LAUTECH Journal of Engineering and Technology 19 (2) 2025: 106-113
10.36108/laujet/5202.91.0211

Performance Analysis of Deep Learning-Based Automatic Modulation Recognition over Wireless Communication

[1]Ismail Y., [2]Abolade R. O., [3]Oyelaran O. and [4]Fakokunde B.

[1,2,3,4]Department of Electronic and Electrical Engineering, Ladoke Akintola University of Technology, P.M.B. 4000, Ogbomoso, Oyo State, Nigeria.

[1]ardeyemi95@gmail.com, [2]roabolade@lautech.edu.ng, [3]oyelaranope@gmail.com
[4]babatunde.fakokunde@gmail.com

Article Info

Article history:

Received: Jan 27, 2025
Revised: Mar 11, 2025
Accepted: Mar 13, 2025

Keywords:

Modulation,
Model,
Optimizer,
Dataset,
and Spectrum

Corresponding Author:
roabolade@lautech.edu.ng
Tel.: +2348035394269

ABSTRACT

Automatic Modulation Recognition (AMR) based on Deep Learning (DL) is an efficient technique to improve spectrum utilization by replacing the old way of detecting modulation type through the allocation of modulation information in the signal frame. However, DL models have the problem of low recognition accuracy when dealing with a dataset containing in-phase and quadrature channel data. Hence, in this work, the enhancement of DL models that automatically recognize different types of modulation techniques with an increase in recognition accuracy was carried out. The two utilized dataset were RadioML2016.10a and RadioML.2016.10b. Convolutional Neural Network with RadioML2016.10a (ECNN-1) and RadioML2016.10b (ECNN-2) and Long Short-Term Memory with RadioML2016.10a (ELSTM-1) and RadioML2016.10b (ELSTM-2) were implemented in Python 3 using Google Colab. Adam optimizer was applied to optimize the hyperparameters of DL models. ECNN-1 and ECNN-2 have recognition accuracy values of 81% and 88%. The accuracy values obtained for ELSTM-1 and ELSTM-2 were 79% and 85%. The ROC AUC score for the ECNN-1, ECNN-2, ELSTM-1, and ELSTM-2 were 89.63%, 92.90%, 90.92%, and 92.81%, respectively. The experimental results showed an improvement in modulation recognition accuracy for both enhanced CNN and LSTM models.

INTRODUCTION

Automatic Modulation Recognition (AMR) is a technique for determining the type of modulation of the received radio signals. Modulation wireless communication systems have become increasingly complicated by the techniques of anti-multipath fading, frequency selection, and time-varying channels, to mention a few. Accordingly, a more accurate and robust signal modulation recognition method is required to counteract the upcoming harsh environments (Kim *et al.*, 2022). Similarly, modulation is carried out by including modulation information in each signal frame for the receiver to recognize the modulation type involved. The spectrum spaces occupied by the extra information lead to spectrum wastage and overhead in the network protocol. To improve the spectrum utilization efficiency in wireless communication, Automatic Modulation Recognition (AMR) was introduced to the system to identify and classify various types of modulation with unknown signals from heterogeneous devices. Recognition of the modulation scheme is the intermediate step between signal detection and demodulation of the received signal in communication networks. Automatic modulation recognition plays a central role in many applications, especially in the military and security sectors (Ansari *et al.,* 2022). AMR methods can be classified into two categories: the Feature Based (FB) and Likelihood Based (LB) methods (Njoku *et al.,* 2021). The likelihood-based classification method can theoretically obtain the optimal classification performance, but it requires substantial prior knowledge and a considerable amount of computation. However, the feature-based method consists of feature extraction and classifier construction (Fu *et al*., 2022). The former entails high computational complexity and suffers from the ambiguity of the parameters. In the second approach, certain features of the signal are extracted first, then decisions are made based on the characteristics. The latter is less complex compared to the first approach. Therefore, it is more convenient to implement the feature-based approach in a practical system (Ansari *et al*., 2022). Compared with the featured-based method, the likelihood-based method treats both noises and channel models which reflect the propagation characteristic of signals as prior information (Ma *et al*., 2020).

Even though the Feature-Based (FB) method is based on the experience and knowledge of signal feature engineering, it is not suitable in a non-cooperative environment where any sensitive signal information such as frequency deviation, transmitting power, and modulation type cannot be easily obtained. FB methods pose a drawback of generating features that are not representative enough. This makes the machine learning classifier not discriminative enough when dealing with challenging modulation types (Njoku *et al*., 2021). On the contrary, deep learning models have been explored extensively due to feature extraction, and modulation classification capabilities. However, sourcing a large dataset of signals to train the models and using a suitable optimizer that can help to improve models' performance are amongst the major challenges DL models are facing.

In this work, Adam optimizer was used to enhance Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models for recognition and classification of different modulation techniques over wireless communication channels. Hyperparameters of the deep learning models were tuned and the performance of the models was evaluated using the following evaluation metrics; accuracy, f1-score, and Receive Operating Characteristic (ROC)/Area Under Curve (AUC) Score.

Numerous researches have focused on DL-AMR (Deep Learning –Automatic Modulation Recognition) models in terms of classification criteria and employed features. Ramjee *et al* (2019) achieved accuracies of 87.3 % and 81.4% for CNN and ResNet respectively on RadioML2016.10b at 18dB. The model was trained with a large dataset but hyperparameter tuning was not applied. Emam *et al* (2020) employed RadioML.2016.10b to identify the modulation types of the signal by using a combination of CNN, LSTM, and DNN architecture, which is called CLDNN (Convolutional Long-Short-Term-Deep Neural Network). The model included three convolutional CNN layers then followed by a single LSTM layer with 50 computing units and two fully connected DNN layers. The model was compared with the individual models of CNN and LSTM, and a 2-3% relative improvement in accuracy was achieved on the test data set with signal-to-noise ratio (SNR) varying from -18dB to 20 dB. The model accuracy is low at the lower signal-to-noise ratio (SNR).

The study is limited to one dataset. Aminfar *et al* (2022) worked on intelligent signal modulation recognition whereby k-nearest Neighbor, SVM, Decision Tree, and Random Forest were employed to identify modulation type based on SNR. The dataset was made up of radio waves of various waveforms. It was generated synthetically using the

AWGN channel model with Watterson fading (for ionospheric propagation) and random frequency and phase offset. However, the model is not robust, only two types of modulation techniques (QPSK and PSK) were considered.

Mohsen *et al* (2023) also worked on modulation recognition by employing the CNN model on RadioML2016.10a., the highest accuracy achieved is 50% at 11dB. However, this model was not trained with sufficient dataset. Referring to the reviewed works, there is still room for significant improvement in getting accurate models for modulation classification techniques and recognition through the impact of hyperparameters tuning on the modified models to boost the accuracy.

RESEARCH METHODOLOGY

In this work, recognition and classification of the modulation type of received radio signals were considered by employing deep learning models that will adaptively incorporate features extracted from the dataset. Source signals that were considered in this research were open source deepsig dataset generated by Gnu's Not Unix (GNU) radio which are RadioML2016.10a and RadioML2016.10b. The GNU radio library has many in built tools such as modulators, encoders, and demodulators.

Signal Model

The received baseband complex envelope signal for the wireless system can expressed as:

$$r(t) = s(t; \boldsymbol{u_i}) + n(t) \tag{1}$$

Where $r(t)$ is the received signal, $s(t; \boldsymbol{u_i})$ is the noise-free baseband complex envelope of the received signal, $\boldsymbol{u_i}$ is the multidimensional vector that includes the deterministic unknown channel parameters of modulation type i , and $n(t)$ is the instantaneous channel noise at time t.

Data Acquisition

The datasets that were employed as the input data in this work are RadioML2016.10a and RadioML2016.10b. These datasets were packaged data generated by the GNU Radio model which was stored as an N-dimensional vector using numpy and cPickle. This makes them useful for training, validating, and testing various deep learning-automatic modulation recognition models.

i. RadioML2016.10a: This dataset was generated with Gnu's Not Unix (GNU) radio and can be found in the Kaggle repository. It includes eight digital and three analog modulation techniques which are: 8PSK, BPSK, CPFSK, GFSK, QAM16, QAM64, QPSK, WBFM, PAM4, AM-DSB, AM-SSB. The dataset has 220,000 samples of modulation techniques with twenty different signal-to-noise ratio (SNR) values from -20dB to 18 dB. There are four attributes with the dataset: multipath fading, additive white Gaussian Noise, sample rate offset, and center frequency offset.

ii. RadioML2016.10b: This is a larger version of the RadioML2016.10a dataset and has 1.2 million samples. It includes 10 different types of modulation techniques that are used for radio signals, eight of which are digital modulations: QPSK, QAM16, QAM64, CPFSK, BFSK, BPSK, 8PSK, and PAM4, and two analog modulations AM-DSB and WBFM. Each sample in the dataset consists of 128 sampling points where the real part and imaginary part represent I and Q signals respectively and each modulation type has 20 different SNRs ranging from -20dB to 18dB.

Enhancement of Deep Learning (DL) Models for Automatic Modulation Recognition (AMR)

In this work, two different architectures of deep learning models were enhanced by applying a dropout rate of 0.6 to the network layers, tuning epochs to 50 and 60, and applying the ReLU activation function. Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) were trained and tested on RadioML2016.10a and RadioML2016.10b datasets.

Considering Figure 1, the architecture of the enhanced CNN model that was applied to the RadioML2016.10a and RadioML2016.10b datasets contained three CNN layers, three dropout layers, a flattened layer, two dense layers, and an activation layer. The first convolutional layer was applied with the ReLU function and contained the input shape. Secondly, another convolutional layer was added to enable the model to detect high-level features that might have been missed in the first layer and was followed by a dropout rate of 0.6. Another convolutional layer was added to extract more features of the data. This layer was then applied with a ReLU function which was followed by a dropout rate of 0.6. The flattened layer was then used to convert the data into a single dimension followed by a dense layer with a dropout rate of 0.6. Finally, a dense layer was applied to the data with a softmax activation function.

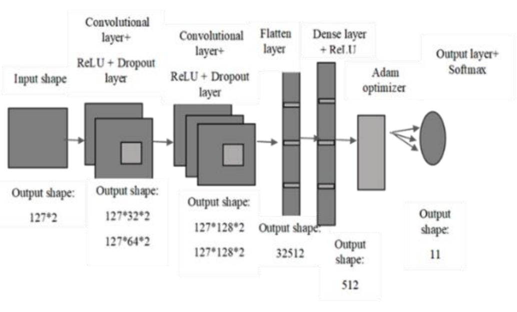

Figure 1: The Architecture of Enhanced CNN Model

Moreover, an Adam optimizer was employed to tune the hyperparameters of the model. Equation (2) stands for the transfer function of the CNN layer.

$$y_i = f\left(b_{j+\sum_i K_{ij} * x_i}\right) \tag{2}$$

Where y_i output feature map, i is the filter number, b_j is learned bias, K_{ij} is kernel, X_i is input, (*) stands for convolution operation and f stands for a nonlinear activation function i.e. ReLU.

Considering Figure 2, the developed architecture of the LSTM model which was applied to the RadioML2016.10a and RadioML2016.10b datasets comprises of 5 layers which are: two LSTM layers, 2 dropout layers, and a dense layer. The first LSTM layer had an input function as an argument with a dropout rate of 0.6. Then followed by a second LSTM layer with a dropout rate of 0.6. Finally, a dense layer was applied with a softmax activation function. Moreover, an Adam optimizer was employed to tune the hyperparameters of the model.

Hyperparameters Tuning of the Enhanced Models

The performance of deep learning models was improved through the tuning of DL models' learning rate, epochs, and ReLU function by applying the Adaptive Moment Estimation (Adam) optimization algorithm. Considering the Equation (3) of the neural network function of the models:

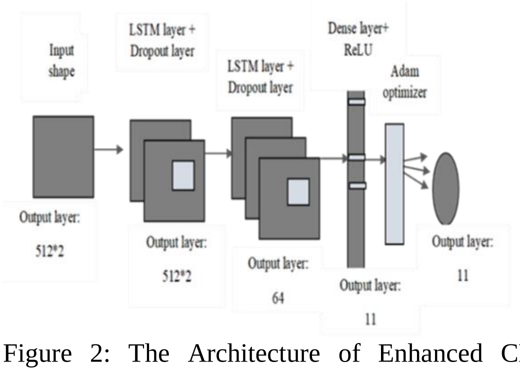

Figure 2: The Architecture of Enhanced CNN Model

$$N(x) = g(Wx + b) \tag{3}$$

Where $N(x)$ stands for neural network function, g represents the activation function, W and b are kernel and bias respectively, x stands for input. From Equation (3), the weights W and b contained the information learned by CNN and LSTM models from the exposure to training data. Equations for Adam optimizer are:

$$m_i = \beta_1 m_i + (1 - \beta_1)\nabla\theta \quad (4)$$

$$v_i = \beta_2 v_i + (1 - \beta_2)\nabla^2\theta \quad (5)$$

Where m_i is the moving average of the gradient, v_i is the moving average of the variance, β_1 and β_2 represent Beta 1 and Beta 2 respectively, $\nabla\theta$ stands for gradient function.

$$w_{t+1} = w_t - \frac{\lambda m_t}{\sqrt{s_t} + \epsilon} \quad (6)$$

Where w_{t+1} represents the parameters after the update, w_t represents the parameters before the update, represents the learning rate, m_t represent the bias-corrected first-moment estimate of gradient, v_t represents the bias-corrected second-moment estimate of gradients, ϵ stands for epsilon which is a small-scale factor to maintains numerical stability

Since the models were implemented using Python 3, the Adam algorithm from Keras (deep learning framework for Python) was employed. The optimizer was instantiated with fine-tuning of the learning rate to 0.001, beta_1 to 0.99, beta_2 to 0.999, and epsilon to $1e-08$ to achieve better performance before passing it as an argument during the model compilation.

Implementation of the Enhanced Models

Considering Figure 3, the enhanced deep learning models' code was written in Python 3 using Google Colab as a cloud computing platform. Firstly, the drive where the dataset was stored was mounted since it is a large dataset. Secondly, all the used libraries were imported and this was followed by the loading of the dataset which was stored in pickle format. The third step was the pre-processing of the uploaded dataset followed by splitting of dataset into 70% training set and 30% of the test set. Then, the CNN model and LSTM models were defined with the number of epochs followed by compilation. Then the next step was the training on the dataset via fitting to train X and train Y. To monitor the accuracy of the model on data during training, 30% of the training set was set apart as a validation set. The loss and accuracy of these data were considered. Then, the models were evaluated on test data. Finally, performance evaluation was carried out on the training dataset and testing dataset using accuracy, and ROC AUC score.

RESULT AND DISCUSSION

The developed DL models were experimentally executed using the Keras framework on cloud computing software: Google Colab with T4 Graphical Processing Unit (GPU). Python programming language is used for the implementation of CNN and LSTM models. The Enhanced CNN and LSTM models trained and tested on RadioML2016.10a were represented as ECNN-1 and ELSTM-1 respectively while the ones tested and trained on RadioML2016.10b were named ECNN-2 and ELSTM-2 respectively.

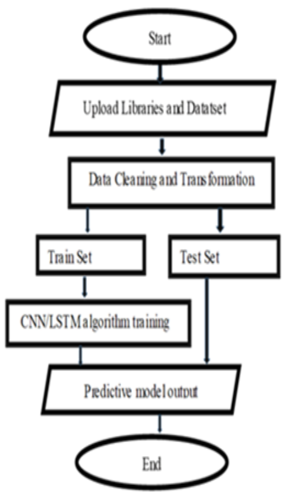

Figure 3: The Software Diagram of Enhanced Deep Learning Model.

Tuning of Adam hyperparameters was carried out on training sets that were used to train the models. Figure 4 shows the values obtained before using the Adam optimizer and after application of it to the models. Adam optimizer produced better results because using Adam, a separate learning rate is maintained for each parameter weight and updated individually. Validation and training accuracy of the first CNN model. CNN-1, CNN-2, LSTM-1 and LSTM-2 have the training accuracy of 48.72%, 51%, 41% and 55% respectively. The accuracy improved of ECNN-1, ECNN-2, ELSTM-1, and ELSTM-2 to 53%, 59%, 52%, and 57% respectively after tuning of Adam optimizer.

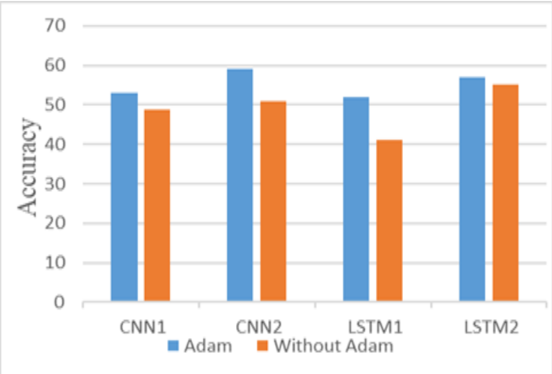

Figure 4: Effect of Adam Optimizer on Models' Accuracy

As illustrated in Figure 6, SNR values for each modulation for the enhanced LSTM model range from -20dB to 18 dB. The achieved highest accuracy for ELSTM-1 is 79% at 10dB and 18 dB while for ELSTM-2 is 85% at 12dB and 18 dB. Also, ECNN-1 and ECNN-2 have the ROC/AUC score of 89.63% and 92.90% respectively while ECNN-1 and ECNN-2 have ROC/AOC scores of 90.92% and 92.81% respectively.

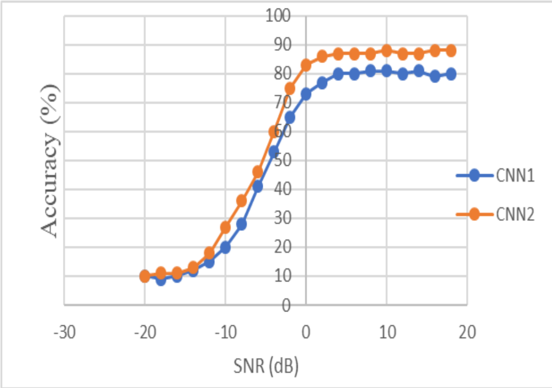

Figure 5: Accuracy Versus SNR Values for the ECNN Models

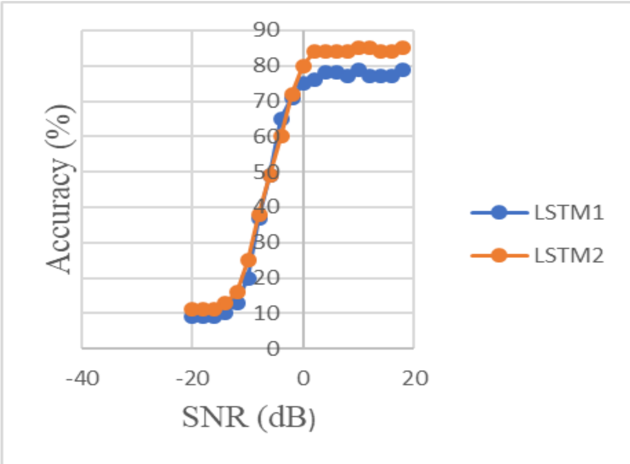

Figure 6: Accuracy Versus SNR Values for the ELSTM Models

Considering Table 1, two enhanced models employed in this research were trained with both RadioML2016.10a and RadioML2016.10b with the application of hyperparameter tuning. The ECNN-2 model has a higher recognition accuracy using RadioML2016.10b at 18dB and ECNN-1 has a higher recognition accuracy using RadioML2016.10b at 14dB when compared with the previous works. In particular, the enhanced ECNN-1, ECNN-2, ELSTM-1, and ELSTM-2 models give 81%, 88%, 79%, and 85% respectively.

CONCLUSION

In this work, CNN and LSTM architectures are presented for recognition of modulation types with high classification accuracy over wide range of SNR. These modulation techniques are labeled based on RadioML2016.10b dataset into ten classes and RadioML2016.10a into eleven classes. The experimental results show the effect of enhancing the models through the application of the Adam optimizer. Significant increment was achieved in models' accuracies after hyperparameters tuning of learning rate, epochs and ReLU activation function. After the implementation of the enhanced models on google colab due to free access to Graphics Processing Unit (GPU) and Tensor Processing Unit

Table 1: Accuracy Comparison Between the Developed Work and Previous Works

Reference	Model	SNR(dB)	Dataset	Accuracy(%)
Ramjee *et al.*, (2019)	CNN	18	RadioML2016.10b	87.3
	ResNet	18	RadioML2016.10b	81.4
Mohsen *et al.*, (2023)	CNN	14	RadioML2016.10a	50
Developed work	ECNN-1	14	RadioML2016.10a	81
	ELSTM-1	10	RadioML2016.10a	79
	ELSTM-2	18	RadioML2016.10b	85
	ECNN-2	18	RadioML.2016.10b	88

(TPU), the enhanced models were able to achieved better performance compared with the previous works the ROC AUC score for ECNN-1, ELSTM-1, and ELSTM-2 are 89.63%, 90.92%, and 92.81%. Since the ROC AUC scores for the proposed models are all above 90% except the CNN1 score, hence, they are all considered to be great models. The enhanced models are useful in spectrum sensing, satellite communication, and signal surveillance.

REFERENCE

Abd, M. H. M., & Aminifar, S. (2022). Intelligent digital signal modulation recognition using machine learning. Journal of Computer Science, 18(10), 896–903.

Ansari, S., Alnajjar, K. A., Saad, M., Abdallah, S., & El-Moursy, A. A. (2022). Automatic digital modulation recognition based on genetic algorithm-optimized machine learning models. IEEE Access, 10, 50265–50277.

Emam, A., Shalaby, M., Aboelazm, M. A., Bakr, H. E. A., & Mansour, H. A. A. (2020). A comparative study between CNN, LSTM, and CLDNN models in the context of radio modulation classification. In 2020 12th International Conference on Electrical Engineering, ICEENG 2020 (pp. 190–195).

Fu, X., Gui, G., Wang, Y., Gacanin, H., & Adachi, F. (2022). Automatic modulation classification based on decentralized learning and ensemble learning. IEEE Transactions on Vehicular Technology, 71(7), 7942–7946.

Heydarian, M., Doyle, T. E., & Samavi, R. (2022). MLCM: Multi-label confusion matrix. IEEE Access, 10, 19083–19095.

Jdid, B., Hassan, K., Dayoub, I., Lim, W. H., & Mokayef, M. (2021). Machine learning based automatic modulation recognition for wireless communications: A comprehensive survey. IEEE Access, 9, 57851–57873.

Kim, S. H., Moon, C. B., Kim, J. W., & Kim, D. S. (2022). A hybrid deep learning model for automatic modulation classification. IEEE Wireless Communications Letters, 11(2), 313–317.

Liu, F., Zhang, Z., & Zhou, R. (2022). Automatic modulation recognition based on CNN and GRU. Tsinghua Science and Technology, 18.

Ma, H., Xu, G., Meng, H., Wang, M., Yang, S., Wu, R., & Wang, W. (2020). Cross-model deep learning scheme for automatic modulation classification. IEEE Access, 8, 78923–78931.

Mohsen, S., Ali, A. M., & Emam, A. (2023). Automatic modulation recognition using CNN deep learning models. Multimedia Tools and Applications.

Ndujiuba, C., Omotayo, O. O., Ndujiuba, C. U., Oni, O., & Ibhaze, A. E. (2015). Comparative analysis of digital modulation techniques in LTE 4G systems. Journal of Wireless Networking and Communications, 5(2), 60–66.

Njoku, J. N., Morocho-Cayamcela, M. E., & Lim, W. (2021). CGDNet: Efficient hybrid deep learning model for robust automatic modulation recognition. IEEE Networking Letters, 3(2), 47–51.

Xu, Y., Li, D., Wang, Z., Guo, Q., & Xiang, W. (2019). A deep learning method based on a convolutional neural network for automatic modulation classification of wireless signals. Wireless Networks, 25(7), 3735–3746.

Ramjee, S., Ju, S., Yang, D., Liu, X., Gamal, A., & Eldar, Y. (2019). Fast deep learning for automatic modulation classification: Special issues on machine learning in wireless communication.