**AMC**

1. *Definizione obiettivi principali ✅*
   1. Creare un sistema AMC basato su AI per adattare dinamicamente la modulazione e la codifica in base alle condizioni del canale.
   2. Integrare il feedback tra TX e RX per ottimizzare le decisioni.
2. *Preparazione ambiente di sviluppo*
   1. Configurare le due Pluto come RX e TX
   2. Implementare nel codice il messaggio di feedback.
   3. Crea un dataset di test raccogliendo dati di trasmissione e ricezione con diverse configurazioni di modulazione e codifica (es. BPSK, QPSK, 16-QAM + vari livelli di codifica).
3. *Sviluppo del modello AI*
   1. Generative network (TX)
      1. Generare configurazioni di modulazione e codifica adattive.
      2. INPUT: metriche storiche (feedback RX). All’inizio il tx assume un canale da definire
      3. OUTPUT: tipo di modulazione, rate di codifica, intervallo temporale di aggiornamento.
   2. Discriminative Network (RX)
      1. Valutare le configurazioni ricevute e fornire un feedback.
      2. INPUT: BER, SNR, throughput, CRC decoding (header e data)
      3. OUTPUT: comunicazione ottimale (1) o subottimale (0)
4. *Addestramento dei modelli*
   1. Utilizzare il dataset costruito al punto 2
   2. Aggiorna il Generator sul feedback del Discriminator
5. *Implementazione del feedback loop (simulato)*
   1. Trasmissione:
      1. Il TX genera una configurazione di modulazione/codifica basata sul modello generativo.
      2. La configurazione è trasmessa nel *frame header* per il riconoscimento al RX.
   2. Ricezione:
      1. Il RX analizza la trasmissione e calcola le metriche (SNR, BER, CRC).
      2. Il feedback è inviato al TX via un canale di controllo dedicato (o simulato).
6. *Test*
   1. Usare le Pluto per testare il sistema
   2. Fine-tuning: addestramento real-time per migliorare le decisioni
7. *Ottimizzazione*
   1. Riduzione della latenza del loop
   2. Reinforcement learning lato RX
   3. Aggiunta di ulteriori metriche per classificare il canale
8. *Documentazione e demo*