IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

AI-Based Automatic Modulation Classification (AMC) Using a Convolutional Neural Network (CNN)

Nalini G[1], Prakashkumar B Tole[2], Kapileshwara N Katti[3], Mahesh CS[4]

Assistant Professor, Department of ECE, Bangalore institute of technology, Bangalore, India[1]

Student,B.E., Department of ECE, Bangalore institute of technology, Bangalore, India[2-4]

Abstract: Automatic modulation classification (AMC) identifies the modulation format of a received signal without help from the transmitter and underpins spectrum monitoring and cognitive radio. This paper presents a reproducible, end-to-end AMC framework that is built on a synthetic signal generator instead of a fixed benchmark file. Six schemes (BPSK, QPSK, 16-QAM, AM, FM and FSK) are synthesised, passed through AWGN, phase-noise and Rayleigh-fading channels at SNRs from −10 dB to 30 dB, and converted to 2×128 baseband I/Q frames by Hilbert-transform preprocessing. Two classifiers are studied: an interpretable baseline that uses six envelope and instantaneous-frequency statistics with Gaussian class models, and a convolutional neural network (CNN) whose layer structure follows a published Gaussian-noise-regularised design. The baseline reaches overall accuracies of 61.1%, 69.2% and 59.4% under AWGN, phase noise and Rayleigh fading respectively, and roughly 80% or more once the SNR is 10 dB or higher; BPSK/QPSK confusion is its dominant failure mode. The CNN reaches 89.5% overall test accuracy on a mixed-channel test set. An interactive dashboard exposes every stage and produces accuracy-versus-SNR curves, confusion matrices and training curves.

Index Terms*:* Automatic modulation classification, convolutional neural network, deep learning, I/Q data, Gaussian noise layer, synthetic dataset, feature-based classification

I. INTRODUCTION

Wireless spectrum is shared by an ever-growing number of cellular, Wi-Fi, IoT and unmanned-vehicle links. A receiver that must monitor or demodulate an unknown transmission first has to answer a basic question: which modulation was used? Automatic modulation classification (AMC) answers it directly from the received samples, and it has long been applied to spectrum surveillance, interference identification, cognitive radio and electronic warfare [1], [2].

Classical AMC follows two routes. Likelihood-based tests are optimal under a known model but need accurate channel knowledge, while feature-based methods compute statistics such as cumulants, spectral peaks or envelope measures and feed them to a decision rule [1], [2], [18]. Since the introduction of public I/Q datasets [3], [4], convolutional neural networks (CNNs) have become the dominant alternative because they learn features directly from raw samples [3], [5], [6], [12], [13].

Two practical gaps motivate this work. First, most CNN studies are evaluated on a fixed downloaded dataset, so the signal chain from waveform to decision is hidden from students and practitioners who want to see why a classifier fails at a given SNR. Second, deep models are rarely reported next to a transparent baseline trained on the same data, which makes it hard to judge what the extra complexity buys [19]. We therefore build the complete chain ourselves and study both a CNN and a simple feature classifier on identical signals.

The contributions of this paper are:

- A synthetic generator for three digital (BPSK, QPSK, 16-QAM) and three analog/FSK-type (AM, FM, FSK) schemes with three channel models (AWGN, phase noise, Rayleigh fading) and a controllable SNR.
- A uniform preprocessing chain that turns each received waveform into a power-normalised 2×128 baseband I/Q frame, the input format used by RadioML-style CNNs.
- A six-feature Gaussian classifier as an interpretable baseline, and a CNN whose layers follow the Gaussian-noise-regularised architecture of [12], adapted to six classes.
- An interactive dashboard that reproduces the standard AMC outputs: accuracy versus SNR, confusion matrices, training curves and live per-signal predictions.

II. RELATED WORK

A. Feature-based and classical machine-learning AMC

Feature-based AMC extracts discriminative statistics and applies a threshold tree or a trained classifier. A recent example computes power-, squared- and fourth-order-spectrum features, classifies with a binary decision tree, and runs on a ZYNQ7045 board, reporting more than 95% recognition accuracy above 10 dB SNR [18]. Another study applies



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.151161

KNN, SVM and logistic regression to flattened I/Q samples of a RadioML 2018 subset and reports accuracies near 90%, with a lightweight CNN only a little higher but much more costly to train [19]. These results motivate keeping a classical baseline next to the CNN.

B. CNN-based AMC

O'Shea et al. showed that a CNN on raw I/Q samples can match or beat expert features [3], and later work extended this to over-the-air data with residual networks [6] and to deeper architectures [5], [7]. Subsequent studies vary the input representation or the network: constellation images [8], I/Q combined with polar (amplitude/phase) features [15], multipath-fading robustness [11], different classification layers and optimisers [14], CNN-recurrent fusion [16], and 3-D constellation diagrams for impulsive noise [17]. A survey of signal representations and preprocessing is given in [13]. Of direct relevance, Jariwala and Captain [12] insert a Gaussian noise layer after the flatten operation of a three-convolution CNN and report 68.94% overall test accuracy on RadioML 2016.04C. Table I summarises these works.

TABLE I
SUMMARY OF RELATED WORK

Ref.	Year	Input / method	Data or setting
[3]	2016	Raw I/Q, CNN	RadioML 2016 (11 classes)
[6]	2018	Raw I/Q, residual CNN	Over-the-air, 24 classes
[11]	2020	CNN, class families	Multipath fading (HisarMod)
[12]	2021	I/Q, CNN + Gaussian noise layer	RadioML 2016.04C; 68.94% overall
[15]	2023	I/Q plus polar features, CNN	RadioML 2018.01A; 96.06% at 20 dB
[14]	2023	CNN, three classification layers	Optimiser / loss comparison
[16]	2023	ResNeXt-GRU, feature fusion	Deep-fusion AMC
[17]	2024	3-D constellation, 3DCNN	Impulsive noise
[18]	2026	Spectral features, binary tree	ZYNQ7045 hardware; >95% above 10 dB
[19]	2026	KNN / SVM / LR vs light CNN	RadioML 2018 subset, ~90%
This work	2026	I/Q CNN + 6-feature baseline	Synthetic, 6 classes, 3 channels

III. SYSTEM MODEL AND SYNTHETIC DATASET

A. Signal generation

Each realisation is a real-valued carrier waveform of N = 480 samples (8 symbols of 60 samples for symbol-based schemes). The number of carrier cycles per frame is interpolated from 16 to 64 as the carrier slider moves from 800 to 5000 MHz; the literal RF frequency is not simulated. With carrier ω_c, message ω_m and frame length N, the schemes are generated as

$$s(t) = Re\{(I_k + jQ_k)\, e^{j\omega_{-}{}^{fc}t}\} \quad (1)$$

$$s[n] = [1 + m\sin(\omega_m n + \varphi)]\cos(\omega_c n) \quad (2)$$

$$s[n] = \cos(\omega_c n + \beta(2\pi/N)\,\Sigma_{k\le n}\sin(\omega_m k + \varphi)) \quad (3)$$

$$s[n] = \cos(\Sigma_{k\le n}[\omega_c + d_k \cdot 2\pi\Delta/N]) \quad (4)$$

Eq. (1) covers BPSK, QPSK and 16-QAM, where (I_k, Q_k) is a random unit-average-power constellation point per symbol. Eq. (2) is AM with m = 0.7, Eq. (3) is FM with deviation index $\beta = 5$, and Eq. (4) is binary FSK with $d_k \in \{-1, +1\}$ and $\Delta = 8$ cycles per frame, accumulated so that the phase stays continuous across symbols. The message phase φ is drawn uniformly at random for AM and FM.

B. Channel models

For a target SNR the noise variance is

$$\sigma^2 = P_s / 10^{SNR/10} \quad (5)$$

where P_s is the mean signal power. Three impairments are applied: (i) *AWGN*, which adds zero-mean Gaussian noise of variance σ^2; (ii) *phase noise*, a random-walk phase drift whose step deviation is $0.05 \cdot 10^{-SNR/40} + 0.01$ radian, followed by AWGN at 60% of the noise power; and (iii) *Rayleigh fading*, a Rayleigh-distributed envelope smoothed by a 25-tap moving average and renormalised to unit mean, followed by AWGN. These models are deliberately simple; they do not include multipath delay spread, carrier-frequency offset or sample-rate offset, all of which are present in GNU Radio–generated benchmark data [4].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

C. Dataset configuration

Table II lists the configuration. SNR takes nine values from −10 dB to 30 dB in 5 dB steps. The CNN dataset mixes the three channels uniformly at random and is split 70/15/15% into training, validation and test sets. The baseline is trained and tested separately for every channel and SNR.

TABLE II
SIMULATION AND DATASET PARAMETERS

Parameter	Value
Classes (6)	BPSK, QPSK, 16-QAM, AM, FM, FSK
Raw frame length	480 samples (8 × 60)
Carrier (default)	2400 MHz, ≈34 cycles per frame
SNR levels	−10 to 30 dB, step 5 (9 levels)
Channels	AWGN, phase noise, Rayleigh
CNN input	2×128 baseband I/Q frame
CNN dataset (default)	150 frames per class per SNR = 8,100
CNN split	70% train / 15% val / 15% test
Baseline training	60 frames per class, per SNR and channel
Baseline testing	40 frames per class, per SNR and channel (2,160)

IV. PREPROCESSING AND FEATURE EXTRACTION

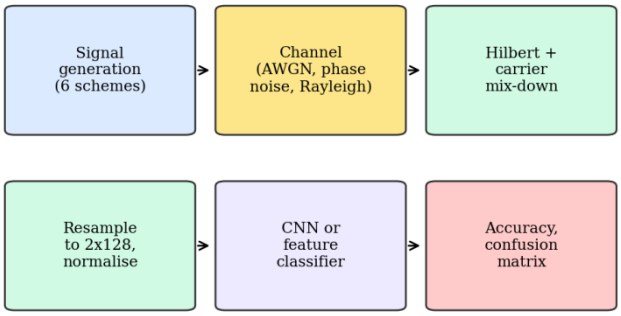


Fig. 1. Processing chain from signal generation to evaluation.

A. CNN input frame

The receiver is assumed to know its tuned carrier. The received waveform is converted to an analytic signal with the Hilbert transform, mixed down to baseband, resampled to 128 samples, and normalised to unit average power:

$$z[n] = (s[n] + j\cdot H\{s\}[n])\, e^{-j\omega_{-}\{c\}n} \quad (6)$$

$$X = [\mathrm{Re}\,\tilde{z}\,;\, \mathrm{Im}\,\tilde{z}] \in R^{2\times 128},\ \tilde{z} = z / \sqrt{(mean|z|^2)} \quad (7)$$

Resampling acts as a low-pass filter, which removes part of the out-of-band noise, and power normalisation removes the absolute signal level. Fig. 2 shows the resulting I and Q traces at 10 dB SNR, and Fig. 3 shows I–Q scatter plots for the three digital schemes at 20 dB.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

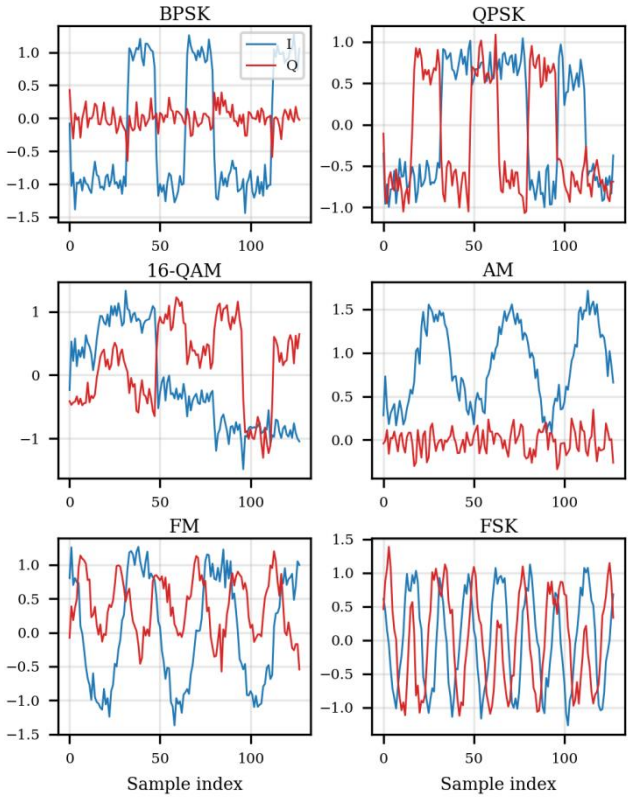


Fig. 2. In-phase (I) and quadrature (Q) components of the 2×128 baseband frame for each scheme (AWGN, SNR = 10 dB).

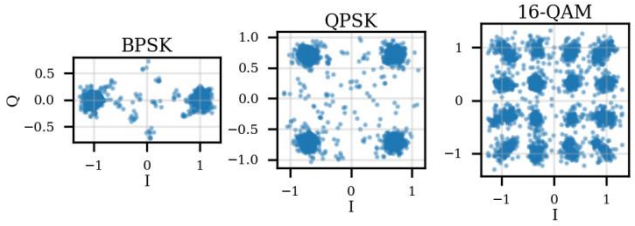


Fig. 3. I–Q scatter of BPSK, QPSK and 16-QAM frames at 20 dB SNR (AWGN).

B. Feature vector for the baseline

For the baseline the analytic signal is formed in the same way; the envelope is divided by its mean, the unwrapped phase is differenced to obtain instantaneous frequency, and a 9-tap moving average suppresses frequency noise. Six statistics are then computed (Table III).

TABLE III
FEATURES USED BY THE BASELINE CLASSIFIER

Feature	**What it measures**
Envelope CV	Spread of normalised amplitude (high for AM, 16-QAM)
Envelope kurtosis	Shape of the amplitude distribution
Frequency std	Spread of smoothed instantaneous frequency (FM, FSK)
Frequency kurtosis	Smooth versus jumpy frequency trajectory
Frequency jumpiness	Mean |Δf| divided by frequency std
Envelope 5th pct.	Depth of envelope dips

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

V. CLASSIFIERS

A. Gaussian feature classifier (baseline)

For every class c, the mean μ_c and standard deviation σ_c of the six features are estimated from 60 synthetic training frames generated at the same SNR and channel. A test frame with feature vector x is scored by the diagonal-Gaussian log-likelihood and the softmax of the scores gives a confidence for each class:

$$\log p(x|c) = -\tfrac{1}{2} \Sigma_d \left((x_d - \mu_{cd})/\sigma_{cd}\right)^2 - \Sigma_d \log \sigma_{cd} \qquad (8)$$

The predicted class is the one with the highest score. Because the models are fitted per SNR, this baseline implicitly assumes that the SNR is known or estimated, which is an advantage over the CNN (Section VIII).

B. CNN architecture

The CNN follows the six-stage layout reported in [12] (Fig. 4, Table IV): three convolution stages with ReLU and dropout, one average-pooling stage, a flatten layer followed by a Gaussian noise layer that acts as a regulariser, and dense layers of 1024, 256 and 64 units before a softmax output. Dropout is 0.3, because the rate is not specified in [12], and the Gaussian noise standard deviation is 10^{-4}, the value reported there. The output layer has six units instead of eleven.

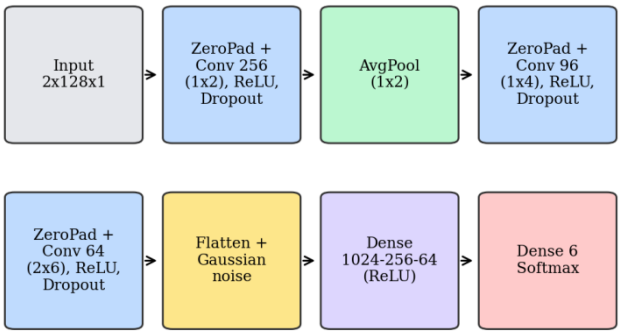


Fig. 4. CNN architecture used in this work (adapted from [12]).

TABLE IV
CNN LAYER SUMMARY (INPUT 2×128×1)

Layer	Output shape	Params
ZeroPad (0,2) + Conv 256, 1×2	(2,131,256)	768
AvgPool 1×2	(2,65,256)	0
ZeroPad (0,2) + Conv 96, 1×4	(2,66,96)	98,400
ZeroPad (0,2) + Conv 64, 2×6	(1,65,64)	73,792
Flatten + GaussianNoise	4,160	0
Dense 1024	1,024	4,261,888
Dense 256	256	262,400
Dense 64	64	16,448
Dense 6 + softmax	6	390
Total		**4,714,086**

Each convolution is followed by ReLU and dropout (0.3). Parameter counts are computed analytically; confirm with model.summary().

C. Training protocol

The CNN is trained with the Adam optimiser [20] and categorical cross-entropy, batch size 128, up to 25 epochs, with early stopping on validation loss (patience 6, best weights restored). Dropout [21] and the Gaussian noise layer provide regularisation. Training uses 70% of the 8,100 frames, validation 15% and testing 15%, drawn at random with a fixed seed.

VI. SOFTWARE IMPLEMENTATION

The framework is implemented in Python with NumPy, SciPy, Plotly, Streamlit and TensorFlow/Keras. The dashboard has five parts: (i) controls for scheme, carrier frequency, SNR, channel, distance and transmit power; (ii) a 3-D carrier-wave view and a 2-D oscilloscope view with the clean and received waveforms; (iii) a confidence bar chart for the six

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

candidate schemes together with a free-space path-loss link-budget panel; (iv) a batch evaluation panel that produces accuracy-versus-SNR curves and confusion matrices; and (v) a CNN panel that generates the dataset, trains the network and reports loss/accuracy curves, per-SNR confusion matrices and a live comparison between the CNN and the baseline on the currently simulated signal.

VII. Results and Discussion

A. Baseline accuracy versus SNR

Table V and Fig. 5 give the baseline accuracy per SNR for the three channels (40 frames per class per SNR, 2,160 test frames per channel, carrier 2400 MHz). At −10 dB and −5 dB the accuracy is between 14% and 27%, close to the 16.7% chance level for six classes. It then rises steeply between 0 dB and 10 dB and settles at about 80–95% for SNR ≥ 10 dB. Overall accuracy is 61.1% (AWGN), 69.2% (phase noise) and 59.4% (Rayleigh fading). The higher phase-noise figure should be read with care: in our model that channel adds noise at only 60% of the nominal power, so its effective SNR is higher than in the other two cases.

TABLE V
Baseline Accuracy (%) versus SNR

SNR (dB)	AWGN	Phase noise	Rayleigh	CNN (mixed)
−10	14.2	19.6	13.8	68.9
−5	20.0	26.7	16.7	76.3
0	32.5	44.6	33.8	85.6
5	62.9	75.0	61.3	93.8
10	81.2	83.8	79.6	93.1
15	84.6	89.2	82.9	95.5
20	82.1	95.8	81.2	96.0
25	85.4	94.2	80.0	97.6
30	86.7	94.2	85.0	98.6
Overall	**61.1**	**69.2**	**59.4**	89.5

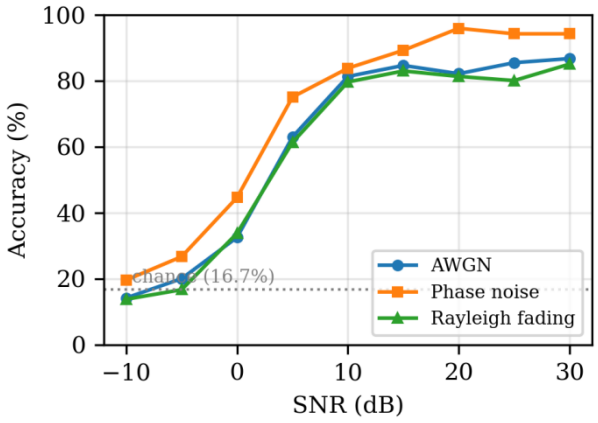


Fig. 5. Baseline classification accuracy versus SNR for the three channels. The dotted line marks chance level (1/6).

B. Per-class behaviour and confusion

Under AWGN, AM is recognised best (79.4%), followed by FM (69.4%), 16-QAM (66.7%), FSK (64.4%) and QPSK (60.8%), while BPSK is weakest (25.6%). The confusion matrix in Fig. 6 shows why: about 49% of BPSK frames are labelled QPSK. Both are constant-envelope phase-modulated signals, so the six features, which describe amplitude and frequency statistics only, carry little information about the phase alphabet. This mirrors the confusion among similar modulations that is common in published AMC results [12], [19], and it is a clear case where learned features could help. AM and 16-QAM are easier because their envelopes vary strongly.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

TABLE VI
PER-CLASS ACCURACY (%), ALL SNR VALUES

Class	AWGN	Phase noise	Rayleigh	CNN (mixed)
BPSK	25.6	53.3	26.7	94.3
QPSK	60.8	59.4	49.2	74.3
16-QAM	66.7	74.4	66.9	75.3
AM	79.4	84.4	82.2	99.0
FM	69.4	71.4	66.1	97.9
FSK	64.4	72.2	65.0	96.5

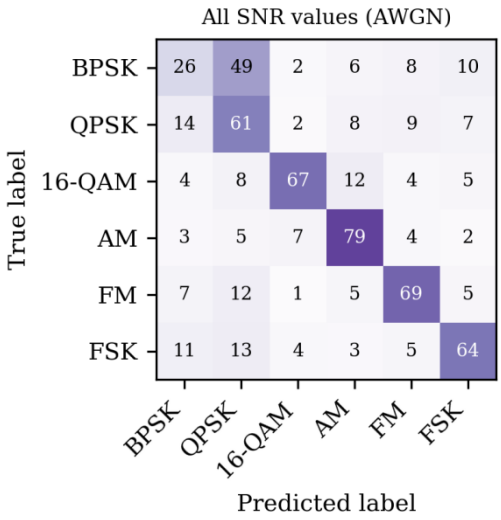


Fig. 6. Baseline confusion matrix over all SNR values (AWGN); entries are row-normalised percentages.

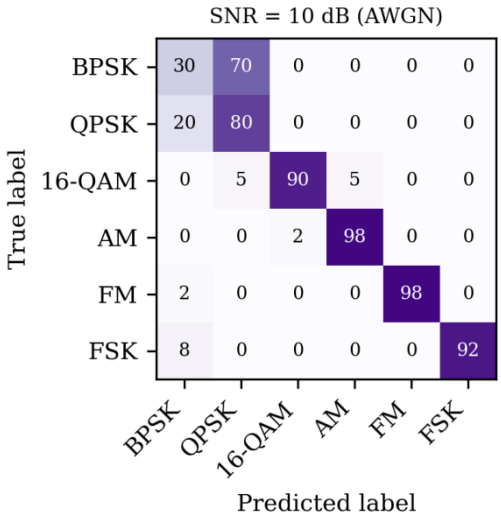


Fig. 7. Baseline confusion matrix at SNR = 10 dB (AWGN).

C. CNN results

After training, the CNN reached 89.5% accuracy on the held-out test split (1215 frames, all SNR values and all three channels mixed) with a test loss of 0.244. Early stopping ended training after 11 epochs (1.2 minutes on the authors' machine); the network has 4,713,062 parameters. Fig. 8 shows the training and validation curves. The final training and validation accuracies are 99.0% and 91.2% (a gap of 7.8 points).

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.151161

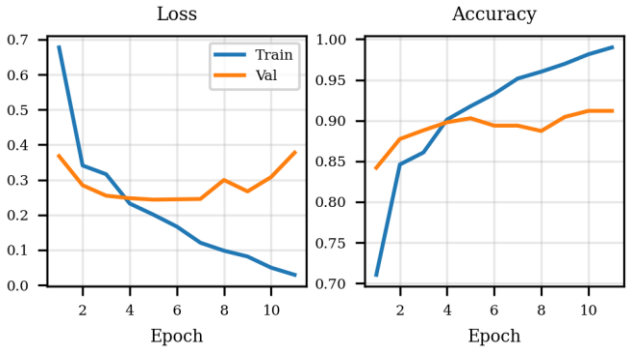


Fig. 8. CNN training and validation loss and accuracy versus epoch.

Fig. 9 compares the CNN with the feature baseline across SNR and Fig. 10 gives the CNN confusion matrix. The overall CNN accuracy of 89.5% is higher than the baseline overall accuracies of 59.4–69.2% (note that the baseline was tested on single-channel sets at known SNR, so the two numbers are not strictly equal-footing). BPSK, the weakest class for the baseline (25.6% under AWGN), is recognised in 94.3% of cases by the CNN, and 1.8% of BPSK frames are labelled QPSK.

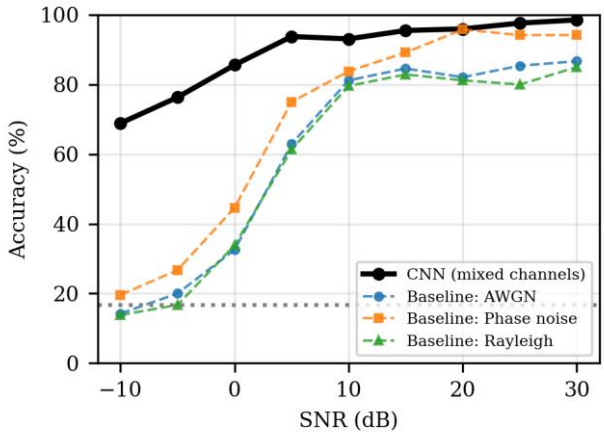


Fig. 9. Accuracy versus SNR: CNN (mixed channels) and feature baseline (per channel).

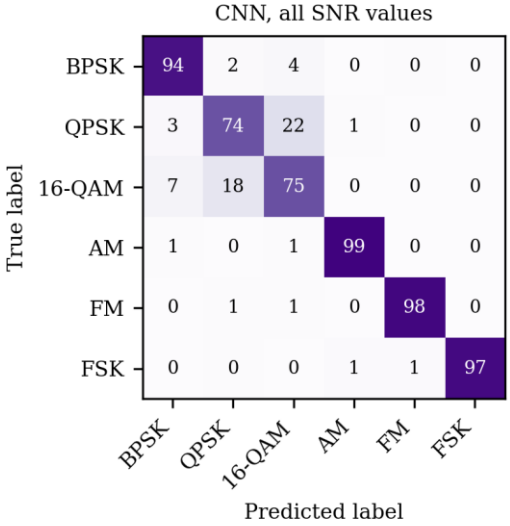


Fig. 10. CNN confusion matrix over all SNR values (row-normalised percentages).

D. Discussion

Three points emerge. First, the SNR curve has the same shape as in benchmark studies: near chance at very low SNR, a steep rise around 0–10 dB, and saturation at high SNR [12], [15]. Second, the baseline is easy to inspect, so its errors can be explained physically, which is valuable in teaching and in applications that need explainable decisions [19]. Third,

DOI: 10.17148/IJARCCE.2026.151161

a learned model removes the need to choose features by hand, at the price of more data, a longer training time and less transparency. Reference [19] reports that a lightweight CNN gave only a small accuracy gain over classical methods at a much higher training cost; whether a similar trade-off appears on our synthetic data is answered by the comparison in Tables V and VI.

VIII. Limitations

- The data are synthetic and the channels are simplified; results do not transfer directly to over-the-air signals or to benchmark datasets such as RadioML.
- Only six classes are used, and the frame holds eight symbols for symbol-based schemes, so the numbers are not comparable with 11- or 24-class benchmarks.
- The baseline is trained per SNR and channel, whereas the CNN is trained on all SNRs and channels together, so the comparison favours the baseline on knowledge of the operating condition.
- The baseline results come from one random seed and 40 frames per class and SNR; confidence intervals are not reported.
- The receiver is assumed to know the carrier frequency used for the mix-down.
- Hardware validation, as in [18], is not performed.

IX. Conclusion and Future Work

We presented a reproducible AMC framework that generates six modulation schemes, applies three channel models, preprocesses signals into 2×128 I/Q frames and compares a six-feature Gaussian baseline with a Gaussian-noise-regularised CNN. The baseline reaches roughly 80% or more above 10 dB SNR but cannot reliably separate BPSK from QPSK; the CNN results are reported in Section VII-C. The accompanying dashboard makes each stage visible, which supports teaching and analysis. Future work includes adding more classes and impairments (frequency offset, multipath), training the baseline across mixed SNRs, repeating experiments over several seeds, evaluating on RadioML and recorded signals, and testing recurrent or attention-based extensions [16].

References

[1] E. E. Azzouz and A. K. Nandi, “Procedure for automatic recognition of analogue and digital modulations,” *IEE Proc.–Commun.*, vol. 143, no. 5, pp. 259–266, 1996.

[2] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, “Survey of automatic modulation classification techniques: Classical approaches and new trends,” *IET Commun.*, vol. 1, no. 2, pp. 137–156, 2007.

[3] T. J. O’Shea, J. Corgan, and T. C. Clancy, “Convolutional radio modulation recognition networks,” in *Engineering Applications of Neural Networks*, Springer, 2016, pp. 213–226.

[4] T. J. O’Shea and N. West, “Radio machine learning dataset generation with GNU Radio,” in *Proc. 6th GNU Radio Conf.*, 2016.

[5] N. E. West and T. O’Shea, “Deep architectures for modulation recognition,” in *Proc. IEEE DySPAN*, 2017, pp. 1–6.

[6] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[7] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, “Deep learning models for wireless signal classification with distributed low-cost spectrum sensors,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 3, pp. 433–445, Sep. 2018.

[8] S. Peng *et al.*, “Modulation classification based on signal constellation diagrams and deep learning,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019.

[9] Y. Wang, M. Liu, J. Yang, and G. Gui, “Data-driven deep learning for automatic modulation recognition in cognitive radios,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[10] A. P. Hermawan, R. R. Ginanjar, D. Kim, and J. Lee, “CNN-based automatic modulation classification for beyond 5G communications,” *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1038–1041, May 2020.

[11] K. Tekbıyık, A. R. Ekti, A. Görçin, G. Karabulut Kurt, and C. Keçeci, “Robust and fast automatic modulation classification with CNN under multipath fading channels,” in *Proc. IEEE 91st Veh. Technol. Conf. (VTC2020-Spring)*, 2020.

[12] D. A. Jariwala and K. M. Captain, “Automatic modulation classification: A novel convolutional neural network based approach,” in *Proc. IEEE 18th India Council Int. Conf. (INDICON)*, 2021, doi: 10.1109/INDICON52576.2021.9691687.

[13] S. Peng, S. Sun, and Y.-D. Yao, “A survey of modulation classification using deep learning: Signal representation and data preprocessing,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 7020–7038, Dec. 2022.

[14] H. S. Hussein *et al.*, “Automatic modulation classification: Convolutional deep learning neural networks approaches,” *IEEE Access*, vol. 11, 2023.

 This work is licensed under a Creative Commons Attribution 4.0 International License

DOI: 10.17148/IJARCCE.2026.151161

[15] A. H. Shah, A. H. Miry, and T. M. Salman, "Automatic modulation classification based deep learning with mixed feature," *Int. J. Electr. Comput. Eng.*, vol. 13, no. 2, pp. 1647–1653, 2023, doi: 10.11591/ijece.v13i2.pp1647-1653.

[16] L. Li, Y. Zhu, and Z. Zhu, "Automatic modulation classification using ResNeXt-GRU with deep feature fusion," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 2519710, doi: 10.1109/TIM.2023.3290301.

[17] J. Bai, J. Yao, Y. Guo *et al.*, "Automatic modulation classification in impulsive noise: Log-domain 3-D constellation diagrams and multiscale dual-convolutional 3DCNN," *IEEE Trans. Cogn. Commun. Netw.*, vol. 10, no. 4, pp. 1204–1217, 2024, doi: 10.1109/TCCN.2024.3375510.

[18] J. Q. Chen, H. L. Lu, Y. X. Li, K. Gao, W. C. Zheng, and S. Cui, "Automatic modulation classification of common electromagnetic signals," in *Proc. 2026 Cross Strait Radio Sci. Wireless Technol. Conf. (CSRSWTC)*, 2026, doi: 10.1109/CSRSWTC71154.2026.11669417.

[19] H. Ma and X. Wang, "Signal modulation classification based on machine learning algorithms," in *Proc. 2026 IEEE 3rd Int. Conf. Big Data Sci. Eng. (ICBDSE)*, 2026, doi: 10.1109/ICBDSE70225.2026.11635747.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2015.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.