.50 mins) with the rate of 0.3. The second step was batch normalization layer which is added after each convolution layer.

Figures 3 and 4 shows model accuracy by Epoch for SNR 0 dB, -6 dB, 8 dB & 14 dB. From this we got maximum classification accuracy of 89.57%. The proposed model was tested with signals having high interference. Some of the signals had noise and those weren't recognized. Figure represents the validation loss of the Robust CNN model. Figures 5 and 6 represents the Train loss by Epoch for SNR 0 dB, -6 dB, 8 dB and 14 dB which give validation loss after 100 epochs.

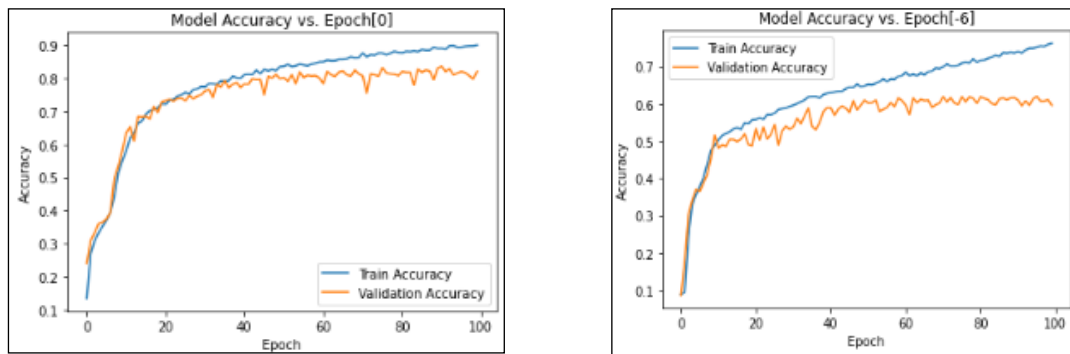


Figure 3. Model Accuracy by epoch for SNR 0 dB and -6 dB

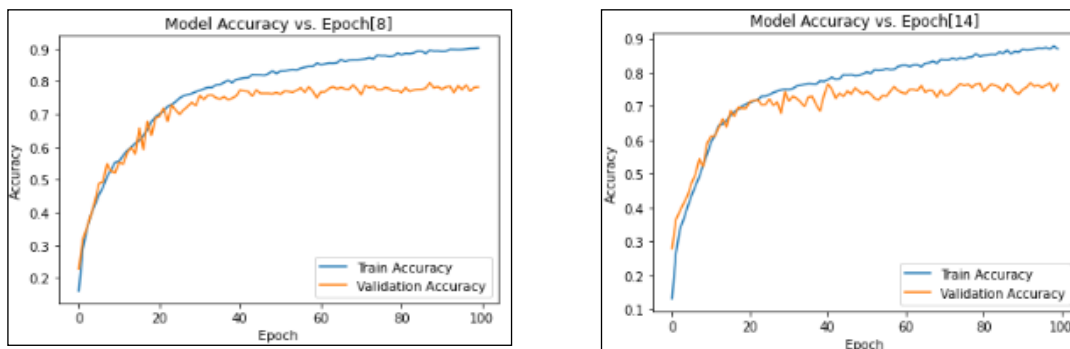


Figure 4. Model accuracy by epoch for SNR 8 dB and 14 dB

The training dataset loss is less than the validation loss it leads to overfitting of data. So the trained dataset is very good and whereas the validation data consists of unseen data. The dataset with all the SNR values are comes under the overfitting where the train data is working fine. The dropout layer is used to prevent the model from overfitting from input to output.

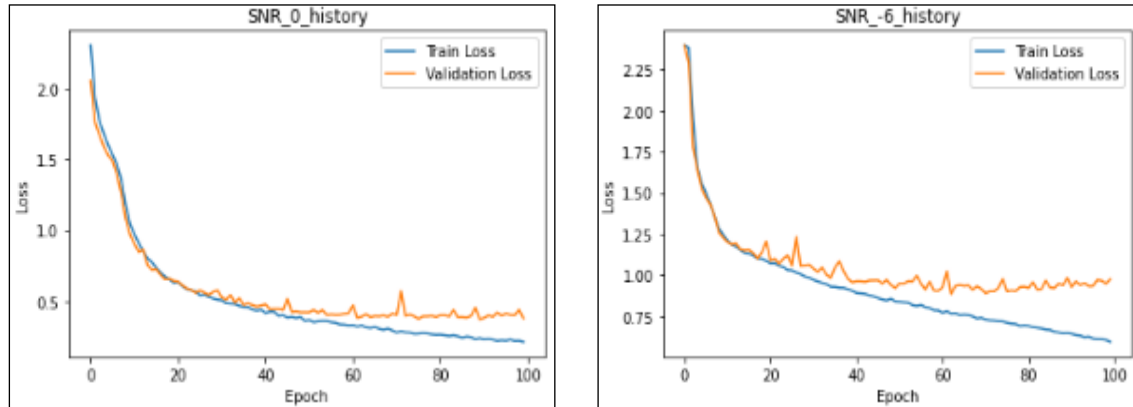


Figure 5. Train loss by epoch for SNR 0 dB and -6 dB

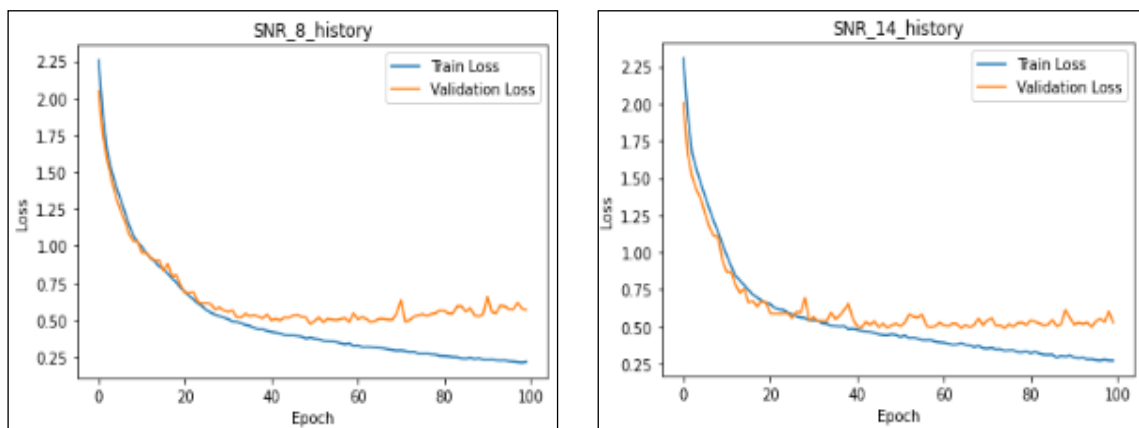


Figure 6. Train loss by epoch for SNR 8 dB and 14 dB

Figures 7 and 8 represents the confusion matrices for different SNR. A confusion matrix predicts how accurately a class aligns with its true class in the form of a probability score. For robust CNN there are 11 modulation classes performs the best and gives accurate results. Some modulation classes are misclassified because of their similar constellation diagrams. The proposed model was tested using the test datasets and the accuracy of the proposed model was 89.57% overall, which was calculated using the following equation. The classification is the major aspect mainly effects the accuracy of the model. The classification errors are common while training the model with dataset. Support vector machine (SVM) is the technique to address the class imbalanced over the dataset. The imbalance is the target class have the uneven distribution of data. The classification of modulation technique QPSK will be recognized as BPSK and WB-FM will be recognized as AM-DSB. The imbalance of data is biased to BPSK and WB-FM. Accuracy = $(TP+TN)/(TP+TN+FP+FN)$ where true positive (TP), true negative (TN), false positive (FP), false negative (FN). The confusion matrix of the network will identify all the signals with SNR 0 and only 10% of signal being unrecognized during learning stage. So the hyper parameter tuning will improve it by at least 5% of 89.7%.

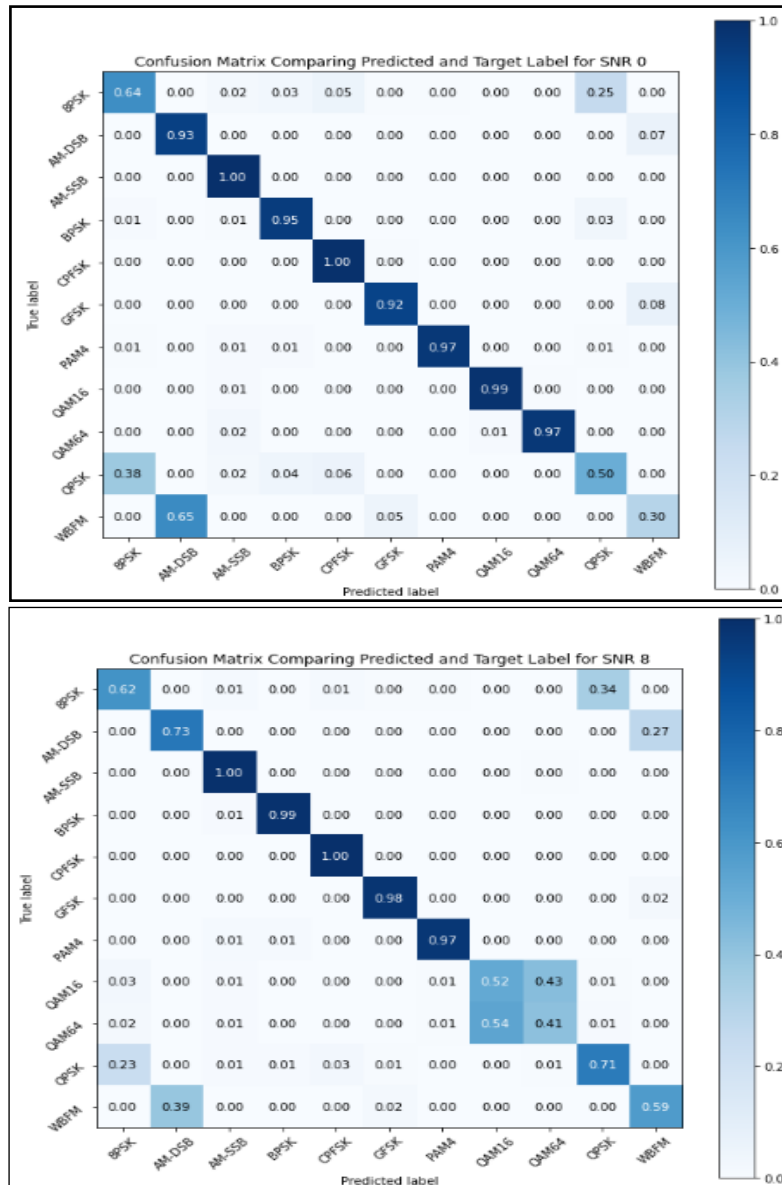


Figure 7. Confusion matrices for SNR 0 dB and 8 dB

Table 2 describes the model accuracy at different SNR (-20 dB to 18 dB). The positive SNR values ranges from 89.57 to 84.03 so the maximum accurate results at SNR-0 along with the 5% that can be improved by using hyper parameter tuning leads to 94% and for the other SNR-2 to SNR-18 improved the accurate results to 90%. The hyper parameter tuning will improve accuracy through which the network will be identify all the signal modulation classes with higher accuracy.

Table 2. Accuracy at different SNR (-20 dB to 18 dB)

SNR (dB)	Accuracy (%)	SNR (dB)	Accuracy (%)
-20	8.64	0	89.57
-18	8.91	2	88.87
-16	27.68	4	88.23
-14	36.18	6	87.5
-12	47.27	8	87.1
-10	62.64	10	85.27
-8	73.5	12	85.04
-6	80.5	14	84.49
-4	82.05	16	84.25
-2	86.47	18	84.03

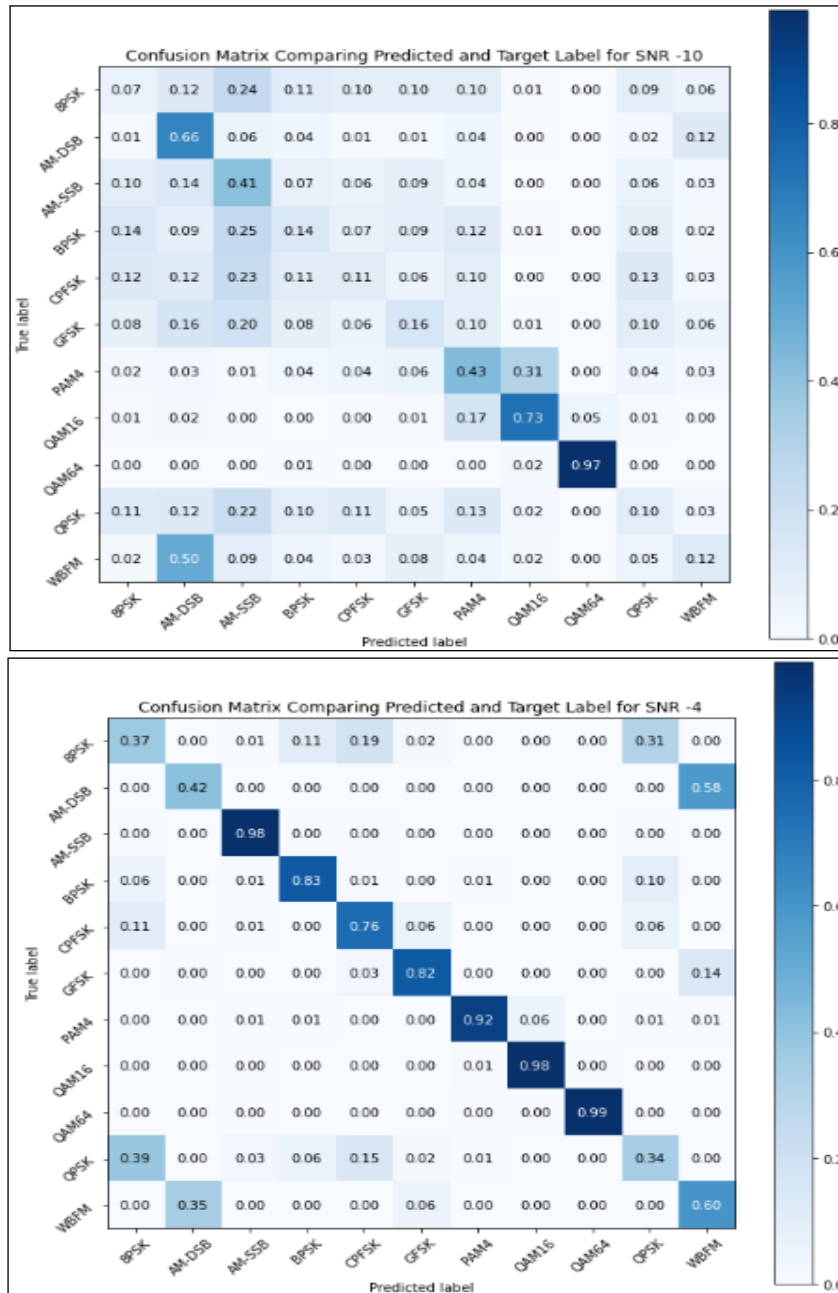


Figure 8. Confusion matrices for SNR -10 dB and -4 dB

4. CONCLUSION AND FUTURE WORK

The real time applications in military, autonomous vehicles and robotic automation processes are increasing day by day using machine learning. The CNN plays an important role in identifying and classifying the tasks in an advanced and easier way. This paper stresses on recognizing the signal modulation. This is then predicted with the actual values and the generated model with robust convolution neural network performs the accuracy. The layers are constructed in such a way to reduce the dimensions of the patterns generated at each convolutional, flatten and dense layers. The training, testing and validation of the model is done to get the better accuracy for each SNR and minimal loss with local minima and global maxima. The future work will focus on feature extraction and deep learning-based ensemble classifier for Interference Mitigation in radar signals. The signal recognition performance of compound signals by effective learning architecture and optimization method.





REFERENCES

- [1] C. Toolbox, D. L. Toolbox, P. C. Toolbox, and C. T. Support, "Modulation classification with deep learning," *MathWorks*, 2020. <https://www.mathworks.com/help/deeplearning/ug/modulation-classification-with-deep-learning.html> (accessed Jun. 12, 2019).
- [2] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Top. Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018, doi: 10.1109/JSTSP.2018.2797022.
- [3] X. Liu, D. Yang, and A. El Gamal, "Deep neural network architectures for modulation classification," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, Oct. 2017, pp. 915–919. doi: 10.1109/ACSSC.2017.8335483.
- [4] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *Proceedings of the 6th GNU Radio Conference*, 2016, p. 1. [Online]. Available: <https://pubs.gnuradio.org/index.php/grcon/article/view/11>
- [5] E. E. Azzouz and A. K. Nandi, "Modulation recognition using artificial neural networks," in *Automatic Modulation Recognition of Communication Signals*, Boston, MA: Springer US, 1996, pp. 132–176. doi: 10.1007/978-1-4757-2469-1_5.
- [6] K. Karra, S. Kuzdeba, and J. Petersen, "Modulation recognition using hierarchical deep neural networks," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, Mar. 2017, pp. 1–3. doi: 10.1109/DySPAN.2017.7920746.
- [7] C. Wang, J. Wang, and X. Zhang, "Automatic radar waveform recognition based on time-frequency analysis and convolutional neural network," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2437–2441. doi: 10.1109/ICASSP.2017.7952594.
- [8] T. J. O'Shea and J. Hoydis, "An introduction to machine learning communications systems," *Mach. Learn. Methods Ecol. Appl.*, pp. 1–37, 2017, doi: 10.1007/978-3-319-20010-1.
- [9] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," Nov. 2015, [Online]. Available: <http://arxiv.org/abs/1511.07289>
- [10] Z. Wu, S. Zhou, Z. Yin, B. Ma, and Z. Yang, "Robust automatic modulation classification under varying noise conditions," *IEEE Access*, vol. 5, pp. 19733–19741, 2017, doi: 10.1109/ACCESS.2017.2746140.
- [11] "Constellation diagrams: the fault in our stars," *NuWaves engineering*. <https://www.nuwaves.com/constellation-diagrams> (accessed Jun. 12, 2019).
- [12] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456. [Online]. Available: <http://proceedings.mlr.press/v37/ioffe15.html>
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.
- [14] "RF datasets for machine learning," *DEPSIG*. <https://www.deepsig.ai/datasets>
- [15] D. Zhang *et al.*, "Automatic modulation classification based on deep learning for unmanned aerial vehicles," *Sensors*, vol. 18, no. 3, p. 924, Mar. 2018, doi: 10.3390/s18030924.
- [16] The Theano Development Team, "Theano: a python framework for fast computation of mathematical expressions," May 2016, [Online]. Available: <http://arxiv.org/abs/1605.02688>
- [17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [18] X. Zhu and T. Fujii, "Modulation classification for cognitive radios using stacked denoising autoencoders," *Int. J. Satell. Commun. Netw.*, vol. 35, no. 5, pp. 517–531, Sep. 2017, doi: 10.1002/sat.1202.
- [19] J. Mu, Q. Liang, W. Wang, B. Zhang, and Y. Pi, *The proceedings of the third international conference on communications, signal processing, and systems*, vol. 322. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-08991-1.
- [20] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," 2016, pp. 213–226. doi: 10.1007/978-3-319-44188-7_16.
- [21] A. Dai, H. Zhang, and H. Sun, "Automatic modulation classification using stacked sparse auto-encoders," in *2016 IEEE 13th International Conference on Signal Processing (ICSP)*, Nov. 2016, pp. 248–252. doi: 10.1109/ICSP.2016.7877834.
- [22] G. J. Mendis, J. Wei, and A. Madanayake, "Deep learning-based automated modulation classification for cognitive radio," in *2016 IEEE International Conference on Communication Systems (ICCS)*, Dec. 2016, pp. 1–6. doi: 10.1109/ICCS.2016.7833571.
- [23] M. Abdullah *et al.*, "Future smartphone: MIMO antenna system for 5G mobile terminals," *IEEE Access*, vol. 9, pp. 91593–91603, 2021, doi: 10.1109/ACCESS.2021.3091304.
- [24] P. M. Rodriguez, Z. Fernandez, R. Torrego, A. Lizeaga, M. Mendicutte, and I. Val, "Low-complexity cyclostationary-based modulation classifying algorithm," *AEU - Int. J. Electron. Commun.*, vol. 74, pp. 176–182, Apr. 2017, doi: 10.1016/j.aeue.2017.02.008.
- [25] 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 Networks*, vol. 93, p. 101913, Oct. 2019, doi: 10.1016/j.adhoc.2019.101913.
- [26] S. Peng *et al.*, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019, doi: 10.1109/TNNLS.2018.2850703.
- [27] J. Jagannath *et al.*, "Artificial neural network based automatic modulation classification over a software defined radio testbed," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6. doi: 10.1109/ICC.2018.8422346.
- [28] R. Li, L. Li, S. Yang, and S. Li, "Robust automated VHF modulation recognition based on deep convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 946–949, May 2018, doi: 10.1109/LCOMM.2018.2809732.
- [29] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, "Neural networks for signal intelligence," in *Machine Learning for Future Wireless Communications*, Wiley, 2020, pp. 243–264. doi: 10.1002/9781119562306.ch13.
- [30] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: a survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020, doi: 10.1109/ACCESS.2020.2986330.
- [31] X. Li, F. Dong, S. Zhang, and W. Guo, "A survey on deep learning techniques in wireless signal recognition," *Wirel. Commun. Mob. Comput.*, vol. 2019, pp. 1–12, Feb. 2019, doi: 10.1155/2019/5629572.
- [32] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-based automatic modulation classification for beyond 5G communications," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1038–1041, May 2020, doi: 10.1109/LCOMM.2020.2970922.
- [33] O. S. Mossad, M. ElNainay, and M. Torki, "Deep convolutional neural network with multi-task learning scheme for modulations recognition," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, Jun. 2019, pp. 1644–1649. doi: 10.1109/IWCMC.2019.8766665.
- [34] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. F. Robot.*, vol. 37, no. 3, pp. 362–386, Apr. 2020, doi: 10.1002/rob.21918.





- [35] J. Jagannath, A. Jagannath, S. Furman, and T. Gwin, "Deep learning and reinforcement learning for autonomous unmanned aerial systems: roadmap for theory to deployment," Sep. 2020, [Online]. Available: <http://arxiv.org/abs/2009.03349>
- [36] A. Jagannath and J. Jagannath, "Communication and radar dataset for modulation and signal classification," *ANDROComputationalSolutions/RadarCommDataset*. <https://github.com/ANDROComputationalSolutions/RadarCommDataset>
- [37] M. L. Bouknia *et al.*, "Analysis of the combinatory effect of uniaxial electrical and magnetic anisotropy on the input impedance and mutual coupling of a printed dipole antenna," *IEEE Access*, vol. 9, pp. 84910–84921, 2021, doi: 10.1109/ACCESS.2021.3085949.
- [38] A. A. Althuwayb, M. Alibakhshikenari, B. S. Virdee, F. Falcone, and E. Limiti, "Overcoming inherent narrow bandwidth and low radiation properties of electrically small antennas by using an active interior-matching circuit," *IEEE Access*, vol. 9, pp. 20622–20628, 2021, doi: 10.1109/ACCESS.2021.3054939.

BIOGRAPHIES OF AUTHORS



Nadakuditi Durga Indira     is pursuing Ph.D. degree in K. L. University, Vijayawada in the Faculty of Electronics and Communication Engineering, specialized in Communications. She received her Master of Engineering Degree in Communication and Radar Systems from K. L. University, Vaddeswaram. Her present affiliation is with K L University since 2012, Vijayawada designated as Assistant professor in ECE. She is a fellow of IETE and life member in IAENG, ISTE. Her research interest includes signal processing, compressive sensing, Communications. She has published papers in various international and national journals, conferences and workshops. She can be contacted at email: durga.indira@gmail.com.



Dr. Matcha Venu Gopala Rao     obtained his AMIETE degree from Institute of Electronics and Telecommunication Engineers, New Delhi, India during 1996. He obtained his M. Tech. from Regional Engineering College, Warangal, and Doctorate from Osmania University, Hyderabad. He possesses 17 years of teaching experience and 21 years of industrial experience. Presently he is working as Professor in the department of ECE and Associate Dean (Quality-Audits), K. L. University, Vaddeswaram, Guntur Dt, A. P., India. He received a 'credit award' in Radio Servicing Theory from City and Guild's London Institute, London in 1978. He is a fellow of IETE and life member in IAE, ISTE, SSI, and ISOI. His research interest includes signal, image processing, GPS compressive sensing, Sparse and Dictionary learning. He has published more than 50 papers in various international and national journals, conferences and workshops. He can be contacted at email: mvgr03@kluniversity.in.