DNN Assisted Sphere Decoder

Ghaya Rekaya-Ben Othman

Joint work with Aymen Askri

Novembre 2020

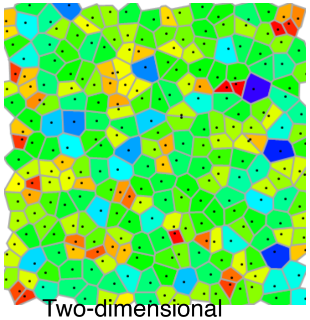
TELECOM Paris
IP PARIS

Introduction

- Lattices have many significant applications in pure mathematics
  - Lie algebras
  - Number theory
  - Group theory

- They also arise in applied mathematics in connection with :
  - Coding theory
  - Cryptography

- A classical problem in lattices: **Counting lattice points in n-dimensional sphere**



Two-dimensional random lattice (16x16)

Still unsolven even today !

Outline

- Overview of Lattices
- Deep Neural Network to Count lattice points in a sphere
- Deep Neural Network assisted Sphere Decoder
- Conclusion

Overview of lattices

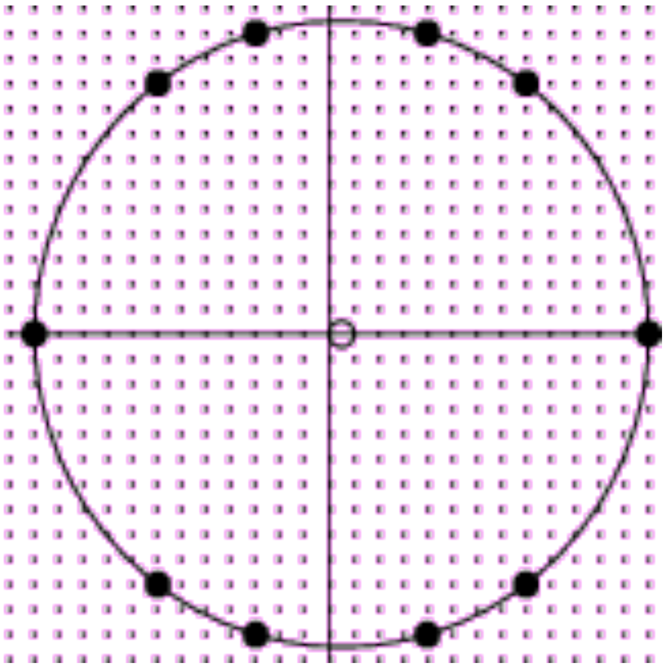
- A lattice Λ in $\mathbb{R}^n$ consists of all integral combinations of a set of linearly independent vectors $(b_1, \ldots, b_n)$ in $\mathbb{R}^n$ called a lattice basis B :

$$\Lambda = \{z_1 b_1 + \cdots + z_n b_n \mid z_1 \ldots z_n \in \mathbb{Z}\}$$

- Let B_r denote the n-dimensional Euclidean ball of radius r centered at the origin.

- Counting lattice points inside B_r is an NP-Hard problem

$$N = \#\{x \in \Lambda \mid \; \| x \| \; < r\}$$



$\mathbb{Z}^2$ Lattice

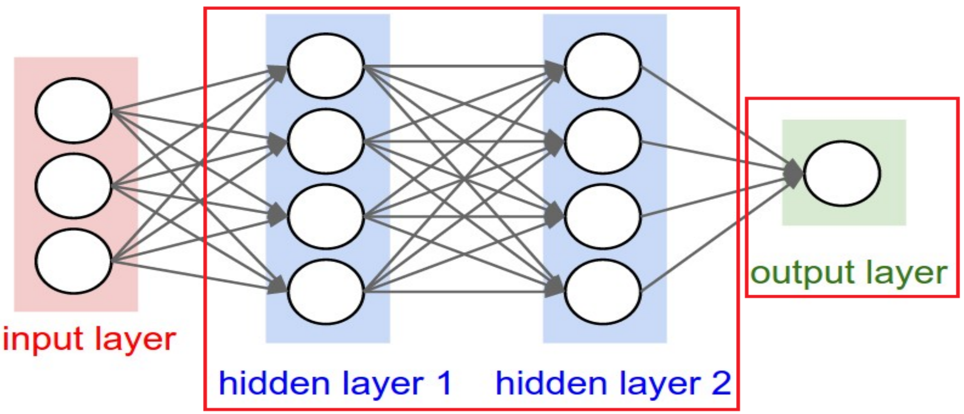
Solutions ?

- **Some existing solutions:**
  - Sphere Decoder which uses the Depth-First tree search strategy
  - Stack decoder which uses the Best-First tree search strategy
- **Problem:** The complexity of the search tree phase increases exponentially as a function of lattice dimension.
- **Contribution: Learn the number of lattice points in the n-dimensional sphere with fixed radius r centered at the origin using Deep Neural Networks (DNNs)**

Deep Neural Network (DNN) Architecture

Deep Neural Network (DNN) architecture

$x^{(i)} = \left(\Lambda^{(i)}, r^{(i)}, N^{(i)}\right)$

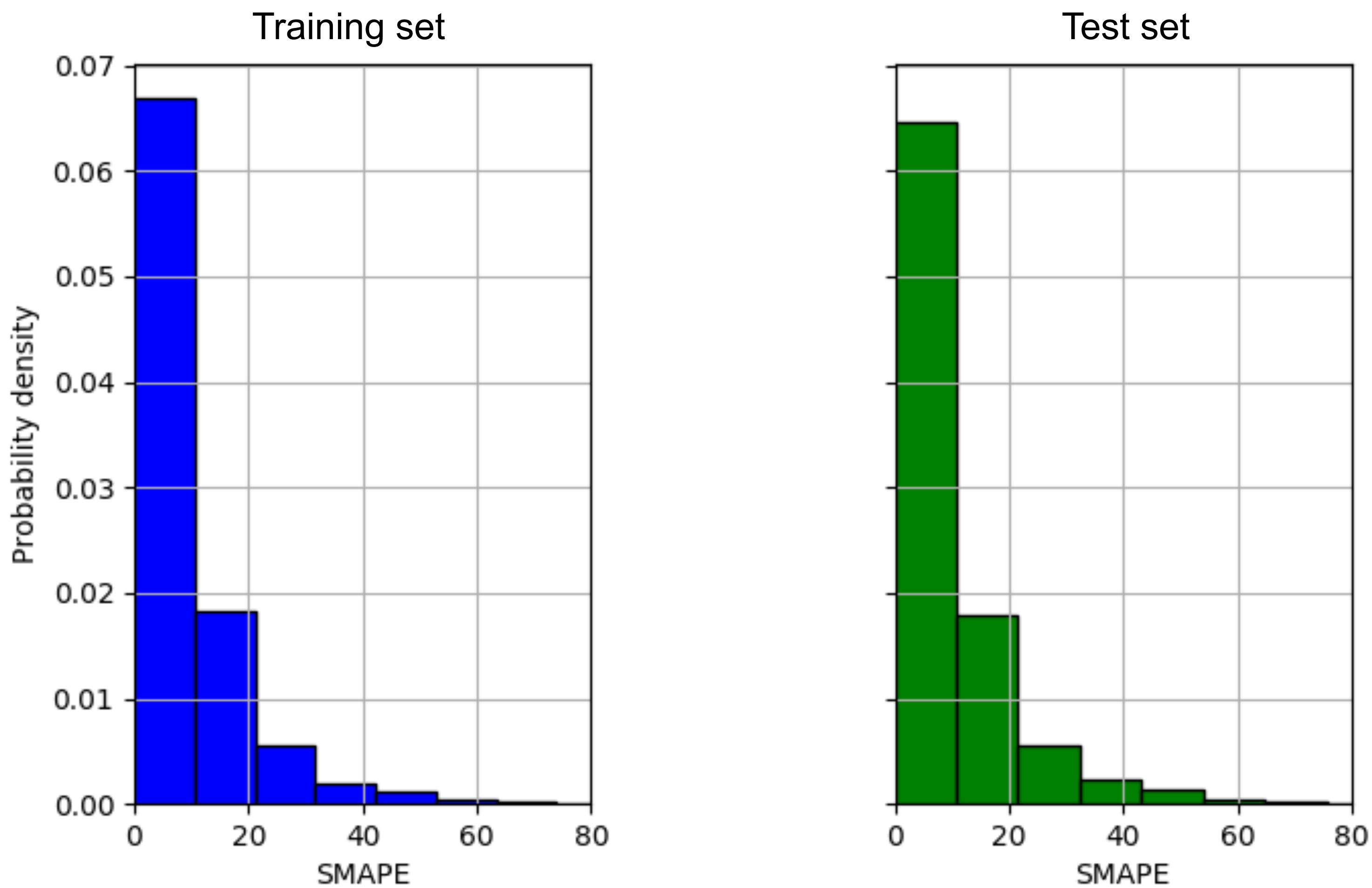


$\hat{N}^{(i)} = f\left(x^{(i)}, \theta\right)$

- The designed DNN is trained with independent input vectors given as $x^{(i)}$
- The loss is defined as the Symmetric Mean Absolute Percentage Error (SMAPE)

$$\text{SMAPE} = \frac{100\,\%}{S} \sum \left| \frac{N^{(i)} - \hat{N}^{(i)}}{N^{(i)} + \hat{N}^{(i)}} \right|$$

Results for arbitrary lattices of dimension $n = 10$



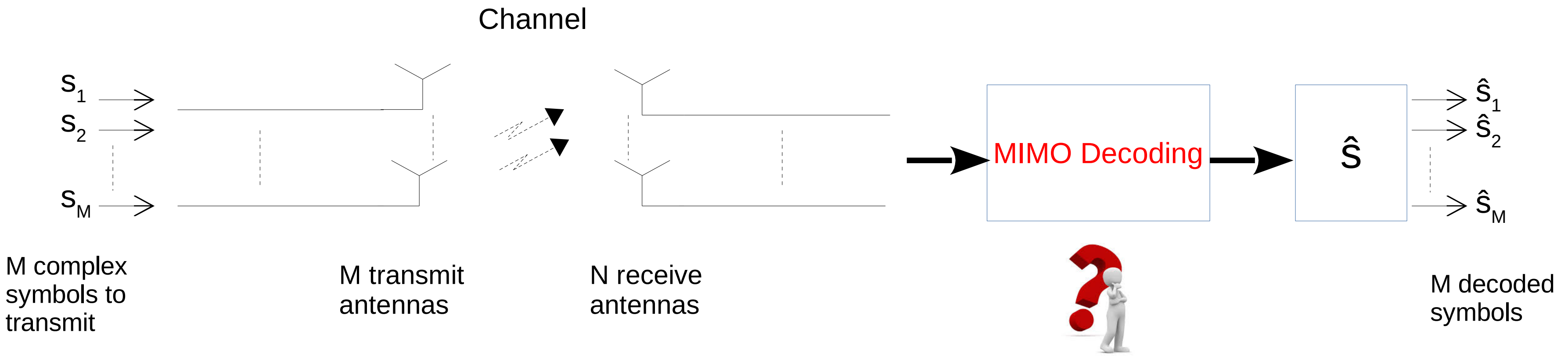
- High percentage of points whose SMAPE is below 10 %.

Results for some Known Lattices

Type	n	*Spherical Bound*	*Gaussian Bound*	**Measured** *Cardinality*	*Predicted Cardinality*	*SMAPE %*
A	5	≤ 24	26	7	6	7.69
	6	≤ 54	47	9	10	5.26
	7	≤ 140	99	13	12	4.00
	8	–	188	41	32	12.33
	9	–	391	69	64	3.76
	10	–	758	119	125	2.46
D	5	16	20	7	6	7.69
	6	≤ 37	42	9	10	5.26
	7	≤ 88	88	11	11	0.00
	8	240	183	77	62	10.79
	9	–	595	103	88	7.85
	10	–	1211	133	130	1.14
E	8	16	77	17	12	17.24

- Reference] Annika Meyer. On the number of lattice points in a small sphere. In WCC, 2011.

- Use of proposed DNN to count lattice points inside an arbitrary sphere to solve the Closest Vector Problem when used for MIMO decoding



$$y = H \cdot s + w$$

MIMO decoding

- Sub-optimal MIMO decoding schemes : low complexity but poor performance

- Optimal MIMO decoding: Maximum Likelihood (ML)

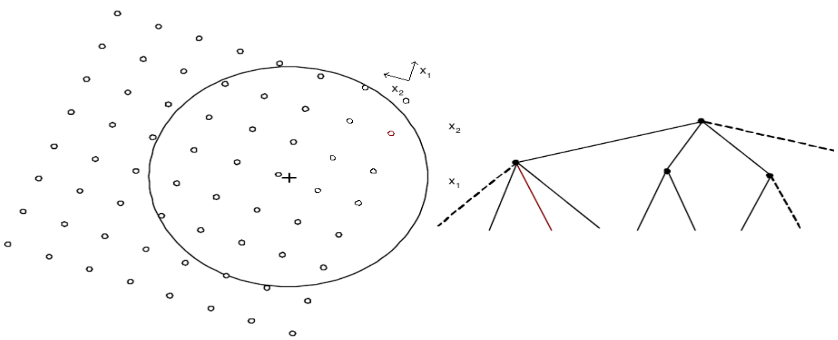
$$\hat{s}_{ML} = \arg(\mathbf{min} \parallel y - H \cdot s \parallel^2)$$

  - Example : Sphere Decoder finds the closest lattice points $\hat{s}_{ML}$ that lie in a sphere with some radius r centered on y.

  - The complexity of the tree search phase increases exponentially function of the number of transmit and receive antennas and the constellation size.

DNN Assisted Sphere Decoder

- **Idea: Find an initial search sphere radius for the SD algorithm using DNN**



For that, DNN learns the number of lattice points in the n-dimensional sphere, where :

➢ The radius is $r \leq 2n\sigma^2$,

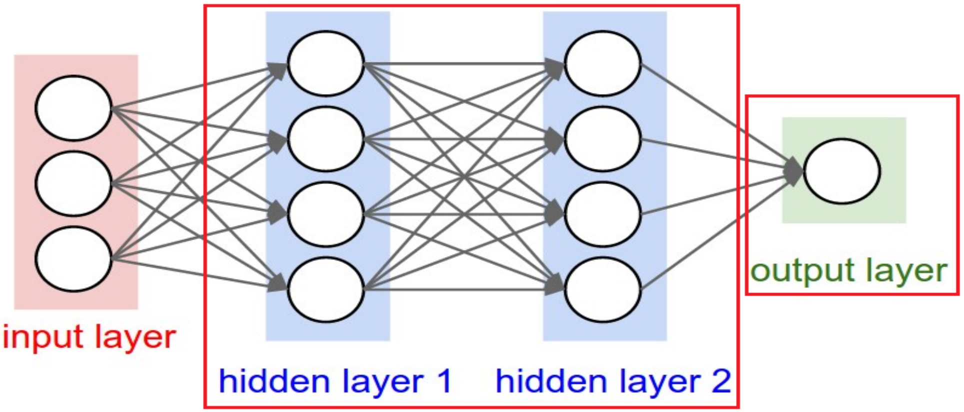
➢ The received signal y is the center,

➢ The constellation is finite.

DNN architecture

Deep Neural Network (DNN) architecture

$$x^{(i)} = \left(y^{(i)}, R^{(i)}, r^{(i)}, N^{(i)}\right)$$



$$\hat{N} = f\left(x^{(i)}, \theta\right)$$

- The designed DNN is trained with independent input vector $x^{(i)}$ received signal y, the generator matrix elements $R_{ij}, 1 \leq i \leq j < n$, the radius of the sphere r, and the measured number of lattice points $N^{(i)}$, create a set of training data.

- Training data is obtained via list sphere decoding implementations to obtain the exact number of lattice points $N^{(i)}$

- The loss is defined as the mean squared error :

$$L(\theta) = \frac{1}{|S|} \sum_{i \in S} \left(N^{(i)} - f\left(x^{(i)}; \theta\right)\right)^2$$

DNN Assisted Sphere Decoder

- We start by predicting the number of lattice points with an initial radius :

$$r^{(0)} = 2n\sigma^2 \Rightarrow N^{(0)}_{Predicted}$$

- If the predicted number is high, we update the radius by using a dichotomic strategy: we divide the radius by two and we recalculate the predicted number of lattice points using the DNN.

$$r^{(1)} = \frac{r^{(0)}}{\sqrt{2}} \Rightarrow N^{(1)}_{Predicted}$$

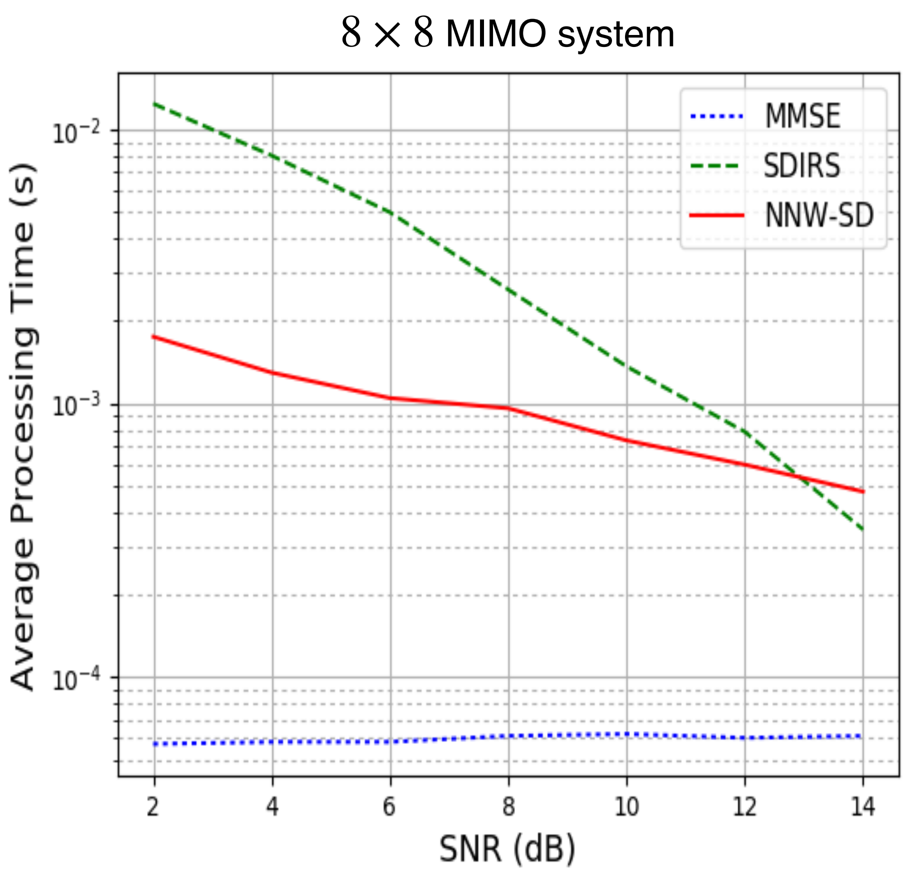
- We repeat the procedure until obtaining a number of lattice points equal or below a fixed threshold.

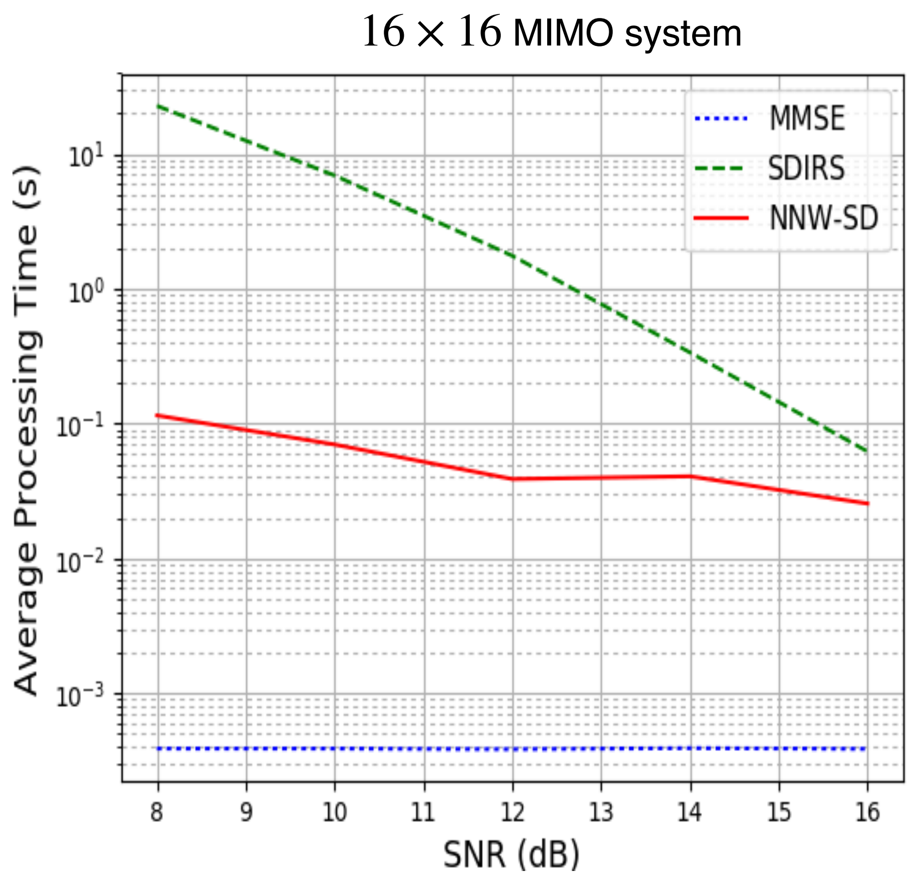
$$N^{(L)}_{Predicted} \leq N_{Threshold}$$

- SD search phase is started with a radius equal to :

$$r = \frac{r^{(0)}}{\sqrt{2^L}}$$

DNN assisted SD : simulations results

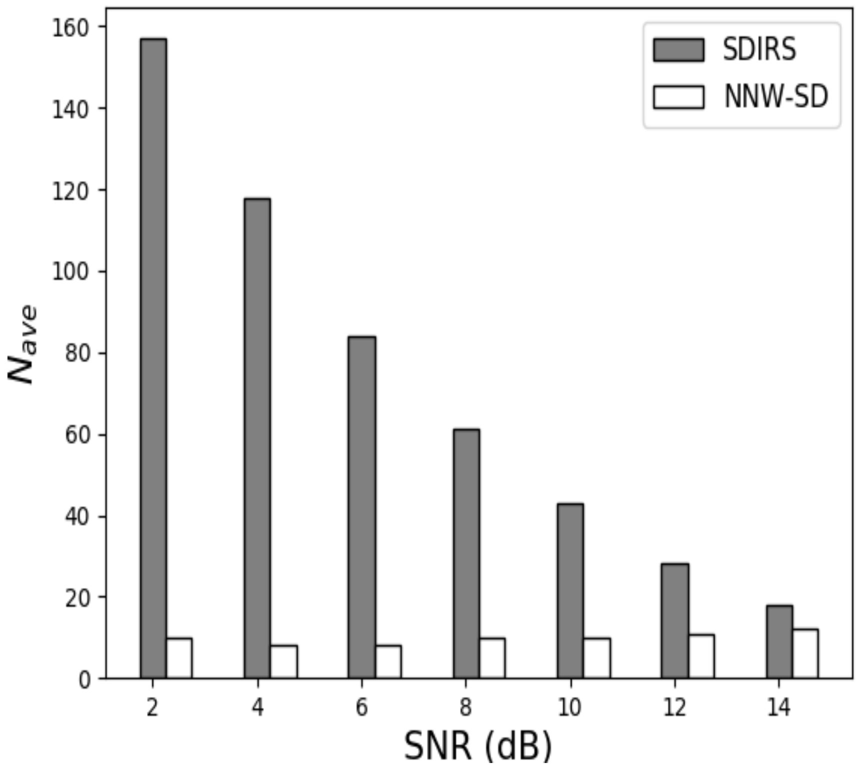




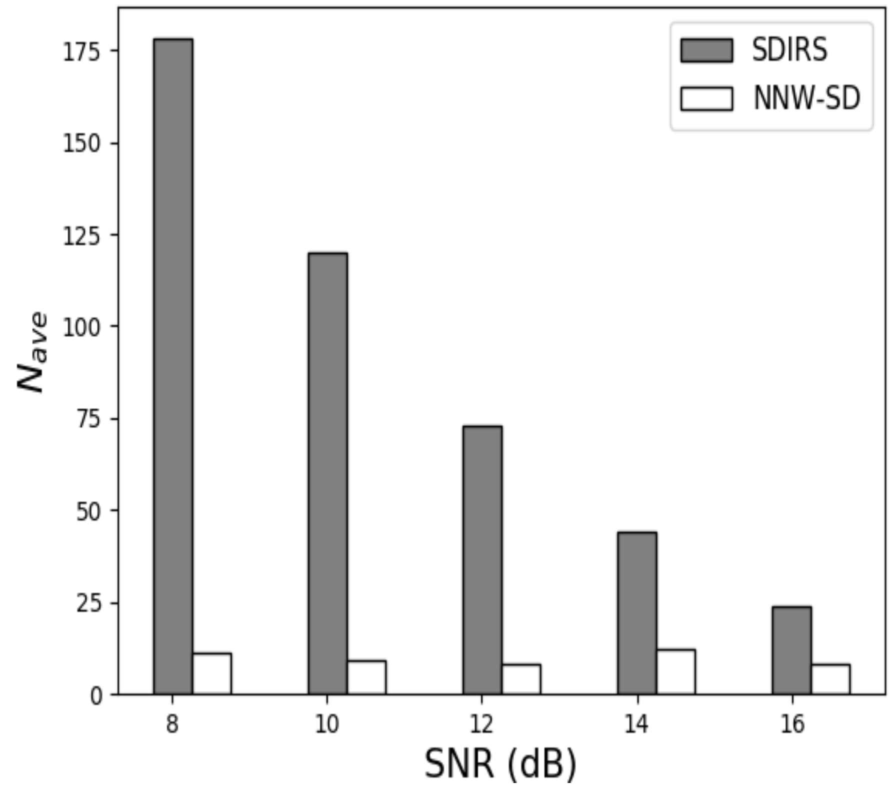
- For 16×16, at 12 dB SNR: Decoding time of the **DNN-SD** is **100** times that of the MMSE

DNN assisted SD : simulations results

8×8 MIMO system



16×16 MIMO system



- The average lattice points is constant for DNN-SD
- DNN SD gives the ML solution faster by examining much fewer number of lattice points.

Smart Sphere Decoder

- We propose to evaluate the average number of radius updates L
- We analyse theoretically this average number of radius updates function of SNR using a well-known counting function of lattice points in n-dimensional sphere :

$$N_p = \text{Card}\left\{ s \in \mathbb{Z}^n \mid \left\| y - H \cdot s \right\| \leq r \right\} \approx \frac{Vol\left(B_r\right)}{det\left(\Lambda\right)}$$

where $Vol\left(B_r\right) = r^n \cdot V_n$, V_n is the volume the volume of a sphere of a unit radius in $\mathbb{R}^n$, and $det\left(\Lambda\right)$ is the determinant of of the lattice Λ

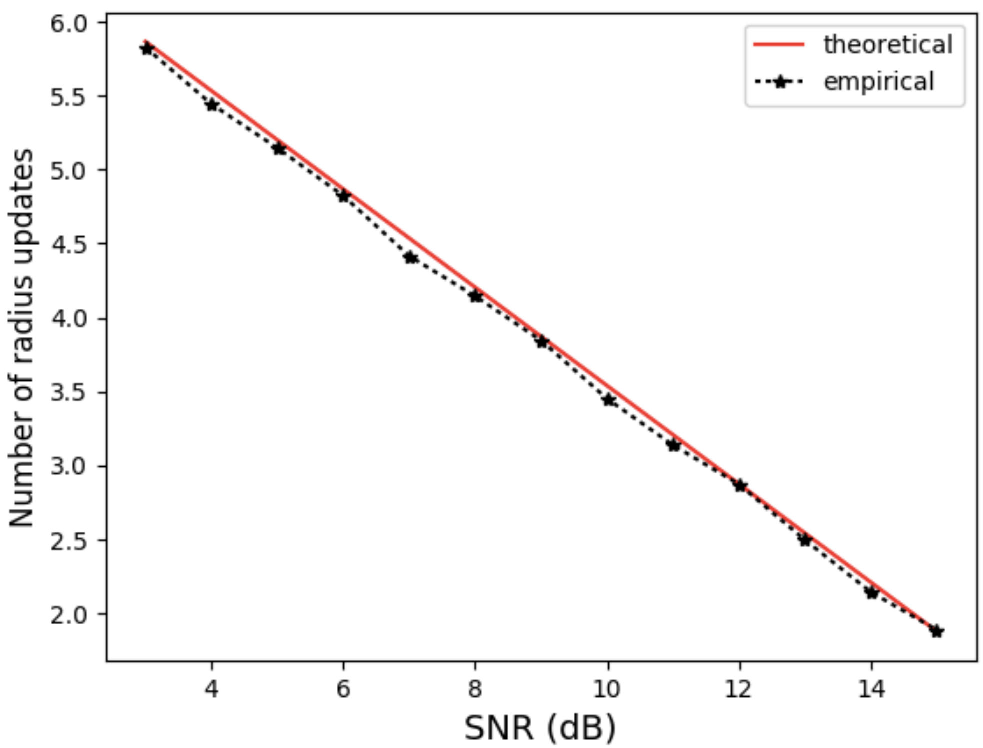
- If we set $r^{(0)} = 2n\sigma^2$, and consider L radius updates $r^{(L)} = r^{(0)}/\sqrt{2^L}$, we can prove that :

$$L = a \cdot \rho + b$$

where ρ is the SNR in dB, a and b are functions of the lattice parameters.

Smart SD : Simulation results

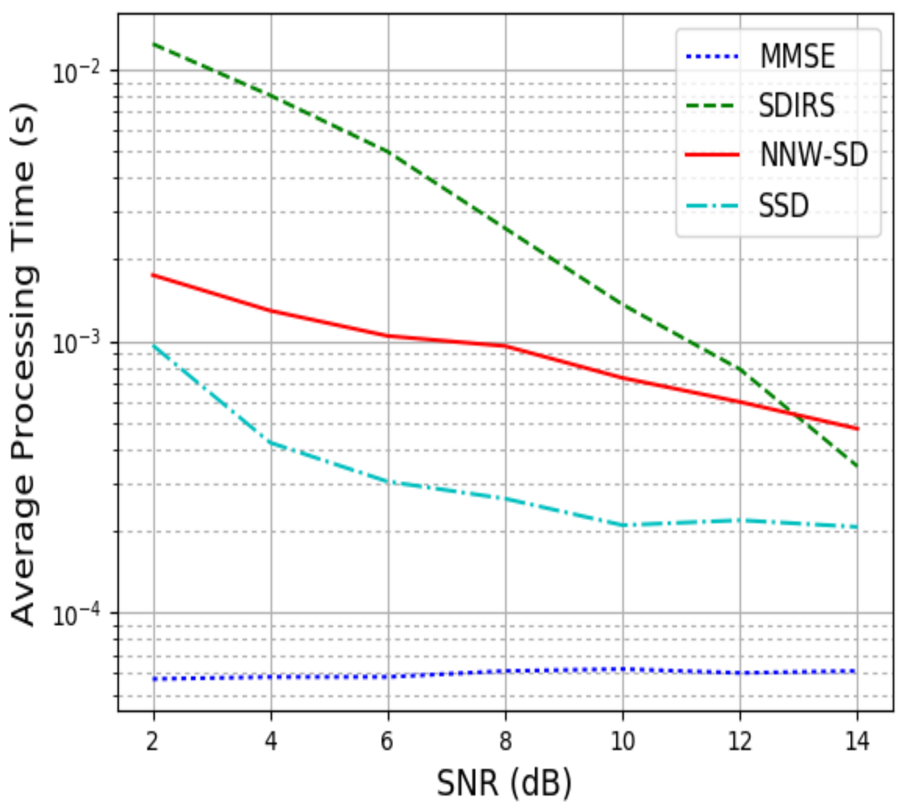
8×8 MIMO system



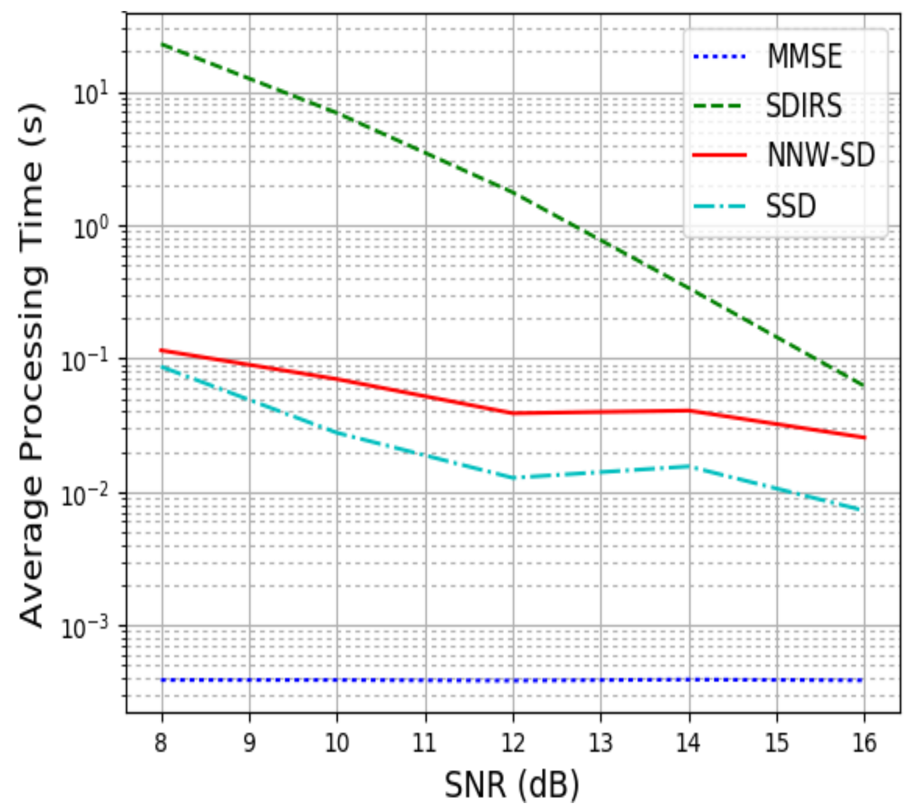
- Number of radius updates is linear function of SNR

Smart SD : Simulation results

8×8 MIMO system



16×16 MIMO system



- For 16×16, at 12 dB SNR: Decoding time of the **SSD** is **33** times that of the MMSE, Decoding time of the **SD** is **4570** times that of the MMSE

Conclusion

- We train a DNN model to predict the number of lattice points falling inside the n-dimensional sphere with an arbitrary radius.
- The proposed model can proceed with accurate approximations for arbitrary lattices compared to analytical upper bounds existing in the literature., and this is for some known lattices.
- A novel low-complexity Sphere Decoder (DNN-SD) based on DNNs is proposed, gives the ML performance with greatly lower computational complexity
- Smart sphere decoder (SSD), achieves ML performance with more significant complexity reduction

Associated publications

- A. Askri, G. Rekaya-Ben Othman and H. Ghauch, “Counting Lattice Points in the Sphere using Deep Neural Networks”, Asilomar Conference on Signals, Systems and Computers”, USA, November 2019.
- A. Askri and G. Rekaya-Ben Othman, “DNN assisted Sphere Decoder”, IEEE ISIT, Paris, France, July 2019.
- G. Rekaya-Ben Othman and A. Askri, “DEVICES AND METHODS FOR MACHINE LEARNING ASSISTED SPHERE DECODING”, European Application July 2019, n° EP 19305886.4.

Question Time !

Smart SD

(2) **it starts the search hypersphere with an enhanced radius r_L** $$r_L^2 = \frac{r_0^2}{2^L}$$

$$\begin{cases} r_0^2 = 2n\sigma_{noise}^2 \\ L = a\rho + b \end{cases}$$

(2) L is the number of radius updates that is a linear function of SNR in dB (**ρ**), with a, b real numbers determined theoretically

$$a = \frac{-1}{10 * \log_{10} 2} \approx 0.332$$

$$b = \frac{2}{n * \log_{10} 2}\left(\log_{10} V_n - E\left[\log_{10} \det(\Lambda)\right] - E\left[\log_{10} N_p\right]\right) + \frac{1}{\log_{10} 2}\left(\log_{10} 2n + \log_{10} \frac{P_t}{2}\right)$$