2025 IEEE 2nd International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)
Bengaluru, India. Aug 30-31, 2025

Experimental Validation of Decision Tree-Based Adaptive Modulation and Coding Prediction for Multiple Users in Dynamic Wireless Environment

1st P G. Varna Kumar Reddy
Department of ECE,
Vels Institute of Science, Technology & Advanced Studies (VISTAS),
Chennai, India
varunkumarreddy.pg@gmail.com

2nd M. Meena
Department of ECE,
Vels Institute of Science, Technology & Advanced Studies (VISTAS),
Chennai, India
meena.se@velsuniv.ac.in

***Abstract*— The increasing complexity of modern wireless communication environments necessitates efficient and accurate modulation and code rate classification to optimize system performance. This paper investigates the use of a Decision Tree (DT) algorithm for adaptive modulation and coding scheme (MCS) prediction in multi-user scenarios. Leveraging real-time datasets collected from a Software Defined Radio (SDR) platform using USRP N210 and GNU Radio, the proposed approach classifies modulation type and code rate based on key channel parameters, including user distance from the base station, channel noise, and signal-to-noise ratio (SNR). The DT algorithm's performance is evaluated using both Gini Index and Entropy criteria, with results analyzed in terms of predicted values, confusion matrices, and classification accuracy. Experimental findings demonstrate that the Gini Index yields superior accuracy compared to Entropy, achieving up to 100% classification performance for certain parameters. Furthermore, higher user counts (50 users) enhance prediction accuracy due to richer training data. These results confirm the potential of decision tree-based classification for real-time adaptive MCS selection in dynamic wireless environments.**

Index Terms*— *Adaptive modulation and coding, decision tree, machine learning, modulation classification, software defined radio.

I. INTRODUCTION

Wireless communication systems have evolved rapidly over the past two decades, driven by the need for higher data rates, improved spectral efficiency, and reliable performance under diverse operating conditions. The transition from early analog systems to today's sophisticated digital multi-antenna and multi-carrier systems has been enabled by advancements in modulation, coding, and adaptive resource management techniques. One of the most significant developments in this regard is Adaptive Modulation and Coding (AMC), a dynamic link adaptation strategy that adjusts the modulation format and coding rate in real time based on instantaneous channel conditions. AMC has become a cornerstone of modern communication standards such as LTE, LTE-Advanced, WiMAX, and 5G New Radio, enabling systems to maximize throughput when the channel is favourable while maintaining robustness during challenging conditions [1].

At its core, AMC operates on the principle that when the channel quality is high often measured through metrics such as the SNR or Signal-to-Interference-plus-Noise Ratio (SINR) the system can use higher-order modulation schemes and higher code rates to transmit more bits per symbol. Conversely, when the channel quality deteriorates, the system switches to more robust, lower-order modulation schemes and/or lower code rates to improve the Bit Error Rate (BER) performance [2]. This dynamic adaptation optimizes the trade-off between spectral efficiency and error resilience, ensuring that the system operates as close as possible to the Shannon capacity limit without sacrificing reliability.

In digital communications, modulation is the process of mapping binary data into discrete symbols represented by points in a signal constellation, such as Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), or higher-order Quadrature Amplitude Modulation (QAM) formats [3]. Channel coding, on the other hand, introduces redundancy into the transmitted data stream using error-correcting codes, allowing the receiver to detect and correct bit errors caused by noise, interference, or fading [4]. The code rate R_c is defined as the ratio of the number of information bits to the total number of transmitted bits. For example, a code rate of ¾ means that for every 4 bits transmitted, 3 carry user data while 1 is a redundancy bit. The overall spectral efficiency η of a given MCS is given by:

$$\eta = R_c \log_2 M \qquad (1)$$

where M is the modulation order.

The selection of an optimal MCS in real time is influenced by several factors beyond SNR, including user distance from the base station, propagation losses, fading statistics, channel noise, interference levels, and user mobility. In practice, wireless systems employ AMC lookup tables that map measured channel quality indicators (CQIs) to predefined MCS entries. For instance, a user near the base station with a strong SNR might be assigned 256-QAM with a 7/8 code rate, achieving very high throughput, while a cell-edge user experiencing poor SNR might be assigned BPSK with a 1/2 code rate to ensure reliable communication [5]. While static threshold-based AMC mappings are straightforward to implement, they are often suboptimal in rapidly changing or heterogeneous environments, as they cannot fully capture the complex, multidimensional relationships between channel parameters and optimal MCS selection. Accurate real-time MCS prediction is a challenging problem because wireless channels are highly dynamic due to multipath fading, shadowing, Doppler effects, and variations in network load. Moreover, traditional threshold-based methods may fail to adapt effectively when the environment changes in ways not anticipated during system design [6]. This limitation has motivated the use of machine learning (ML) techniques, which can learn complex nonlinear mappings between multiple input parameters such as SNR, channel noise, and distance and the optimal MCS configuration. Once trained on representative datasets, ML models can generalize to unseen channel conditions, enabling more robust and adaptive link adaptation strategies [7]-[9].

A number of studies have examined modulation In contrast, the present work addresses the research gap by exploring a lightweight, interpretable, and computationally efficient Decision Tree model for joint modulation and code rate classification [10]-[12]. By training on a realistic SDR-generated dataset, the model learns the relationship between multiple channel parameters and optimal MCS selection, enabling accurate real-time prediction. This approach also allows for direct comparison of Gini Index and Entropy as splitting criteria, offering practical insights into their performance trade-offs for this application [13]-[15]. Moreover, the analysis includes an evaluation of scalability by comparing results for datasets with different numbers of users, providing a deeper understanding of how training data volume impacts accuracy [16].

The key contributions of this work are Decision Tree-based framework is proposed for predicting both modulation scheme and code rate in adaptive wireless systems using SNR, channel noise, and distance as input features. Second, a real-world dataset is generated using a USRP N210 SDR platform and GNU Radio, capturing the effects of realistic channel impairments. Third, the performance of Gini Index and Entropy criteria is compared systematically, with results showing that the Gini Index yields superior classification accuracy in this context. Fourth, the impact of dataset size on prediction performance is investigated, with findings indicating that larger datasets such as those with 50 users enhance accuracy due to richer training information. Finally, detailed experimental results are presented, including predicted values, confusion matrices, and classification accuracy, demonstrating the feasibility of deploying the proposed approach in practical systems.

The rest of this paper is organized as follows. Section 2 introduces the Decision Tree algorithm, including the theoretical background and the mathematical formulation of impurity measures such as Gini Index and Entropy. Section 3 describes the experimental setup, detailing the SDR hardware and software configuration and the process of dataset generation. Section 4 presents and discusses the results, comparing the performance of different split criteria under various conditions. Section 5 concludes the paper by summarizing the findings and outlining potential directions for future research.

II. DECISION TREE ALGORITHM

Decision Tree algorithms are among the most widely used supervised machine learning techniques for both classification and regression tasks, owing to their simplicity, interpretability, and computational efficiency. In the context of AMC, DTs offer a structured approach to mapping multiple channel parameters such as SNR, channel noise, and user distance to an optimal MCS. Unlike complex black-box models such as deep neural networks, DTs provide clear, rule-based decision paths that are easily understood, verified, and deployed in real-time wireless systems.

A Decision Tree consists of three fundamental types of nodes:

Root Node – Represents the entire dataset and is the starting point of the classification process.

Internal Nodes (Decision Nodes) – Contain decision rules based on feature values that split the data into subsets.

Leaf Nodes (Terminal Nodes) – Represent final classification outcomes, which, in this application, correspond to specific MCS configurations.

The algorithm works by recursively partitioning the feature space into increasingly homogeneous subsets based on a chosen splitting criterion. The objective at each node is to select the feature and corresponding threshold that best separates the data into distinct classes, thereby minimizing impurity [17].

A. Splitting Criteria

The performance of a DT depends heavily on the impurity measure used to evaluate potential splits. In this work, two widely adopted criteria are considered: Entropy/Information Gain and Gini Index [18].

1) Entropy

The value of entropy is always between 0 and 1. When it is equal to 0, it is better than when it is equal to 1, while when it is equal to 1, it is worse. The entropy is computed as follows:

$$Entropy\,(H) = \sum_{x=1}^{n} P_x \log_2^x \tag{2}$$

Where Px is the ratio of the sample number of the subset and the nth attribute value.

2) Information *gain*

One of the most common metrics used for segmentation is information gain. This is a measure of how much knowledge there is about a random variable's value. It's the opposite of entropy, where the higher its value, the better. The data gain is defined as the following:

$$\begin{aligned}\text{Gain(H,A)} &= \text{Entropy}(H) \\ &\quad - \sum_{\text{Values}(A)} \frac{|H_v|}{|H|}.\text{Entropy}(H_v)\end{aligned} \tag{3}$$

Where the range of attribute A is *Values*(A), and Hv is a subset of set H equal to the attribute value of attribute v.

3) GINI index

The GINI index is a measure of the purity of a class after it has split along a specific attribute. The best split results in an increase in the sets' purity. If a dataset D has multiple class labels k, the GINI index is calculated as follows.

$$Gini(L) = 1 - \textstyle\sum_{i=1}^{k} P_i^2 \tag{4}$$

The relative frequency pi of the ith class is computed if the data is split into two subsets, D1 and D2, with sizes S1 and S2. Then, the Gini is defined as

$$Gini_A(D) = \frac{S_1}{S}Gini(D_1) + \frac{S_2}{S}Gini(D_2) \tag{5}$$

Reduction in impurity is calculated as

$$\Delta Gini(A) = Gini(D) + Gini_A(D) \tag{6}$$

Compared to Entropy, the Gini Index is computationally less expensive and often yields similar results. In this work, empirical results demonstrate that the Gini Index outperforms

Entropy for MCS classification tasks. Fig.1 shows the basic structure of DT.

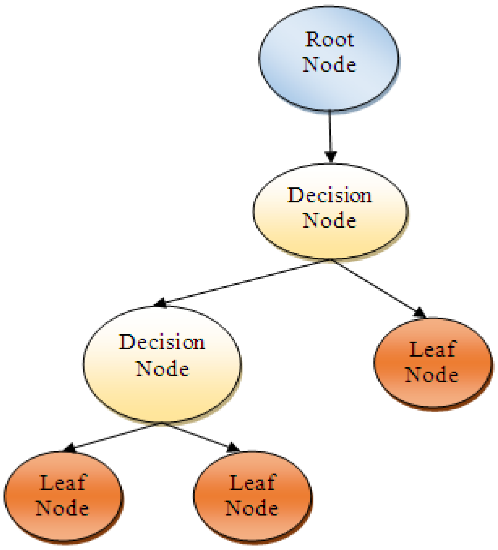

Fig. 1. Decision tree model

B. *Tree Construction*

The Decision Tree algorithm employs a top-down, greedy strategy known as Recursive Binary Splitting to construct the model. At each stage, the algorithm evaluates all possible feature-based splits using the selected impurity measure, such as the Gini Index or Entropy. The split that yields the greatest improvement in node purity is chosen, and the dataset is partitioned accordingly. This process is applied recursively to each resulting subset, progressively creating a tree structure. The recursion terminates when one of several stopping criteria is met: all samples in a node belong to the same class, the maximum tree depth is reached, the number of samples in a node falls below a predefined threshold, or the reduction in impurity falls below a set limit. Once the tree is trained, classifying a new instance involves traversing it from the root node to a leaf node, following the decision rules established during training to determine the predicted class.

C. *Advantages of DT in AMC Prediction*

In the context of AMC prediction, DTs present several notable advantages that make them well-suited for practical wireless communication systems. A key strength of DTs is their interpretability, as each decision path within the tree represents a transparent set of conditions based on channel parameters such as SNR, user distance, and channel noise. This transparency allows engineers to trace and understand how specific classification outcomes are derived, facilitating both validation and fine-tuning of the model. Another advantage is their low computational complexity, as DTs rely primarily on simple threshold comparisons during inference. This efficiency enables real-time operation in resource-constrained environments, such as SDR based systems. Furthermore, DTs are capable of modelling nonlinear decision boundaries, making them effective for capturing the intricate relationships between channel conditions and the optimal MCS without requiring explicit analytical models. They can also handle diverse input types, including continuous features and categorical features, without extensive preprocessing. However, DTs are not without limitations, as they can be prone to overfitting when trained on small or noisy datasets. This challenge can be mitigated through techniques such as pruning, which eliminates branches that offer minimal improvement to classification performance.

III. EXPERIMENTAL SETUP

The experimental setup for this study was designed to generate a realistic dataset for training and evaluating the DT algorithm in predicting the MCS under varying wireless channel conditions. A SDR platform was employed, comprising the USRP N210 hardware from Ettus Research and the GNU Radio software framework. The USRP N210 offers high-speed, high-resolution data acquisition with flexible RF front-end configuration, making it suitable for implementing custom transmitter and receiver designs. GNU Radio was used to create a modular signal processing chain, enabling real-time control over modulation type, code rate, and channel impairments.

In the data collection phase, the system was configured to transmit and receive signals across multiple MCS configurations, including various modulation and different code rates. The wireless channel conditions were varied systematically by adjusting three key parameters: distance between the user and the base station, channel noise (measured in dBrn), and SNR (in dB). Data was collected for two scenarios: one involving 20 active users and another involving 50 users, to study the impact of dataset size on classification accuracy.

The dataset included the presence or absence of a user, the encoded MCS label, and the measured channel parameters. For each configuration, multiple samples were recorded to ensure statistical reliability and to capture variations due to channel randomness. The resulting dataset provided a comprehensive set of labeled examples for supervised learning. By using an SDR-based setup instead of purely simulated data, the collected measurements reflect real-world channel impairments such as multipath fading, hardware imperfections, and environmental noise, thereby enhancing the robustness and practical relevance of the trained DT model. This approach ensures that the experimental evaluation closely matches realistic deployment scenarios.

IV. RESULTS AND DISCUSSIONS

The performance of the DT algorithm for MCS prediction was evaluated using the real-time dataset generated from the SDR-based experimental setup. The classification was performed using two different split criteria Gini Index and Entropy to assess their comparative effectiveness. The evaluation metrics considered include predicted values, confusion matrices, and overall classification accuracy. Results are presented for multiple input–output mappings, such as user presence versus distance, modulation code rate encoded data, channel noise, and SNR for both 20-user and 50-user cases.

Fig. 2 illustrates the visual relationships between the key channel parameters and the modulation code rate encoded data in the dataset collected using the SDR testbed. Subfigure (a) plots Distance (km) versus Channel Noise (dBrn), revealing the general trend that as the user's distance from the base station increases, channel noise levels tend to rise due to increased path loss and susceptibility to interference. Subfigure (b) shows Distance versus Modulation Code Rate encoded data, indicating that higher-order MCS configurations are predominantly assigned to users closer to the base station, while lower-order, more robust configurations are assigned at greater distances. Subfigure (c) presents Channel Noise versus Modulation Code Rate encoded data, where a clear inverse relationship is observed—lower noise levels correspond to higher-order MCS assignments. Subfigure (d) plots Channel Noise versus SNR (dB), highlighting the expected negative correlation; higher noise results in lower SNR values. Finally, subfigure (e) illustrates Distance versus SNR, confirming that users farther from the base station generally experience reduced SNR due to propagation losses.

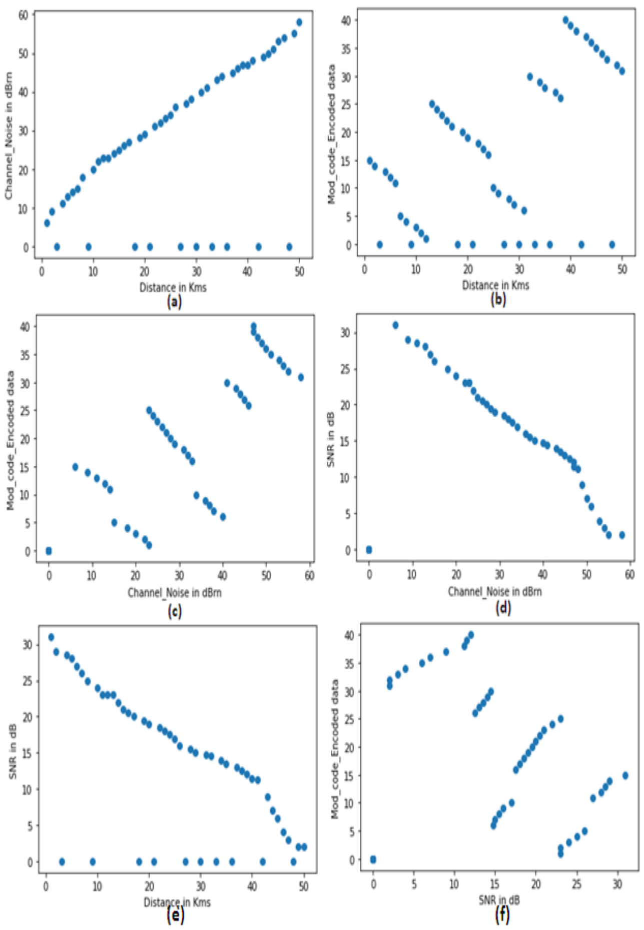

Fig. 2. Dataset visualization (a) Distance in Kms Vs Channel noise in dBm (b) Distance in Kms Vs Modulation code rate encoded data (c) Channel noise in dBm Vs Modulation code rate encoded data (d) Channel noise in dBm Vs SNR in dB (e) Distance in Kms Vs SNR in dB

The results in Table I indicate that the Gini Index criterion consistently achieved high classification accuracy, with several parameters reaching 100% accuracy. Notably, the model achieved perfect classification for modulation code rate encoded data, channel noise, and SNR (50 users). Even in cases where perfect accuracy was not attained, such as user presence versus distance and SNR (20 users), the algorithm achieved 90% and 95% accuracy, respectively. The superior performance with 50 users can be attributed to the larger training dataset, which improved the model's ability to generalize across varying channel conditions.

As shown in Table II, the Entropy-based DT classifier yielded comparatively lower performance across most parameters. While it achieved high accuracy in predicting modulation code rate encoded data (90%) and channel noise (90%), its performance for other parameters—such as user presence versus distance (80%) and SNR (20 users) (75%)—was noticeably lower than that of the Gini Index. The slightly lower performance is consistent with known tendencies of the Entropy criterion to be more sensitive to imbalanced splits and noise in the dataset.

TABLE I. DECISION TREE ALGORITHM RESULTS USING GINI INDEX

Presence of user Vs	Parameters		
	Predicted values	Confusion Matrix	Accuracy (%)
Distance	[1 1 1 1 1 1 1 1 1 1]	[[9 0] [1 0]]	90
Modulation code rate encoded data	[1 1 1 0 0 1 1 1 0 1]	[[2 0] [0 8]]	100
Channel Noise	[1 0 1 0 1 1 1 0 1 1]	[[6 0] [0 4]]	100
SNR (50 users)	[1 0 0 1 0 1 1 1 1 1]	[[3 0] [0 7]]	100
SNR (20 users)	[1 1 1 0]	[[2 0] [0 2]]	95

TABLE II. DECISION TREE ALGORITHM RESULTS USING ENTROPY

Presence of user Vs	Parameters		
	Predicted values	Confusion Matrix	Accuracy (%)
Distance	[1 1 1 1 1 1 1 0 1 1]	[[0 2] [0 8]]	80
Modulation code rate encoded data	[1 1 1 0 0 1 1 1 0 1]	[[2 0] [1 7]]	90
Channel Noise	[1 0 1 0 1 1 1 0 1 1]	[[6 1] [0 3]]	90
SNR (50 users)	[1 0 0 1 0 1 1 1 1 1]	[[5 0] [1 4]]	90
SNR (20 users)	[1 1 1 0]	[[0 0] [1 3]]	75

Table III summarizes the DT model's ability to jointly predict modulation type and code rate using individual channel features such as SNR, distance, and channel noise. The model achieved 94.2% accuracy when predicting MCS from SNR, 100% accuracy from distance, and 96% accuracy from channel noise. The high accuracy values indicate that each of these parameters has a strong correlation with the optimal MCS, although distance appears to be the most reliable standalone predictor in this dataset. Table 4 presents a representative excerpt from the dataset, showing the recorded parameters for 10 users in the experimental setup. For each user entry, the table lists the assigned MCS, the corresponding label-encoded MCS value, the distance from the base station (in kilometers), the measured channel noise (in dBm), the SNR (in dB), and a binary indicator for the presence of the user. The label encoding maps each MCS configuration—such as 256 QAM-7/8, 128 QAM-5/6, and BPSK-1/2 to an integer value for compatibility with the Decision Tree training process.

The Table IV demonstrates clear trends consistent with link adaptation principles. For example, users closer to the base station with higher SNR values are assigned higher-order modulations and higher code rates (256 QAM-7/8), while those farther away or in noisier conditions receive lower-order, more robust MCS configurations (128 QAM-3/4). The inclusion of the presence-of-user flag enables additional classification scenarios, such as determining whether a user is actively connected under given channel conditions.

TABLE III. ESTIMATION OF MODULATION AND CODE RATE USING SNR, DISTANCE AND CHANNEL NOISE

X-Axis, Y-Axis	Prediction		Accuracy (%)
	X_test	X_train	
X= SNR;	[6, 8, 20, 31, 12, 0, 34, 1, 28, 19]	40, 1, 23, 0, 10, 16, 33, 15, 22, 38, 0, 11, 30, 27, 5, 35, 4, 29, 13, 0, 37, 0, 21, 26, 9, 0, 24, 0, 17, 18, 14, 7, 36, 3, 0, 31, 0, 0, 1, 39]	94.2
Y=Modulation code rate	Decoded Labels: 16 QAM-1/2, 16 QAM-3/4, 32 QAM-7/8, BPSK-1/2, 256 QAM-2/3, 0, BPSK-5/6, 128 QAM-1/2, 8 QAM-3/4, 32 QAM-5/6]		
X= Distance;	[6, 28, 0, 23, 37, 32, 5, 25, 4, 33]	[20, 14, 0, 0, 39, 19, 3, 27, 0, 0, 18, 12, 8, 0, 0, 9, 34, 7, 11, 30, 15, 17, 31, 2, 0, 24, 16, 26, 0, 0, 21, 40, 22, 36, 1, 13, 35, 38, 29, 10]	100
Y=Modulation code rate	Decoded Labels: [16 QAM-1/2, 8 QAM-3/4, 0, 64 QAM-3/4, QPSK-2/3, BPSK-2/3, 128 QAM-7/8, 64 QAM-7/8, 128 QAM-5/6, BPSK-3/4]		
X= Channel noise;	[0, 4, 31, 2, 18, 0, 0, 30, 1, 16]	[26, 29, 9, 39, 17, 15, 36, 0, 8, 0, 0, 20, 32, 39, 0, 11, 28, 34, 5, 0, 12, 0, 7, 37, 19, 14, 24, 3, 0, 13, 21, 10, 33, 23, 27, 38, 1, 35, 6, 22]	96
Y=Modulation code rate	Decoded Labels: [0, 128 QAM-5/6, BPSK-1/2, 128 QAM-2/3, 32 QAM-3/4, 0, 0, 8 QAM-7/8, 128 QAM-1/2, 32 QAM-1/2]		

TABLE IV. SAMPLE DATASET CONSIDERED FOR 10 USERS

Modulation and code rate	Modulation and code rate label encoder	Distance	Channel Noise	SNR	Presence of user
256 QAM-7/8	15	1	6	31	0
256 QAM-5/6	14	2	9	29	1
0	0	3	0	0	0
256 QAM-3/4	13	4	11	28.5	1
256 QAM-2/3	12	5	13	28	0
256 QAM-1/2	11	6	14	27	1
128 QAM-7/8	5	7	15	26	1
128 QAM-5/6	4	8	18	25	0
0	0	9	0	0	0
128 QAM-3/4	3	10	20	24	1

The results confirm that DT classifiers, particularly when using the Gini Index criterion, are effective for real-time AMC prediction in wireless systems. The high accuracy across multiple channel features demonstrates that the model can reliably map physical-layer measurements to optimal MCS configurations. Furthermore, the interpretability of the decision paths makes the approach suitable for deployment in SDR-based systems, where transparency and computational efficiency are essential. However, while the model performs well with single-feature inputs, real-world deployments may benefit from combining multiple features to improve robustness against measurement noise and environmental variability. This approach, though, may increase the tree depth and inference latency, introducing a trade-off between accuracy and computational overhead. Finally, these findings highlight the practicality of using Decision Trees for MCS prediction in dynamic wireless environments. The proposed method not only offers high classification performance but also meets the real-time and low-complexity requirements necessary for integration into operational communication systems.

V. CONCLUSION

This paper presented the experimental validation of adaptive modulation and coding prediction using a decision tree algorithm, trained on real-time datasets generated through a software defined radio platform using the Universal Software Radio Peripheral N210 and the GNU Radio framework. The predictive model used key channel parameters, namely signal-to-noise ratio, channel noise, and the distance of the user from the base station, to determine both the modulation type and the coding rate. Two splitting criteria, the Gini Index and Entropy, were compared, and the results showed that the Gini Index consistently produced higher classification accuracy, reaching perfect prediction in several cases. It was also observed that larger datasets, such as those collected with fifty users, improved performance, highlighting the importance of diverse training data. The findings demonstrate that decision tree models are interpretable, computationally efficient, and effective for real-time selection of modulation and coding schemes in dynamic wireless environments. Future developments could include the use of ensemble learning methods to enhance robustness, the incorporation of additional channel features such as Doppler shift and interference levels to improve prediction reliability, the adoption of online learning to adapt to changing conditions, and the deployment of the trained model in a live software defined radio testbed for validation under operational scenarios.

REFERENCES

[1] Peng, Shengliang, Shujun Sun, and Yu-Dong Yao. "A survey of modulation classification using deep learning: Signal representation and data preprocessing." *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[2] Ramjee, Sharan, et al. "Fast deep learning for automatic modulation classification." *arXiv preprint arXiv:1901.05850* (2019).

[3] Venkata Subbarao, M., Keerthana, B., Ramesh Varma, D., Terlapu, S.K., Challa Ram, G. (2023). Automatic Modulation Classification Under AWGN and Fading Channels Using Convolutional Neural Network. In: Chakravarthy, V., Bhateja, V., Flores Fuentes, W., Anguera, J., Vasavi, K.P. (eds) Advances in Signal Processing, Embedded Systems and IoT. Lecture Notes in Electrical Engineering, vol 992. Springer, Singapore. https://doi.org/10.1007/978-981-19-8865-3_20.

[4] Peng, Shengliang, Shujun Sun, and Yu-Dong Yao. "A survey of modulation classification using deep learning: Signal representation and data preprocessing." *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[5] M. V. Subbarao and P. Samundiswary, “K-nearest neighbors based automatic modulation classifier for next generation adaptive radio systems,” *International Journal of Security and Its Applications (IJSIA)*, vol. 13, no. 4, pp. 41–50, 2019.

[6] Ghasemzadeh, Pejman, et al. "Analysis of distribution test-based and feature-based approaches toward automatic modulation classification." *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2019.

[7] Wu, Peng, et al. "Automatic modulation classification based on deep learning for software-defined radio." *Mathematical Problems in Engineering* 2020 (2020).

[8] M. V. Subbarao and P. Samundiswary, “Performance Analysis of Automatic Modulation Classification Using Time Frequency Transforms Under Non-Ideal Channel Conditions,” Int. J. Innov. Technol. Exploring Eng., vol. 8, no. 12, pp. 1685–1691, Oct. 2019.

[9] M. V. Subbarao and P. Samundiswary, “Spectrum sensing in cognitive radio networks using time–frequency analysis and modulation recognition,” in Microelectronics, Electromagnetics and Telecommunications, J. Anguera, S. Satapathy, V. Bhateja, and K. Sunitha, Eds., Lecture Notes in Electrical Engineering, vol. 471. Singapore: Springer, 2018, pp. 867–877. doi: 10.1007/978-981-10-7329-8_85.

[10] M. Abu-Romoh, A. Aboutaleb, and Z. Rezki, “Automatic modulation classification using moments and likelihood maximization,” *IEEE Communications Letters*, vol. 22, no. 5, pp. 938–941, 2018.

[11] T. O’Shea, T. James, T. Roy, and T. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, pp. 168–179, 2018.

[12] S. Hu, Y. Pei, and P. P. Liang, “Robust modulation classification under uncertain noise condition using recurrent neural network,” in *Proceedings of the IEEE Global Communications Conference*, Abu Dhabi, United Arab Emirates, December 2018.

[13] Venkata Subbarao, M., and P. Samundiswary. "Automatic modulation classification using cumulants and ensemble classifiers." In Advances in VLSI, Signal Processing, Power Electronics, IoT, Communication and Embedded Systems: Select Proceedings of VSPICE 2020, pp. 109-120. Springer Singapore, 2021.

[14] Charbuty, Bahzad, and Adnan Abdulazeez. "Classification based on decision tree algorithm for machine learning." *Journal of Applied Science and Technology Trends* 2.01 (2021): 20-28.

[15] Tangirala, Suryakanthi. "Evaluating the impact of GINI index and information gain on classification using decision tree classifier algorithm." *International Journal of Advanced Computer Science and Applications* 11.2 (2020): 612-619.

[16] RekhaMolala, “Entropy, Information gain and Gini Index; the crux of a Decision Tree,” Medium, Mar. 23, 2020. https://blog.clairvoyantsoft.com/entropy-information-gain-and-giniindex-the-crux-of-a-decision-tree-99d0cdc699f4.

[17] Subbarao, M. Venkata, and P. Samundiswary. "Automatic modulation recognition in cognitive radio receivers using multi-order cumulants and decision trees." Int. J. Rec. Technol. Eng.(IJRTE) 7 (2018): 61-69.

[18] M. V. Subbarao and P. Samundiswary, “Performance Analysis of Modulation Recognition in Multipath Fading Channels Using Pattern Recognition Classifiers,” Wireless Pers. Commun., 2020, doi: 10.1007/s11277-020-07564-z.

[19] P. G. Varna Kumar Reddy and M. Meena, “Adaptive modulation classification with deep learning for various number of users and performance validation,” in *Third Congress on Intelligent Systems (CIS 2022)*, S. Kumar, H. Sharma, K. Balachandran, J. H. Kim, and J. C. Bansal, Eds., Lecture Notes in Networks and Systems, vol. 613. Singapore: Springer, 2023, pp. 255–266. doi: 10.1007/978-981-19-9379-4_21.

[20] P. G. Varna Kumar Reddy and M. Meena, “Automatic modulation classification using a support vector machine-based pattern recognition algorithm,” *Ingénierie des Systèmes d'Information*, vol. 27, no. 6, pp. 1–8, 2022.