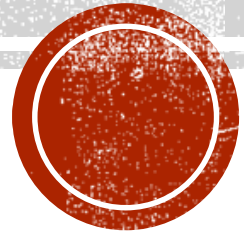
Artificial Intelligence (ECSE 526) Final Project

McGill UNIVERSITY

BLIND DIGITAL MODULATION IDENTIFICATION USING NEURAL NETWORKS & HIGHER ORDER STATISTICS

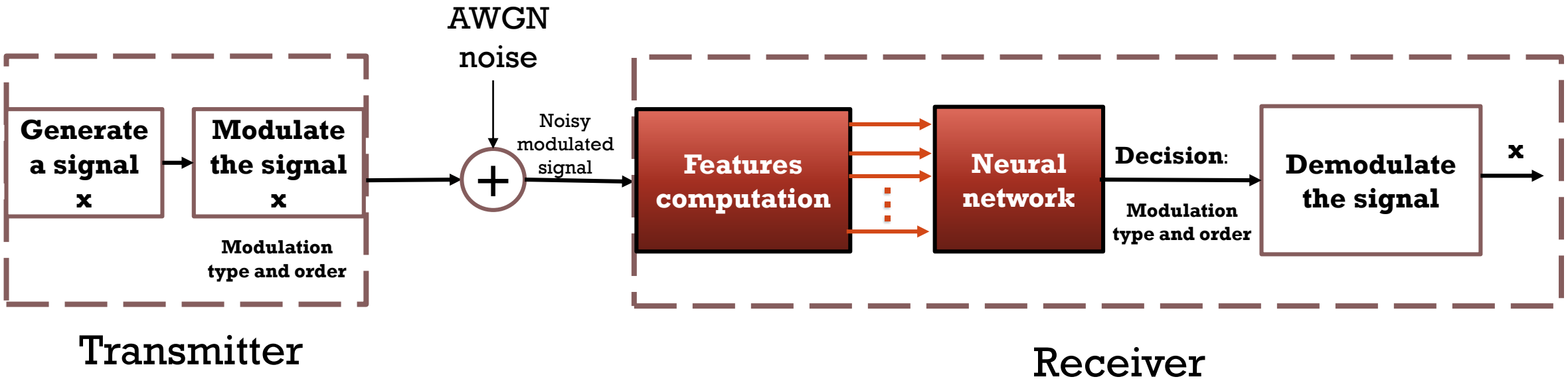


Amen Memmi

INTRODUCTION

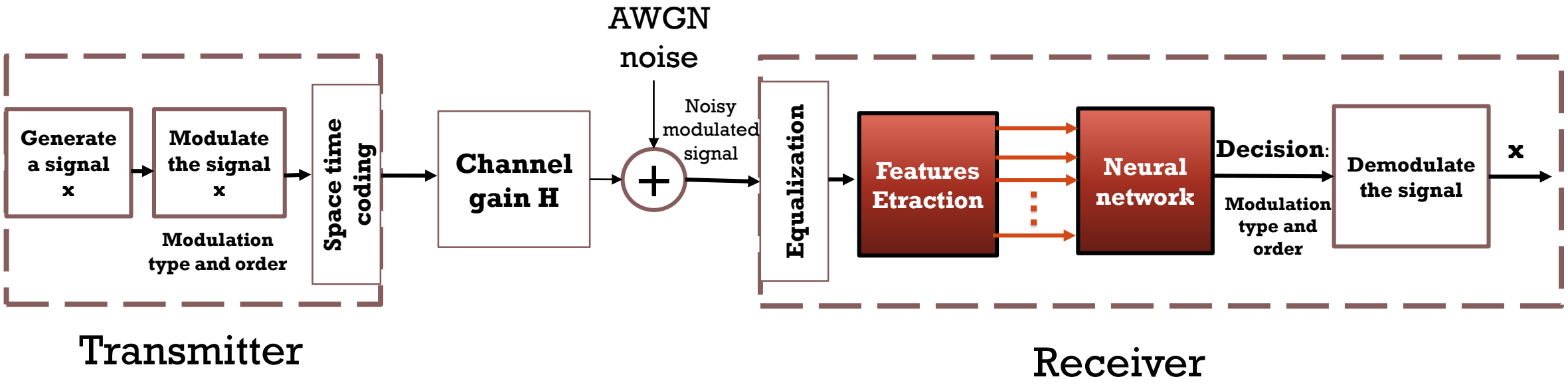
- Blind digital modulation identification for 2*2 MIMO systems: to discriminate among different M-ary shift keying linear modulation schemes without any priori signal information
- Applications: cognitive radios, military surveillance…
- Neural networks NN (Pattern recognition NN) are used for classification digital modulations
- Used features for the NN: higher order statistical moments and cumulants of the received signal
- Matlab: Digital communication & neural networks toolboxes

SYSTEM MODEL



Simplified scheme of the communication system

SYSTEM MODEL



Scheme of the communication system

FEATURES

- Many recent works have shown that one of the best candidates for signal identification are higher order moments and cumulants of the received signal.
- Moments: $M_{km}(x) = E[x^k (x^*)^{k-m}]$

$$\approx \frac{1}{N} \sum_{p=1}^{n} x^{k-m}(i)\ (x^*(i))^m$$

- Cumulants:

$$C_{km}(x) = Cum\left[\underbrace{x, \ldots, x}_{(k-m)times}\ \underbrace{x^*, \ldots, x^*}_{mtimes}\right]$$

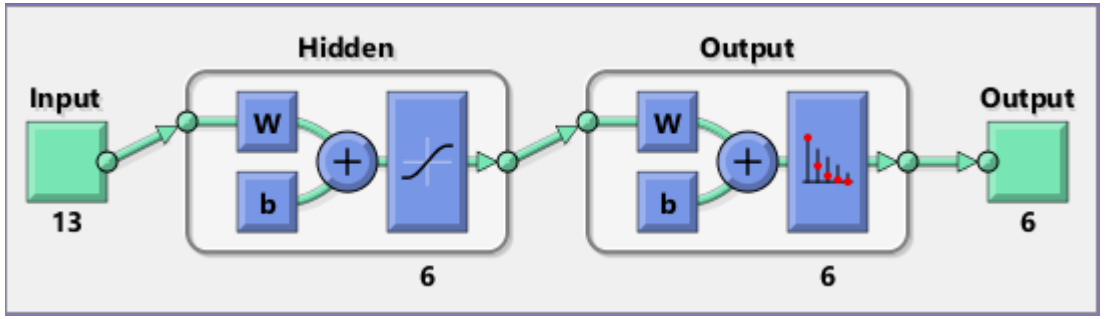
$$Cum[x, y, z, w] = E(xyzw) - E(xy)E(zw) - E(xz)E(yw) - E(xw)E(yz)$$

- Closed form expressions

TABLE I
SOME THEORETICAL STATISTICAL MOMENTS AND CUMULANTS VALUES FOR DIFFERENT MODULATION SCHEMES OF INTEREST [1], [5], [7]

	2-PSK	4-PSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM
C20	1	0	0	1	1	0	0
M40	1	1	0	1.64	1.77	-0.67	-0.18
M41	1	0	0	1.64	1.77	0	0
M42	1	1	1	1.64	1.77	1.32	1.34
C40	-2	1	0	-1.36	-1.24	-0.68	-0.62
C41	-2	0	0	-1.36	-1.24	0	0
C42	-2	-1	-1	-1.36	-1.24	-0.68	-0.62
M60	1	0	0	2.92	3.62	0	0
M61	1	-1	0	2.92	3.62	-1.32	0.38
M63	1	1	1	2.92	3.62	1.96	2.08
C60	16	0	0	8.32	7.19	0	0
C61	16	-4	0	8.32	7.19	2.08	1.8
C62	16	0	0	8.32	7.19	0	0
C63	16	4	4	8.32	7.19	2.08	1.8

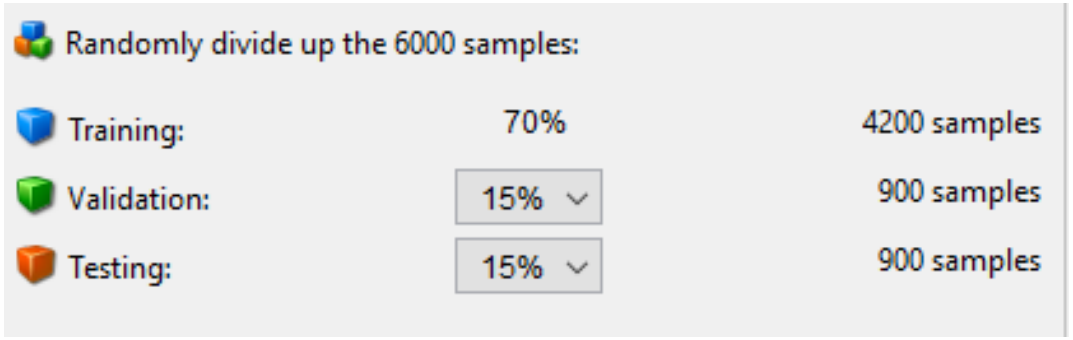
NEURAL NETWORK DIAGRAM



- Feedforward NN – Backpropagation scaled conjugate gradient
- 13 Input features
- 6 Hidden layers + 6 Output layers
- 6 Outputs or 6 classes: 2-PSK, 4-PSK, 8-PSK, 8-QAM, 16-QAM, 64-QAM
  Optimally 1 for the right class and 0 for others, but when decision is not fully sure, values between 0 and 1 and the maximum is the most probable class

NEURAL NETWORK TRAINING

- **Example**:
- Low noise: SNR = 15 dB (Useful signal ≈ 30 times Noise)
- 13*(1000*6) input training matrix: 13 input features * 1000 samples for each of the 6 class

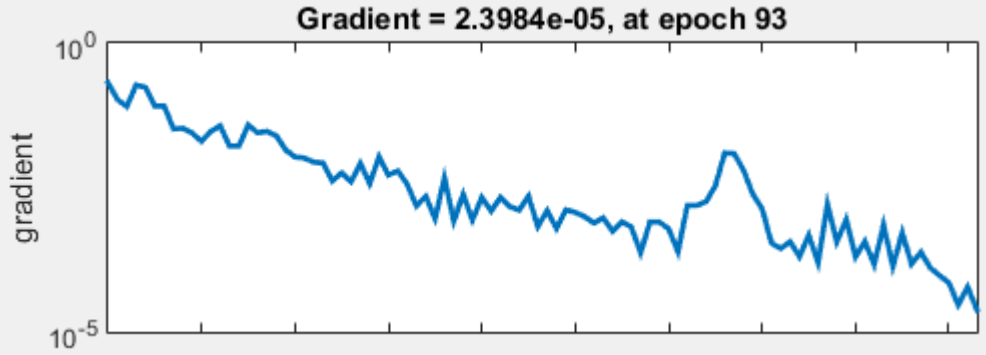


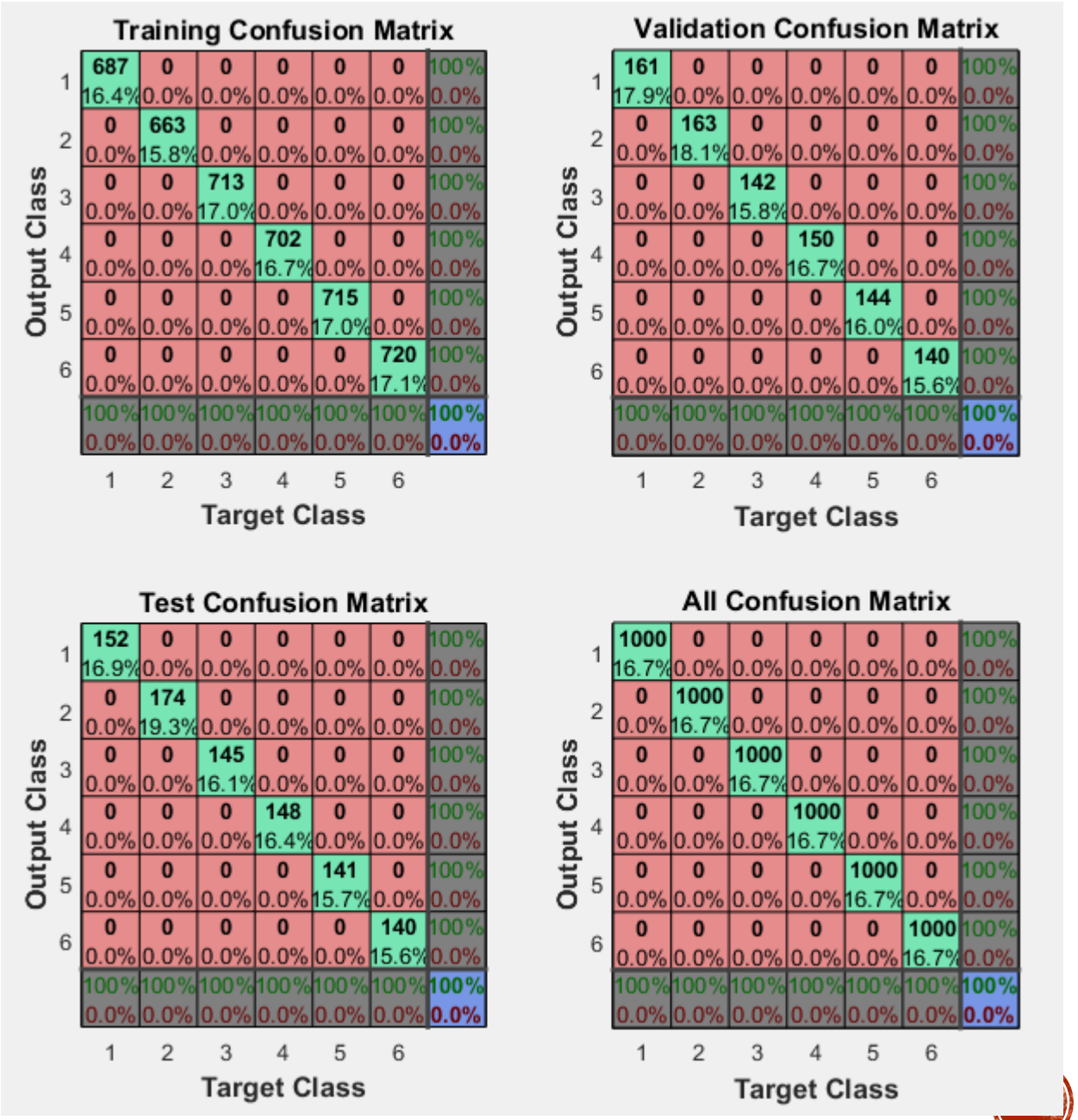
Algorithms

Data Division:	Random	(dividerand)
Training:	Scaled Conjugate Gradient	(trainscg)
Performance:	Cross-Entropy	(crossentropy)
Calculations:	MEX	

Progress

Epoch:	0	93 iterations	1000
Time:		0:00:01	
Performance:	0.462	1.95e-05	0.00
Gradient:	0.204	2.40e-05	1.00e-06
Validation Checks:	0	6	6

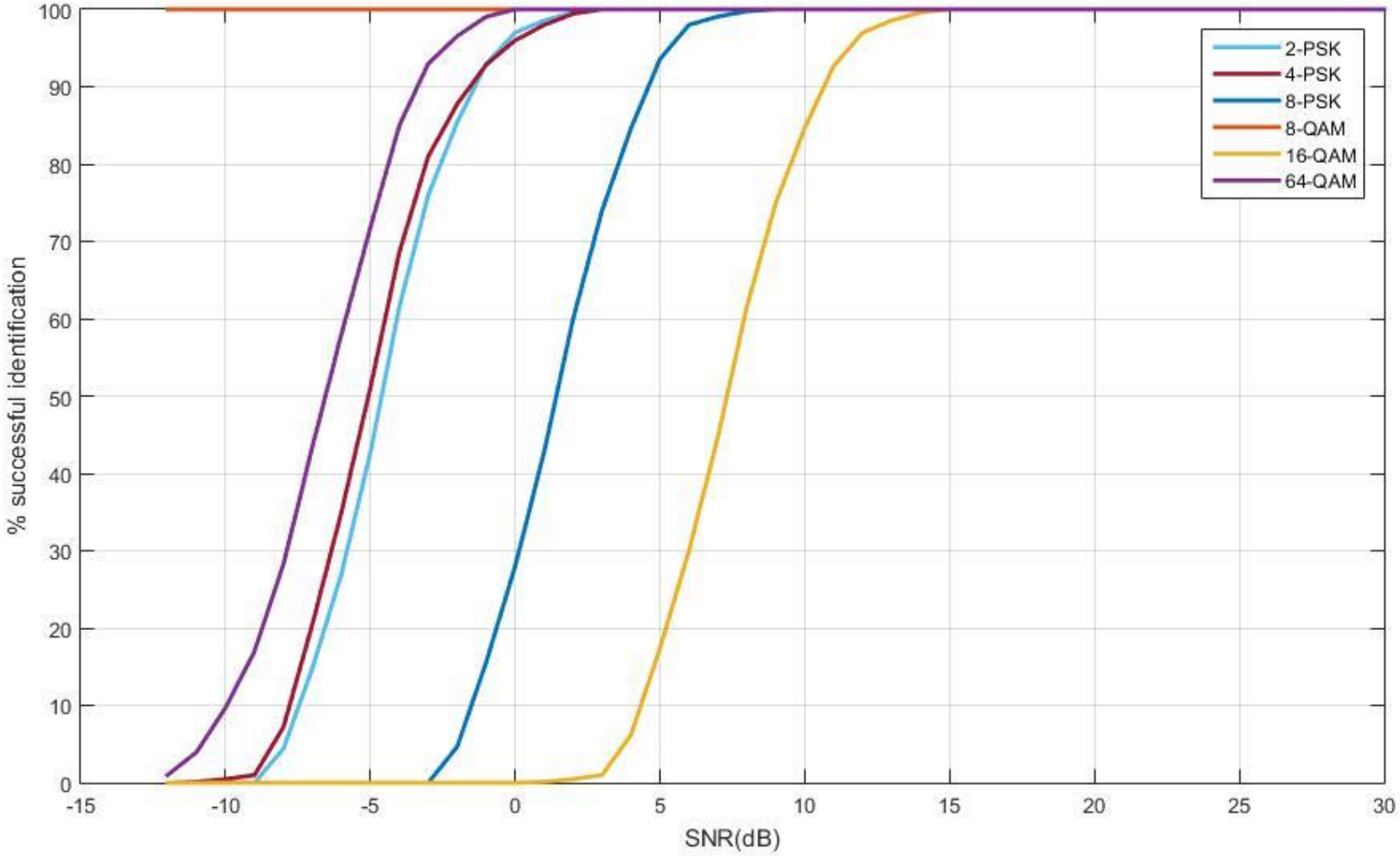




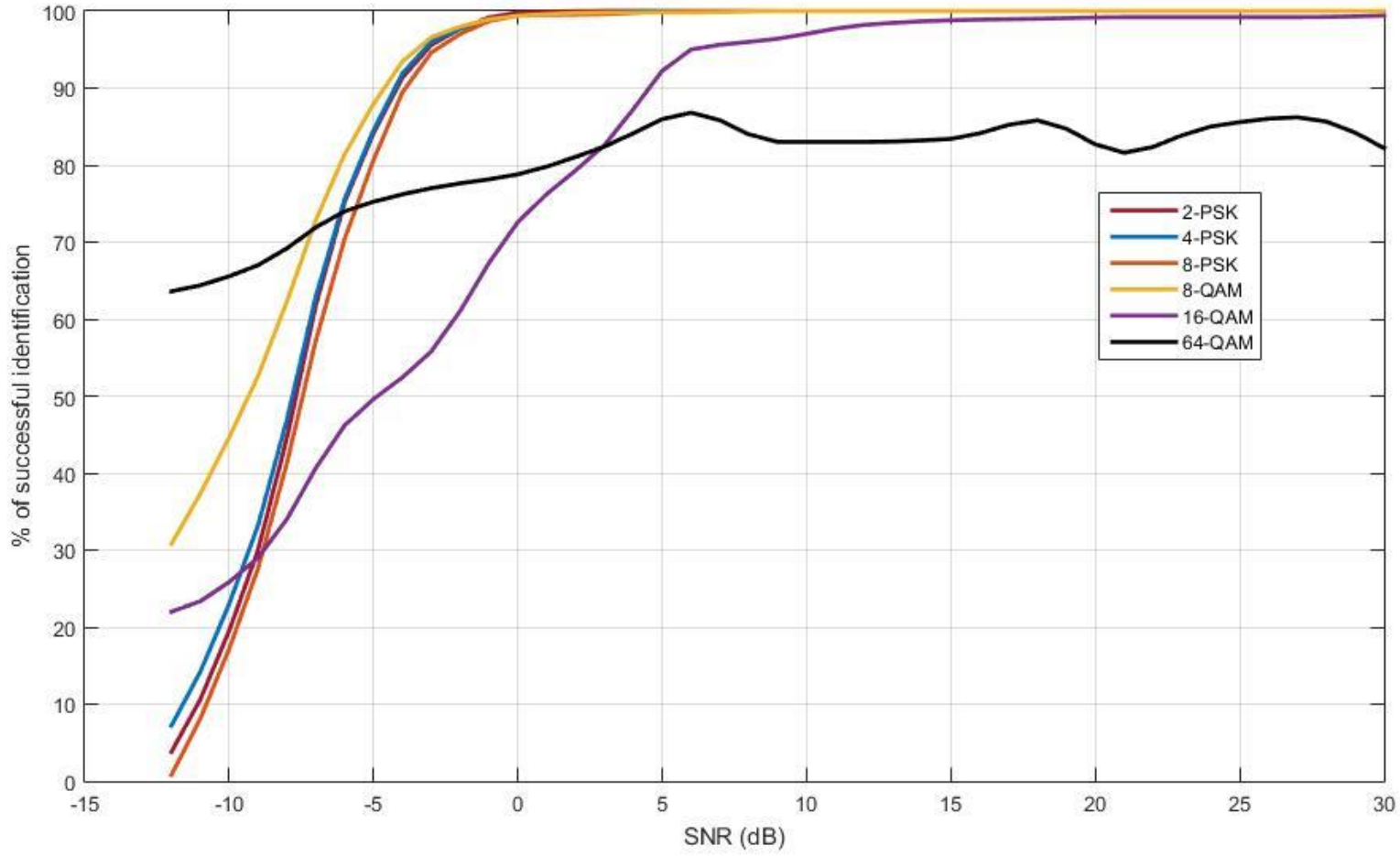
MONTE-CARLO SIMULATION $N=10^5$ SAMPLES FOR THE SAME SNR=15 dB

Modulation	2-PSK	4-PSK	8-PSK	8-QAM	16-QAM	64-QAM
% successful identification	100	100	100	100	98,63	93,45

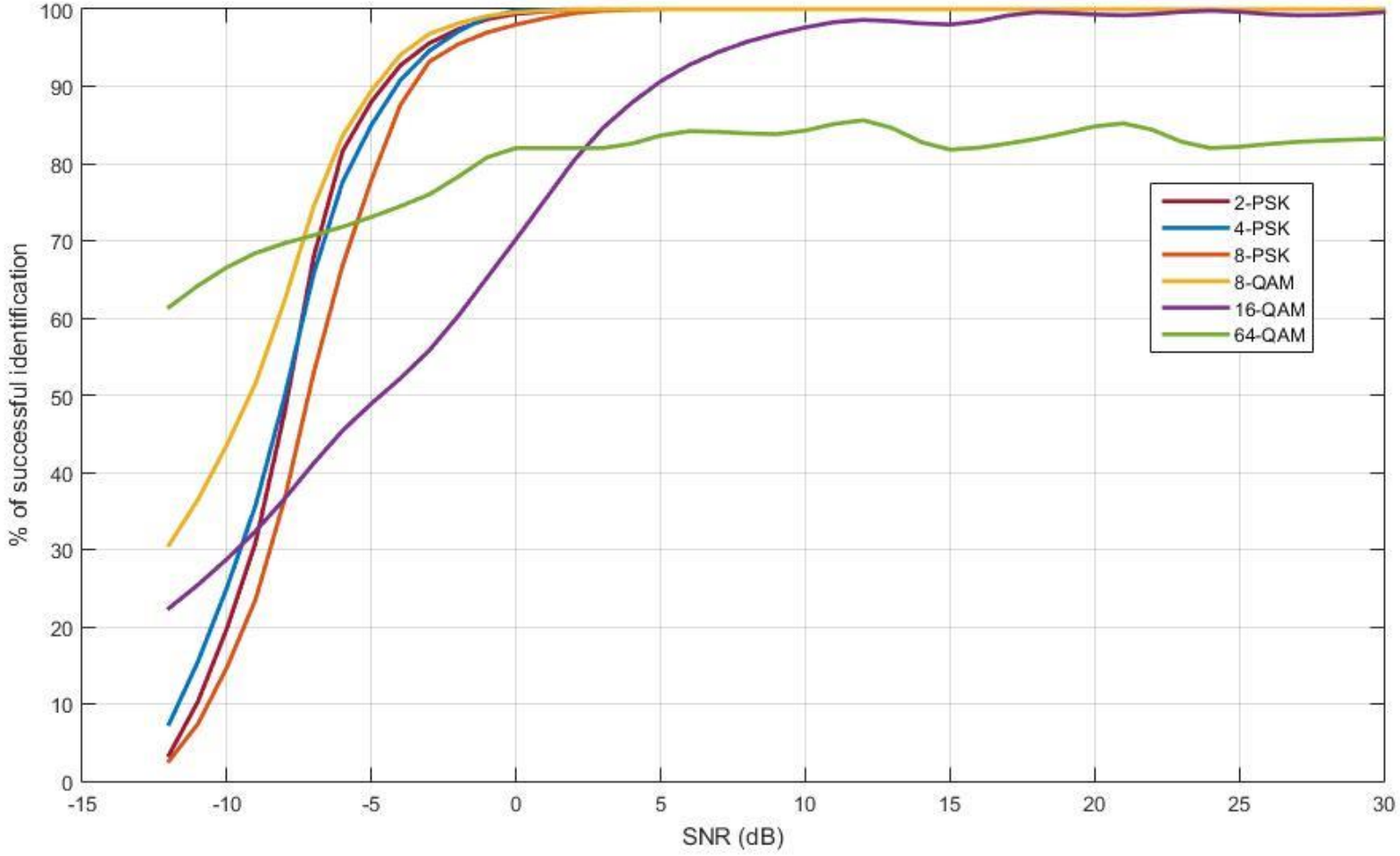
SNR EFFECT (NN TRAINED AT 15dB)



NN TRAINED AT 3dB



NN TRAINED AT -9dB

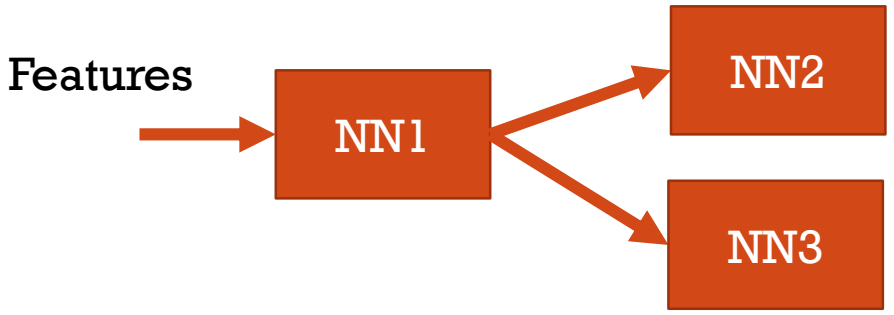


IDEAS

- Performance affected by SNR in training and in identification

→ idea: consider several NNs:

- ✓ 2 NN depending on the SNR: High and low SNR
- ✓ 3 NN: one, NN1, to detect modulation type as a first step, then use one of the two others NN2/NN3 to detect the order of the modulation



NN CASCADE TRAINED AT 3dB