



# **Rapport du stage**

## **Prédiction du trafic réseau**

### **Réalisé par:**

Hamza Amal

### **Encadré par:**

Riadh Dhaou  
Urtzi Ayesta  
Herwig Wendt

## **Remerciements**

**Je tiens à remercier toutes les personnes qui ont contribué à mon stage et qui m'ont aidé lors de la rédaction de ce rapport.**

**Tout d'abord, j'adresse mes remerciements à mon professeur, Mr Riadh Dhaou de l'école ENSEEIHT qui m'a beaucoup aidé dans ma recherche de stage. Il m'a permis de trouver un stage qui était en totale adéquation avec mes attentes.**

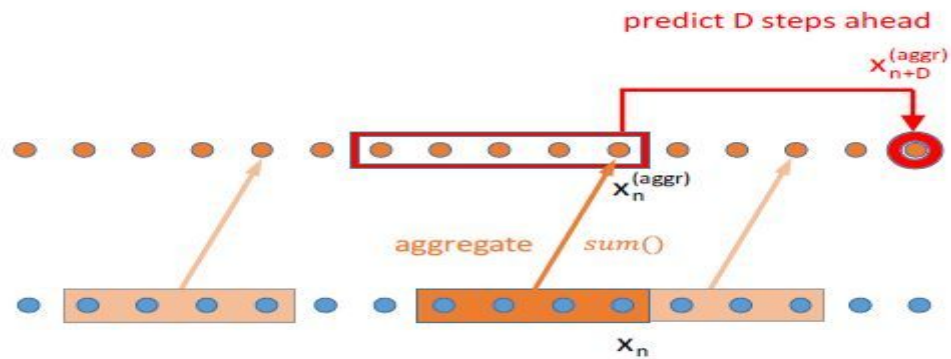
**Je tiens à remercier vivement Mr Urtzi ayesta, et Mr Herwig Wendt Chercheurs à l'IRIT, pour leur accueil, le temps passé ensemble et le partage de leur expertise au quotidien. Grâce à eux j'ai pu avancer dans mon sujet et surmonter tous les problèmes que j'ai rencontré.**

## **1-Introduction.**

Dans cette section, nous allons évaluer les performances des algorithmes d'apprentissage sur un trafic qui relie un terminal qui contient un nombre particulier d'utilisateur que l'on va faire varier par la suite de 500 jusqu'à 10000, à un gateway. Ces performances sont liées au calcul du SDR ( rapport signal distorsion ), du MAD qui représente la déviation moyenne absolue normalisé entre les données d'entrées et le signal prédit, les valeurs du SDR et du MAD se calculent de la manière suivante.

$$\text{SDR} = 10 \log_{10} \left( \|x\|_2^2 / \|x - \hat{x}\|_2^2 \right) \text{dB} \quad \text{MAD} = 100 \left( \frac{\|x - \hat{x}\|_1}{\|x\|_1} \right)$$

On peut également changer la granularité des agrégats de flux afin de réduire la quantité de données applicatives pour l'adapter à la bande passante disponible, autrement dit, pour un nombre particulier «n» d'échantillons on considère qu'un seul échantillon obtenu par une somme cumulée afin d'obtenir une corrélation entre les données, par contre agréger les données suppose une perte d'information due aux algorithmes employés. Le schéma de principe de l'agrégation des données est donné par:



Le paramètre  $D$  est le pas d'estimation que l'on peut ajuster par la suite, appelé aussi l'horizon de prédiction, la figure ci dessus correspond à une distance prédiction égale à 5 et une granularité égale à 4.

## 2-Nombre d'utilisateurs fixe.

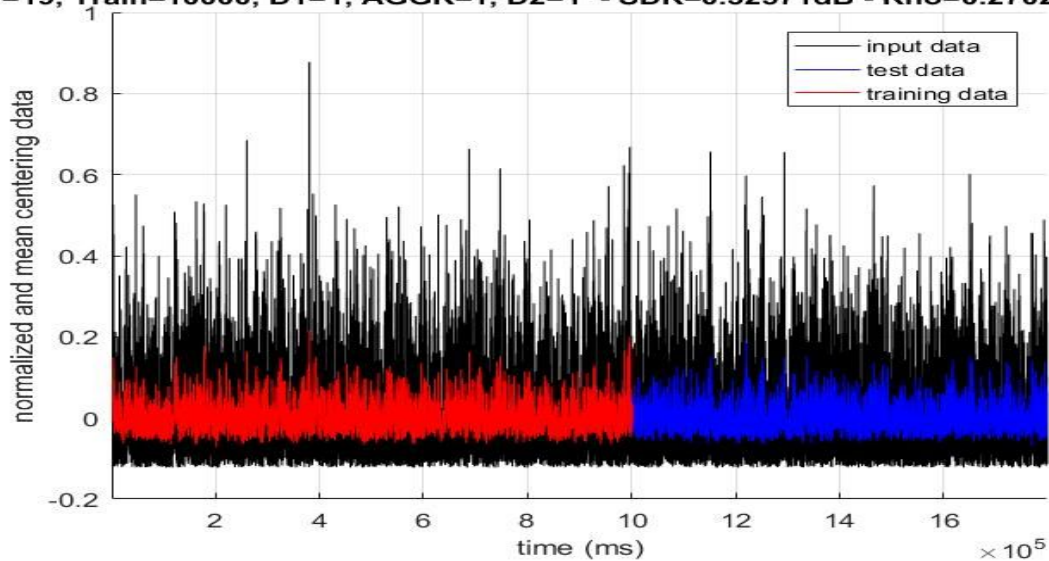
Dans un premier temps, on considère le scénario le plus simple qui consiste à considérer une granularité égale 1 et qui correspond à 100 ms, cette durée représente la différence entre deux mesures consécutives de débit instantané. Ainsi l'horizon de prédiction est fixé à 100 ms. Nous disposons de 500 utilisateurs, nous allons donc tester sur les données de la voie aller associées au débit entrant.

Pour procéder à la prédiction, on utilise 10000 échantillons pour entraîner notre modèle, afin d'estimer les échantillons restants.

Avant d'établir la prédiction, il faut estimer les paramètres de chaque algorithme, et pour ce faire, on utilise le principe de la validation croisé, nous allons donc considérer les 5000 premiers échantillons parmi les 10000 premiers pour estimer/apprendre le modèle, afin de tester sur les 5000 restants, et puis on sélectionne le paramètre optimal qui correspond à celui dont le SDR est élevé. Dans la suite on a pris une liste de valeur de l'ordre de prédiction qui varie de 10 à 20, et puis l'ordre optimal est déterminé par le principe de la validation croisé. Par contre dans les réseaux de neurones, on fixe par la suite le nombre de couches à 3, et on considère 3 neurones dans chaque couche vu la complexité de calcul.

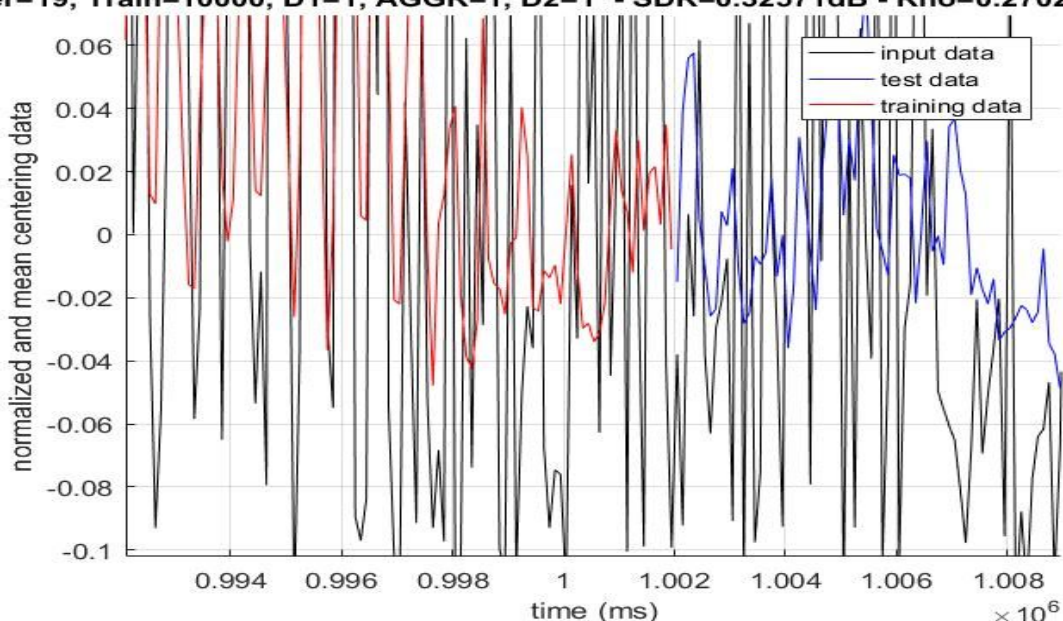
On trace donc le débit instantané obtenu après prédiction superposé avec le signal de d'origine pour visualiser le taux de prédiction de l'algorithme considéré (dans ce cas on utilise la méthode AR). Il est à noter que avant de faire la prédiction on centre et on normalise les données, ce qui justifie le fait que le débit est compris entre -1 et 1.

der=19, Train=10000, D1=1, AGGR=1, D2=1 - SDR=0.32571dB - Rho=0.27029 - SDI

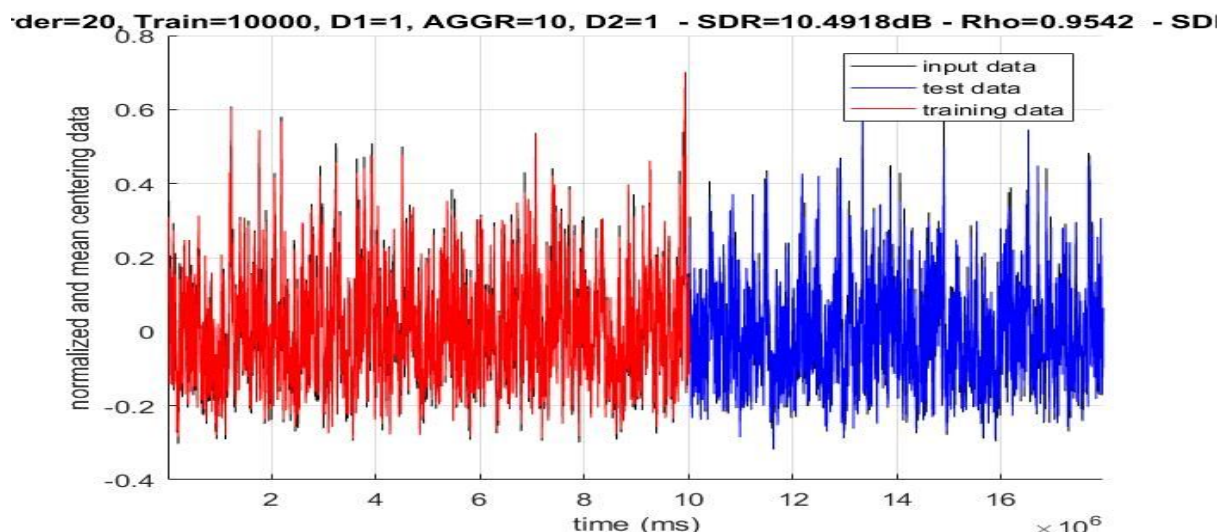


La courbe en rouge correspond au échantillon d'entraînement, celui du bleu représente les échantillons de test, et le signal en noir correspond aux données d'origine. On constate donc qu'on a abouti à une mauvaise prédiction, par contre si on agrandit l'image sur une partie des échantillons en faisant un zoom, ceci peut être justifié par le fait que les données présentent de forte variation dans le temps, ce qui rend difficile l'estimation comme le montre la figure suivante.

der=19, Train=10000, D1=1, AGGR=1, D2=1 - SDR=0.32571dB - Rho=0.27029 - SDI



On peut donc se ramener à effectuer une agrégation des données afin de diminuer le taux de variation, et pour cela nous allons considérer une granularité des agrégats de flux égale à 1s par exemple. On obtient alors une forte corrélation entre le signal estimé et le signal d'origine comme le montre la figure suivante.



On peut donc tirer la même conclusion dans le cas d'augmentation du nombre d'utilisateurs, en fournissant les tables qui mesurent les performances des méthodes d'apprentissage utilisées pour la prédiction (AR, NN, AR NN, ARMA, Baseline, pour une granularité de 100 ms, ce temps correspond à la durée entre deux mesures consécutives, ce qui signifie donc qu'il n'y a pas d'agrégation des données, et on va prédire à l'horizon de 100 ms ( $D = 1$ ).

## 2.1-Granularité de 100 ms, et horizon de 100 ms.

La table suivante représente les résultats obtenus pour le débit entrant après le calcul du SDR, du MAD, et d'erreur absolue pour les différents algorithmes d'apprentissage, le nombre d'utilisateurs est fixé à 500.

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	-2.6851	4.5169e+03	91.1189
AR	0.3257	3.3281e+03	67.1380
ARMA	0.3253	3.3285e+03	67.1443
NN	0.3116	3.3354e+03	67.4408
AR NN	0.2985	3.3099e+03	67.7900

On remarque d'abord que la méthode de base (baseline) fournit une mauvaise prédiction vu que la valeur de rapport signal distorsion (SDR) est négative, ainsi on remarque que les autres algorithmes sont proches en terme de tous les performances (SDR, MAD, Erreur absolue), et ils ne permettent pas d'avoir une bonne prédiction même si la valeur du SDR est positive, ainsi le pourcentage de l'erreur est estimé au alentour de 67 pour cent après le calcul

du MAD, ce qui signifie donc qu'on est amené à faire une agrégation des données comme on a montré précédemment.

### **750 utilisateurs.**

On change maintenant le nombre d'utilisateur, et on fixe les autres paramètres et on calcule les paramètres de mesures de performances associées pour les méthodes d'apprentissage utilisées

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.6163	5.9521e+03	77.0651	-3.1446	7.1852e+03	93.0349
AR	0.4312	4.3211e+03	55.9484	0.4729	4.9049e+03	63.5093
ARMA	0.4311	4.3213e+03	55.9502	0.4747	4.9051e+03	63.5118
NN	0.3710	4.3945e+03	56.6909	0.4276	4.9474e+03	64.7789
AR NN	0.3937	4.3738e+03	56.2293	0.4142	5.0074e+03	63.3044

On peut donc tirer la même conclusion que dans le cas de 500 utilisateurs, par contre dans ce scénario on remarque une légère augmentation de la valeur du SDR et une diminution du pourcentage d'erreur (MAD), ainsi les deux méthodes linéaires AR, et ARMA fournissent de bonne prédiction par rapport aux autres algorithmes non linéaires (NN, AR NN), et comme précédemment, on remarque que la méthode de base ne fonctionne pas car elle fournit une valeur négative pour le SDR.

### **5000 utilisateurs.**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.4837	1.1762e+04	22.8397	-2.8179	1.5636e+04	30.3671
AR	0.3324	8.5298e+03	16.5636	0.0861	1.1280e+04	21.9065
ARMA	0.3320	8.5305e+03	16.5648	0.0860	1.1280e+04	21.9068
NN	0.3182	8.5465e+03	16.6663	0.0762	1.1308e+04	21.8637

AR NN	0.2534	8.5827e+03	16.6024	0.0483	1.1391e+04	22.0497
-------	--------	------------	---------	--------	------------	---------

On voit donc que les valeurs du SDR pour les algorithmes linéaires et non linéaires sont toujours faibles, on a pas abouti donc à la prédiction attendue, ainsi la méthode de base ne fonctionne pas.

## 10000 utilisateurs

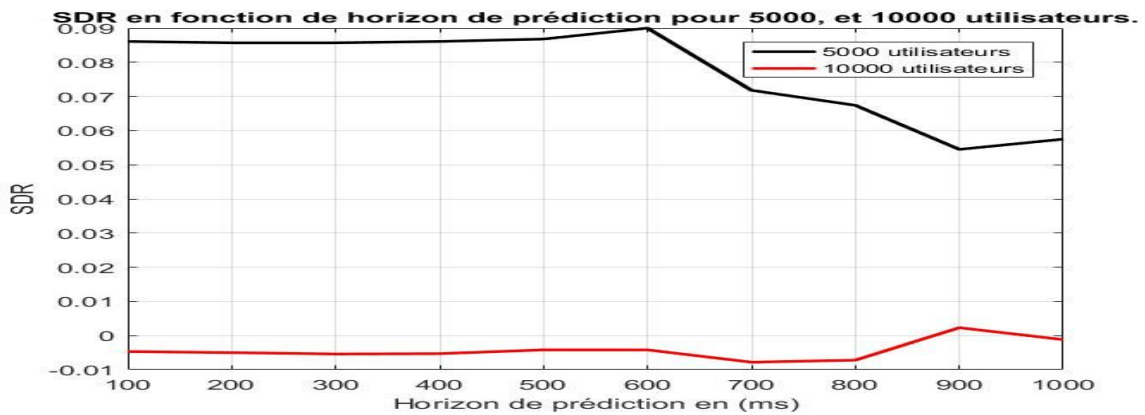
	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.8552	9.4097e+03	15.3557	-2.9938	1.5821e+04	25.8217
AR	-0.0049	6.7899e+03	11.0805	-0.0047	1.1273e+04	18.3989
ARMA	-0.0025	6.7898e+03	11.0805	-0.0015	1.1269e+04	18.3988
NN	-0.0345	6.7918e+03	11.0814	-9.4205e-04	1.1270e+04	18.4012
AR NN	-0.0510	6.8109e+03	11.0992	-0.0331	1.1320e+04	18.5843

On remarque donc que dans le cas d'augmentation brutale du nombre d'utilisateurs, tous les algorithmes ne fonctionnent pas car ils donnent des valeurs négatives du SDR même si le pourcentage d'erreur a diminué par rapport aux cas précédents, ceci peut être justifié par le fait que le signal varie rapidement, il résulte donc une difficulté au niveau de la prédiction, on est donc amené à faire une agrégation des données.

## 2.2-Granularité de 100 ms, et horizon de 1s

On a vu précédemment que tous les algorithmes ne donnent pas de bonne prédiction pour une granularité de 100 ms, et un horizon de 100 ms, ainsi on peut tirer la même conclusion si on essaie à prédire à l'horizon de 1s. Pour affirmer ce postulat, nous allons considérer par exemple les données associées à 5000, 10000 utilisateurs pour le débit sortant, et on va utiliser la méthode AR ( on peut également choisir la méthode ARMA vu qu'ils présentent tous des performances assez proches en terme du SDR et du MAD, et qu'ils fournissent de bon résultats par rapport aux autres algorithmes non linéaires dans le cas d'une distance de prédiction égale à 100 ms) , et nous allons donc tracer le SDR en fonction de l'horizon de prédiction que nous allons faire varier de 100 ms à 1s.





On voit donc que pour toutes les valeurs de l'horizon de prédiction, la valeur de SDR est faible pour 5000 utilisateurs, et négatives pour 10000 utilisateurs, ce qui prouve que la prédiction n'est pas satisfaisante, on fournit comme même les tables des valeurs de SDR, du MAD, et d'erreur absolu pour différents utilisateurs.

### 500 utilisateurs

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	-3.0470	4.7354e+03	95.5186
AR	0.0945	3.4683e+03	69.9601
ARMA	0.0965	3.4678e+03	69.9498
NN	0.0838	3.4762e+03	69.5432
AR NN	0.0925	3.4916e+03	69.4766

### 750 utilisateurs

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-3.0111	6.2529e+03	80.9754	-2.9066	6.9352e+03	89.8362
AR	0.1668	4.4835e+03	58.0617	0.2407	5.0789e+03	65.7895
ARMA	0.1659	4.4843e+03	58.0712	0.2390	5.0810e+03	65.8173
NN	0.1416	4.5131e+03	58.3830	0.2118	5.1155e+03	66.2946

AR NN	0.1125	4.5497e+03	58.9861	0.1783	5.1703e+03	65.7090
-------	--------	------------	---------	--------	------------	---------

### **5000 utilisateurs.**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.8366	1.2315e+04	23.9151	-2.8107	1.5582e+04	30.2610
AR	0.1590	8.7106e+03	16.9157	0.0575	1.1331e+04	22.0052
ARMA	0.1582	8.7110e+03	16.9166	0.0572	1.1331e+04	22.0053
NN	0.1447	8.7208e+03	16.9673	0.0306	1.1343e+04	22.0220
AR NN	0.1294	8.7600e+03	17.0117	0.0134	1.1458e+04	22.0769

### **10000 utilisateurs**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.9969	9.6299e+03	11.0513	-3.0445	1.5931e+04	25.9993
AR	0.0228	6.7727e+03	11.0513	-0.0012	1.1271e+04	18.3940
ARMA	0.0226	6.7727e+03	11.0513	-0.0012	1.1271e+04	18.3943
NN	-0.0421	6.8069e+03	11.0798	-5.2152e-04	1.1277e+04	18.4072
AR NN	-8.3128e-04	6.7852e+03	11.0898	-0.0200	1.1291e+04	18.5328

### **2.3-Granularité de 1s, et horizon de 1s**

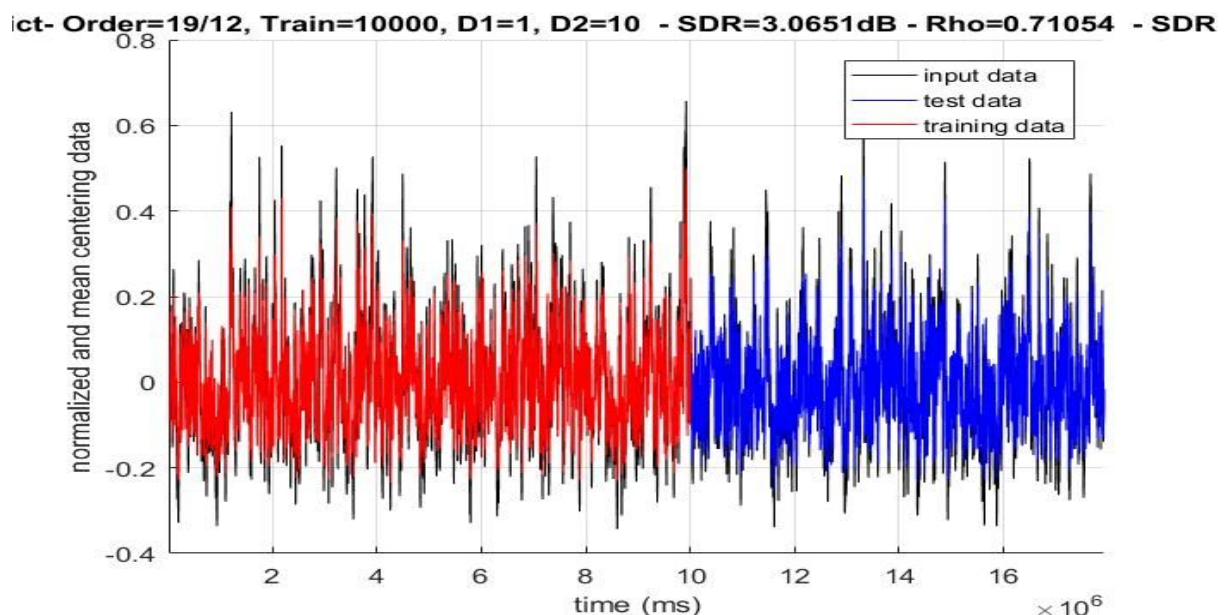
Dans cette partie, nous allons effectuer une agrégation des données, on va donc considérer une granularité égale à 1s, et nous allons faire une prédiction à l'échelle de 1s. On fournit donc les tables qui représentent les performances obtenues par le calcul du SDR, le MAD, et l'erreur absolu en utilisant les

différents algorithmes ( AR, NN, AR NN, ARMA), et la méthode de base pour différents nombre d'utilisateurs.

## 500 utilisateurs

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	-1.2566	1.5947e+04	32.1704
AR	0.5567	1.3091e+04	26.4082
ARMA	0.4993	1.3208e+04	26.3938
NN	0.5624	1.3084e+04	26.4187
AR NN	3.0651	8.7328e+04	17.6387

On remarque donc que la méthode AR NN présente une bonne prédiction, vu que la valeur de SDR associée est grande par rapport à celle des autres méthodes, ainsi le pourcentage de l'erreur est plus petit par rapport aux autres algorithmes ce qui présente donc une forte corrélation entre le signal prédit et le signal d'origine comme le montre la figure ci dessous.



Par contre cette méthode (AR NN) présente une grande complexité de calcul vu le nombre de couches et de neurones dans chaque couche, en effet dans le programme, on a pris une liste de valeurs de neurones et 3 couches, et puis on a utilisé le principe de la validation croisée pour sélectionner la valeur optimale du nombre de neurones .Ainsi les deux méthodes linéaires ARMA et

AR donnent de bonnes résultats au niveau de MAD ( 25 pour cent de l'erreur), par contre la valeur du SDR n'est pas satisfaisante, ainsi ils présentent une bonne complexité de calcul par rapport aux autres méthodes non linéaires.

Dans la suite on va évaluer les performances pour 750 , 5000, 10000 utilisateurs sur les données associées aux débits entrants, et sortants.

### **750 utilisateurs**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-0.7710	1.9359e+04	25.0793	-1.0563	2.0621e+04	26.7181
AR	0.9146	1.6131e+04	20.8967	0.8008	1.6856e+04	21.8396
ARMA	0.9192	1.6124e+04	20.8884	0.8021	1.6852e+04	21.8350
NN	0.8744	1.6264e+04	21.0841	0.7679	1.6992e+04	21.9078
AR NN	3.6195	1.0613e+05	13.7557	3.2777	1.1228e+05	14.4746

On peut donc tirer la même conclusion que dans le cas de 500 utilisateurs, la méthode de base ne fonctionne pas très bien car elle fournit une valeur négative de la SDR, la méthode AR NN présente une bonne estimation et un bon pourcentage de l'erreur, ainsi les autres méthodes fournissent des résultats satisfaisants au niveau du MAD, surtout les deux méthodes linéaires ARMA, AR qui sont moins complexes.

### **5000 utilisateurs.**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-0.2072	3.8492e+04	7.4754	-1.1846	4.8204e+04	7.5498
AR	1.2288	3.2576e+04	6.3265	0.6892	3.8874e+04	9.3619
ARMA	1.2241	3.2603e+04	6.3317	0.6893	3.8875e+04	7.5500
NN	1.1593	3.2853e+04	6.3559	0.6570	3.9043e+04	7.6109
AR NN	4.0203	2.1584e+05	4.1891	3.0556	2.6260e+05	5.0780

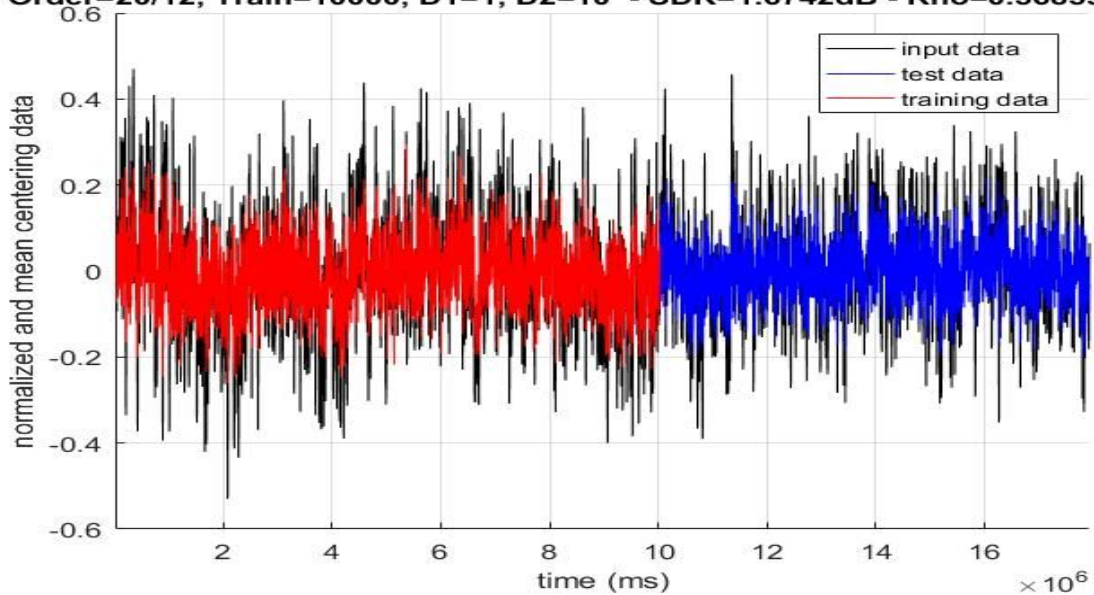
De même, l'algorithme qui présente une meilleure prédiction est celui associé à la méthode hybride AR NN, qui fournit une valeur très élevée du SDR si on passe de l'échelle logarithmique à l'échelle linéaire, et une petite valeur du pourcentage de l'erreur, ainsi les performances des méthodes AR, ARMA, et NN sont très proches en terme de SDR, Erreur absolu, et du MAD, sauf que la méthode NN présente une grande complexité de calcul, on a donc le choix entre une méthode linéaire ( AR, ARMA), et la méthode AR NN.

### **10000 utilisateurs**

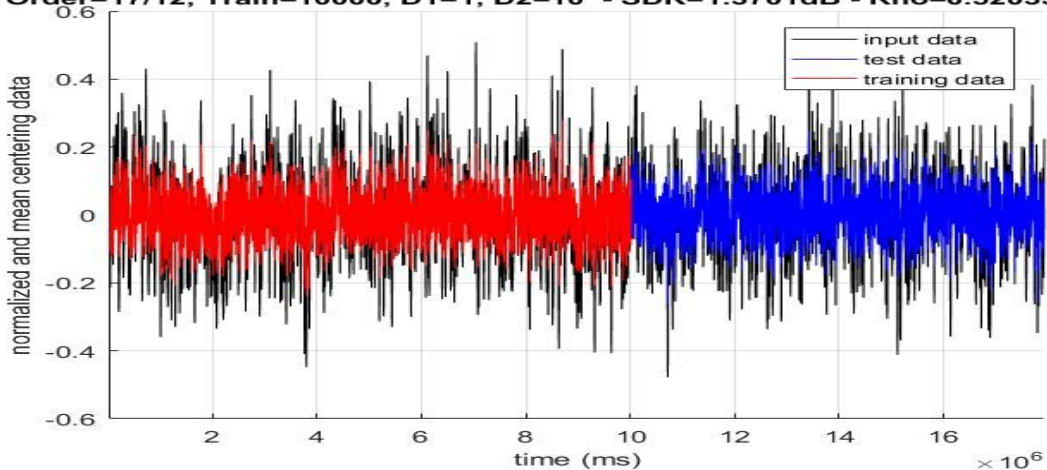
	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-3.1442	3.1409e+04	5.1247	-3.3001	5.1227e+04	8.3599
AR	0.2478	2.1358e+04	3.4848	-0.0028	3.4983e+04	5.7089
ARMA	0.2498	2.1358e+04	3.4848	0.2498	2.1358e+04	5.7080
NN	0.0279	2.1854e+04	3.5592	-0.0251	3.5009e+04	5.7115
AR NN	1.6742	1.4856e+05	2.4295	1.3701	2.4187e+05	3.9525

On voit donc que les méthodes AR, ARMA, NN ne fournissent pas des résultats de prédiction satisfaisants dans le cas d'augmentation brutale du nombre d'utilisateurs, par contre la méthode AR NN reste toujours valable, et donne une bonne prédiction comme le montre les deux figures suivantes ( débit entrant pour la figure en haut, et celle en bas est associée au débit sortant ).

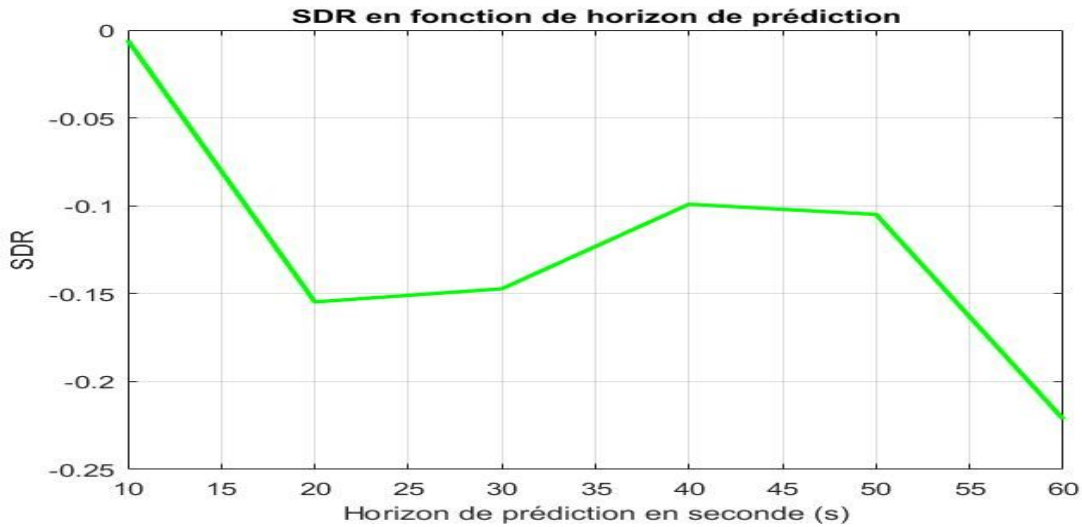
ict- Order=20/12, Train=10000, D1=1, D2=10 - SDR=1.6742dB - Rho=0.56859 - SDR



ict- Order=17/12, Train=10000, D1=1, D2=10 - SDR=1.3701dB - Rho=0.52033 - SDR



Pour visualiser l'impact de l'horizon de prédiction, on prend le cas de 10000 utilisateurs, et on considère les données associées au débit entrant. On va donc tracer le SDR en fonction de la distance de prédiction pour les valeurs 10s, 20s, 30s, 40s, 50s, et 60s, en considérant une granularité égale à 1s pour la méthode AR NN qui a pu fournir de bonne prédiction dans les scénarios précédents.



On voit donc que les valeurs de SDR sont négatives, ce qui signifie donc que la prédiction est faible, et donc on peut pas prédire à l'échelle de 1 min pour une granularité de 1s ( même chose pour 10 min, ....).

## 2.4-Granularité de 10s.

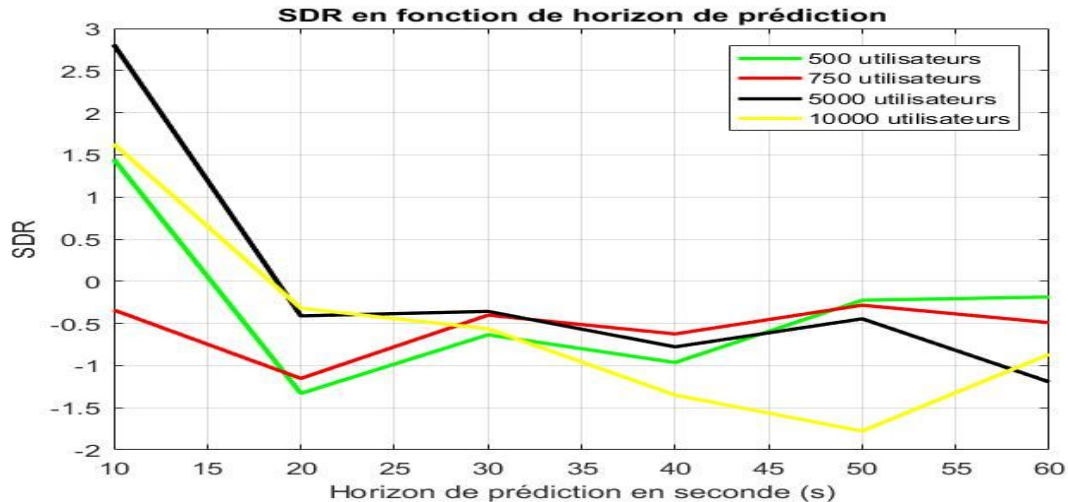
Dans cette partie nous allons augmenter la granularité des agrégats de flux à 10 s pour voir à quelle mesure on peut faire la prédiction en utilisant les différents algorithmes.

On commence donc par fournir la table de mesure de performance pour 500 utilisateurs, et un horizon de prédiction égale à 10s pour conclure sur l'algorithme le plus efficace.

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	-3.2771	9.7466e+04	19.6200
AR	-0.0073	6.8011e+04	13.6907
ARMA	-0.0254	6.8108e+04	13.7101
NN	-0.0695	6.8456e+04	13.7799
AR NN	1.4691	4.8559e+06	9.7730

On remarque donc l'algorithme le plus efficace est celui de la méthode hybride AR NN, tandis que les autres algorithmes ( AR, NN, AR NN) ne fournissent aucune prédiction vu que les valeurs de SDR sont négatives.

On va donc tracer les valeurs du SDR en fonction de l'horizon de prédiction pour différent nombre d'utilisateurs en utilisant l'algorithme AR NN.



On voit donc que le seul cas où l'algorithme fournit une prédiction est le cas où l'horizon de prédiction est égale à la granularité ( 10 s) quelque soit le nombre d'utilisateur, par contre au fur et à mesure on augmente la distance de prédiction on voit que la valeur du SDR devient négative et donc l'algorithme ne fonctionne pas. Donc on peut pas prédire à l'échelle de 1 min et 10 min, 30 min.

## 2.5-Granularité de 1 min.

On reprend la même démarche que la partie précédente, on fixe donc le nombre d'utilisateur à 750 par exemple et les données associées au débit entrant, et on calcule les valeurs de SDR, d'erreur absolue, et du MAD pour une granularité de 1 min, et on va prédire à l'échelle de 1 min.

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	-2.3395	3.2809e+05	7.1318
AR	-0.0030	2.5493e+05	5.5414
ARMA	-0.0022	2.5484e+05	5.5395
NN	-0.6836	2.7802e+05	6.0433
AR NN	-1.4248	1.6060e+08	5.8878

On remarque donc que les valeurs de SDR pour tous les algorithmes sont négatives ce qui permet donc de conclure sur l'impossibilité de prédire les échantillons de futurs pour une granularité des agrégats de flux égale 1 min, ainsi on peut constater aussi qu'on peut pas faire une prédiction si on augmente l'horizon de prédiction de 1 min jusqu'à 10 min.



### **3-Nombre d'utilisateurs variable.**

#### **3.1-Granularité de 1s, et 10s.**

Dans cette section, nous allons utiliser les données pour un nombre variable d'utilisateur, nous disposons de 6 scénarios pour le débit entrant et sortant, nous allons donc mesurer les performances de chaque méthode comme précédemment. On commence donc par visualiser le cas le plus simple qui consiste à considérer une granularité nulle, et faire une prédiction à l'échelle de 100 ms, on prend par exemple le scénario 1.

Après le test de tous les méthodes, les algorithmes qui fournissent la meilleur prédiction sont AR et ARMA, ils fournissent une valeur de SDR de l'ordre de 0.79, cette valeur est susceptible de diminuer si on essaie à prédire à une échelle un peu loin ( de la minute par exemple ), nous allons donc augmenter la valeur de la granularité à 1 s et on verra à quelle mesure on va réussir notre prédiction.

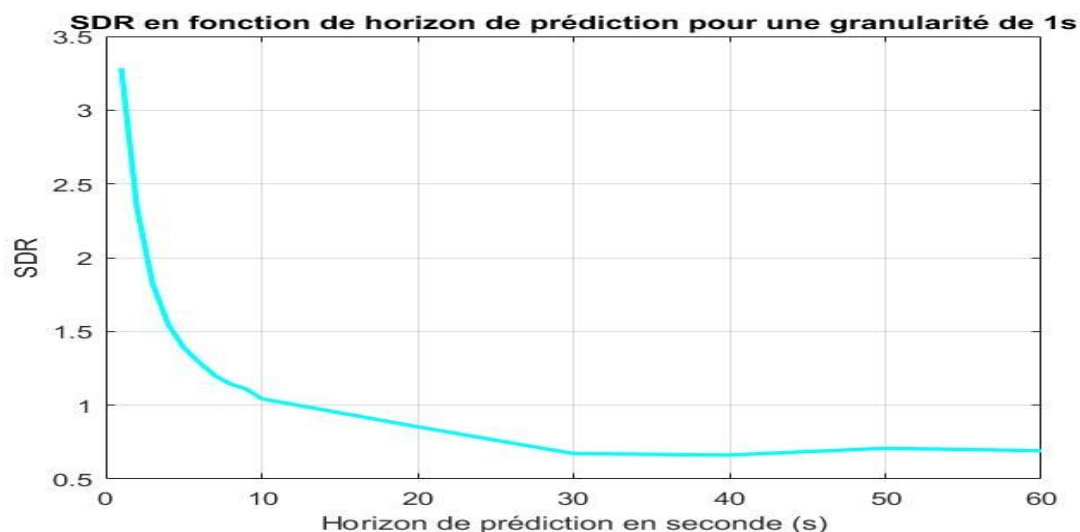
On fournit donc la table de mesure de SDR, de l'erreur absolu, et du MAD pour un horizon de 1s.

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	2.3086	2.9079e+04	8.3365
AR	3.1829	2.6319e+04	7.5453
ARMA	3.2831	2.6041e+04	7.4656
NN	0.3415	3.8011e+04	10.8972
AR NN	5.9261	1.8362e+05	5.2643

On remarque cette fois ci que l'algorithme baseline donne de bon résultat par rapport au cas de nombre fixe d'utilisateur, ainsi les algorithmes linéaires fournissent de très bon résultat par rapport à la méthode de réseau de neurones et la méthode hybride (AR NN) en terme de SDR, même si la méthode AR NN donne une valeur élevé de SDR et un bon pourcentage de calcul, par contre le temps et la complexité de calcul sont assez élevés dans le cas d'augmentation de la granularité ou bien de l'horizon de prédiction, ainsi l'erreur de pourcentage obtenu pour les méthodes linéaires est satisfaisante,

on va donc se baser sur la méthode ARMA pour voir l'horizon de prédiction maximal qu'on peut atteindre.

On trace donc le SDR obtenu en fonction de la distance de prédiction pour le même scénario en utilisant l'algorithme ARMA ( On peut également utiliser l'algorithme AR).



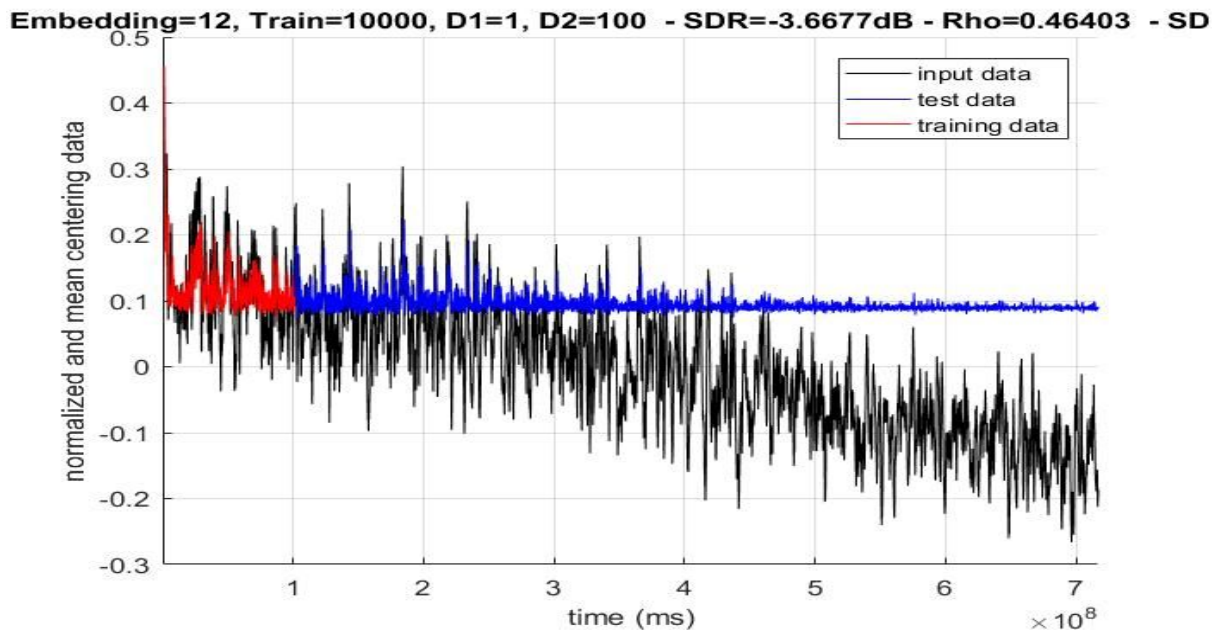
On voit donc que la valeur de SDR ( en dB ) diminue exponentiellement en fonction de l'horizon de prédiction, et on constate aussi que l'algorithme nous a permis de faire une prédiction à l'échelle de 10s, ainsi on peut prédire à l'échelle de 1 min mais la valeur de SDR est non satisfaisant.

On essaie donc d'augmenter la granularité à 10s , et on fournit la table des performances pour tous les algorithmes pour le même scénario afin de conclure sur l'efficacité des méthodes utilisées.

Méthodes de prédiction	SDR	Erreur absolue	MAD
Baseline	2.0646	2.3002e+05	6.5944
AR	2.7221	2.1328e+05	6.1145
ARMA	3.1009	2.0349e+05	5.8340
NN	-3.6677	4.7321e+05	13.5667
AR NN	4.6579	1.6356e+07	4.6912

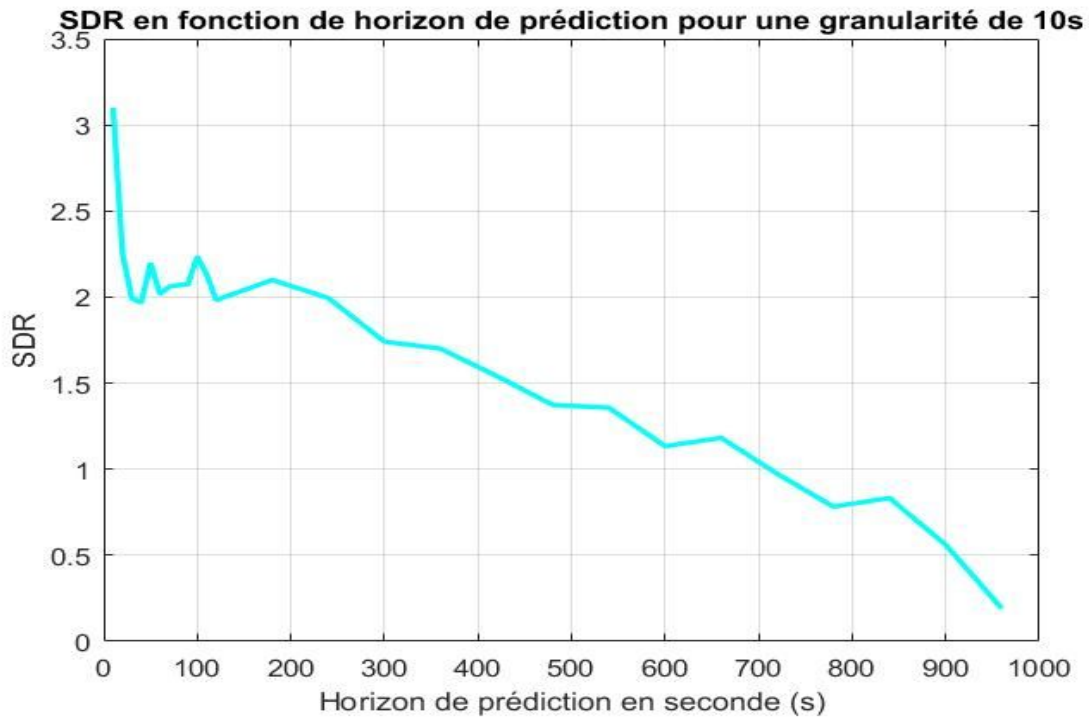
On peut donc tirer la même conclusion que dans le cas d'une granularité de 1s, la méthode AR NN est la plus puissante en terme de prédiction, par contre il présente une grande complexité de calcul, et donc la méthode qu'on peut choisir dans ce cas est la méthode ARMA car elle fournit une bonne valeur de SDR, et un pourcentage d'erreur tolérable, ainsi on remarque une chute brutale

de la valeur de SDR pour l'algorithme de réseau de neurones, on fournit donc le tracé du débit prédit superposé avec le signal d'origine.



On voit donc que l'algorithme de réseau de neurones n'a pas pu gérer la décroissance du signal d'origine, car l'enveloppe du signal prédit est constant au cours du temps.

On trace donc le SDR en fonction de l'horizon de prédiction en utilisant le même scénario et en appliquant l'algorithme ARMA qui pu fournir de bon résultat par rapport aux autres méthodes.



On remarque donc que l'algorithme ARMA nous a permis de faire une prédiction à l'échelle de la minute pour une granularité de 10s, par exemple pour un horizon de 11 min on a pu obtenir un SDR de 1.1822, ce qui est à peu près satisfaisante.

### 3.2-Granularité de 2 min.

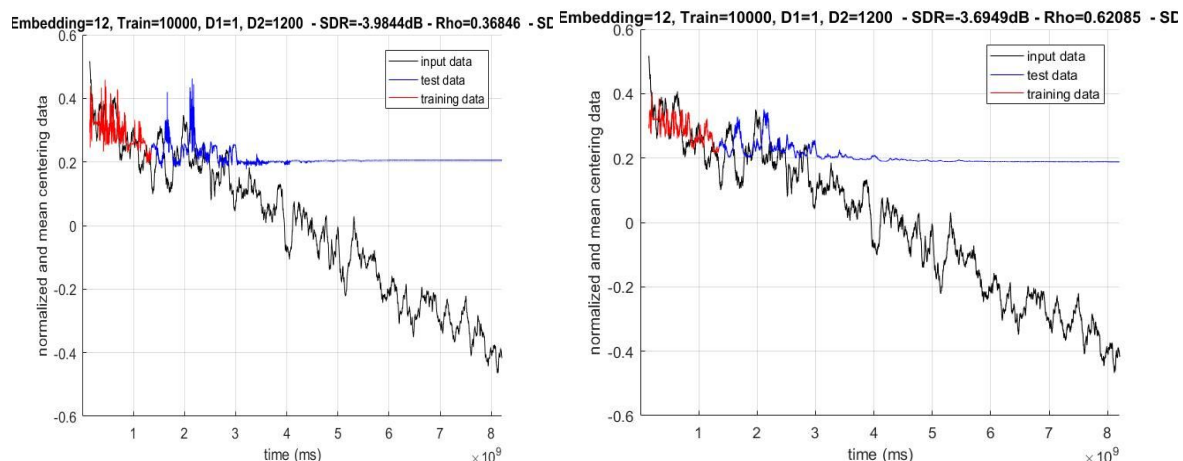
Dans cette partie nous allons augmenter la granularité à l'échelle de la minute, le flux d'agrégation est fixé à 2 min et on commence à prédire à l'horizon de 2 min.

On commence donc par fournir les tables de performances pour tous les scénarios en utilisant les données associées aux débit entrant et sortant, on va garder les même paramètres des algorithmes, sauf que pour la méthode hybride AR et NN, nous allons considérer qu'une seule valeur du nombre de neurones vu la complexité de calcul à cause de l'augmentation des agrégats de flux, et de l'horizon de prédiction. Après la sélection de l'algorithme le plus efficace en terme de SDR et de pourcentage de l'erreur ( MAD ), nous allons essayer d'augmenter la distance de prédiction et évaluer la valeur du SDR pour voir à quelle mesure on peut faire la prédiction.

## Scénario 1

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	8.4982	1.0211e+06	2.4431	8.5776	1.0135e+06	2.4250
AR	8.0392	1.0929e+06	2.6151	8.0919	1.0868e+06	2.6001
ARMA	7.8857	1.1217e+06	2.6839	7.9294	1.1158e+06	2.6696
NN	-3.9844	4.4911e+06	10.7457	-3.6949	4.3477e+06	10.4022
AR NN	7.3613	1.3606e+09	2.7183	7.6163	1.3250e+09	2.6469

On voit donc que toutes les méthodes donnent de bonne prédiction en terme du SDR, et du MAD sauf l'algorithme de réseau de neurones qui présente une valeur négative du rapport signal distortion. La figure suivante représente le signal de test superposé avec le signal d'origine ( la figure à gauche pour le débit entrant, et celle à gauche pour le débit sortant ).

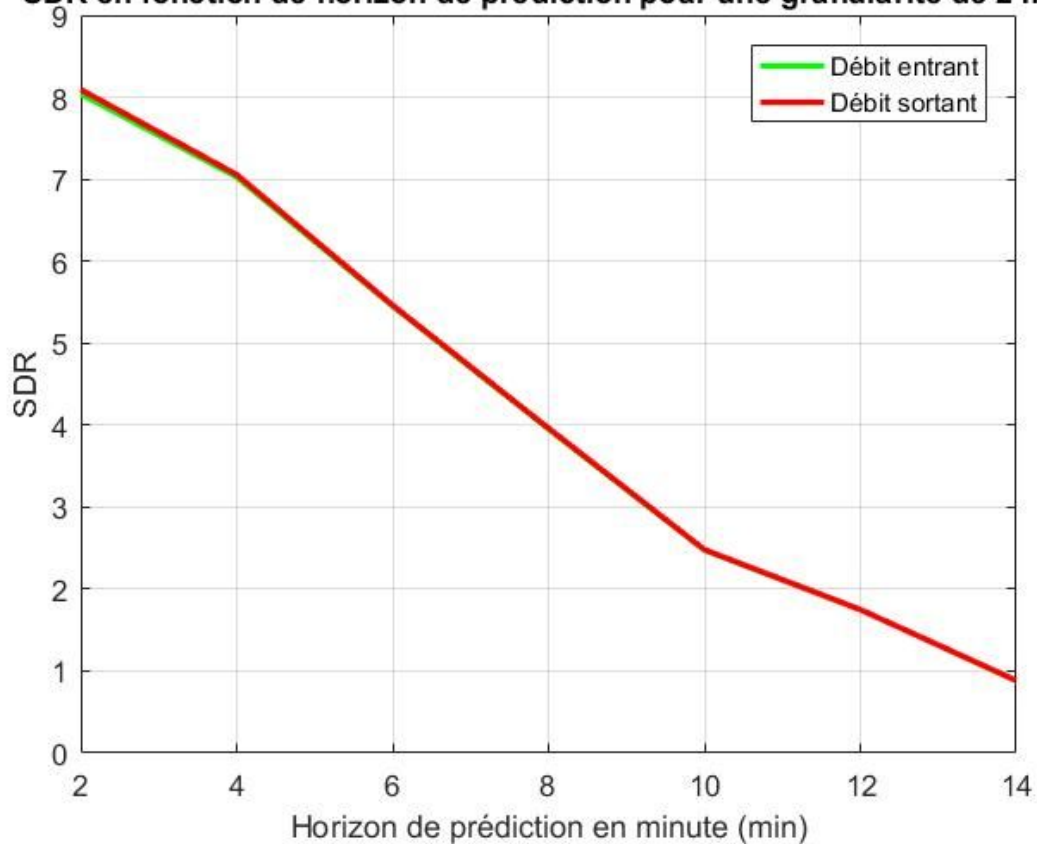


On voit donc que le signal prédit devient constant après un certain temps, on constate donc que l'algorithme de réseau de neurones n'a pas pu poursuivre la monotonie du signal d'origine.

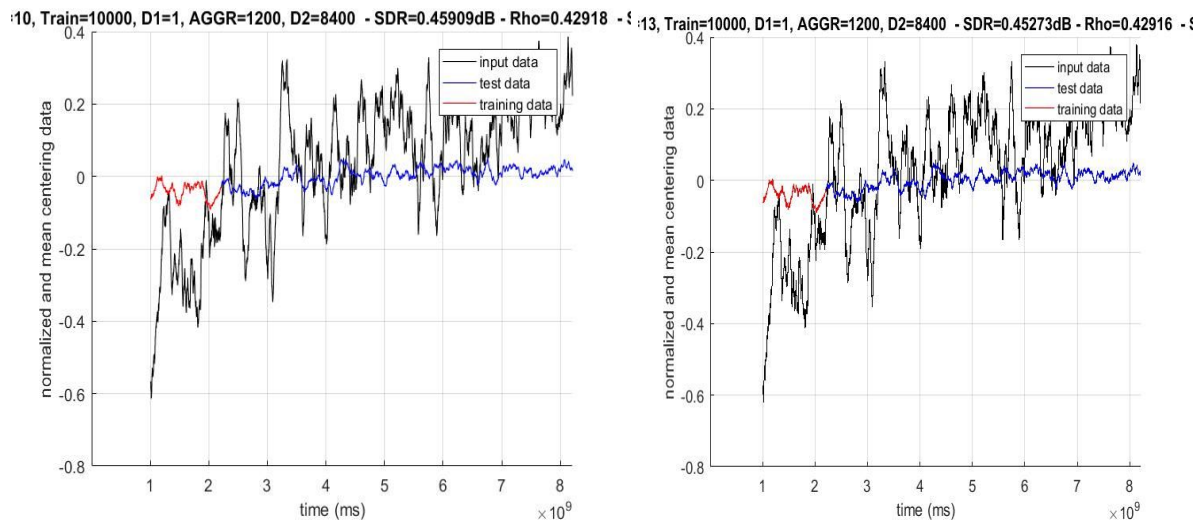
la meilleure méthode adaptée à ce scénario est la méthode AR vu qu'il présente une valeur élevée de SDR si on passe à l'échelle linéaire, ainsi le pourcentage de l'erreur est petit.

On va donc tracer le SDR (en dB) en fonction de l'horizon de prédiction pour voir à quelle mesure cet algorithme reste valable.

**SDR en fonction de horizon de prédiction pour une granularité de 2 min**



On voit donc qu'en utilisant l'algorithme AR, on obtient une bonne prédiction pour un horizon allant de 2 min jusqu'à 12 min, par contre pour une distance de prédiction de 14 min la valeur de SDR devient faible, nous allons donc tracer le signal prédit pour le débit entrant et sortant pour un horizon de 14 min superposé avec le signal d'origine ( la figure à droite représente le débit sortant et celle à gauche le débit entrant).



On voit bien que le signal prédit est quasiment constant au cours du temps, ainsi la cadence de la fluctuation est faible est très faible, et l'algorithme n'a pas pu détecter la décroissance du débit dans les 2 cas.

## **Scénario 2**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-1.6555	8.5616e+05	3.5342	-1.6507	8.5830e+05	3.5428
AR	0.3706	6.7711e+05	2.7951	0.3684	6.7932e+05	2.8040
ARMA	0.4841	6.6519e+05	2.7459	0.4940	6.6607e+05	2.7493
NN	-5.6308	1.5378e+06	6.3481	-5.5266	1.5222e+06	6.2833
AR NN	-2.8923	1.0259e+09	3.5384	-3.6339	1.1183e+09	3.8566

On remarque donc que les méthodes baseline, NN, et AR NN ne donnent pas de prédiction, par contre les deux algorithmes linéaires donnent des valeurs très petits pour le SDR ce qui signifie donc qu'on peut pas faire une prédiction à l'échelle de 10 min.

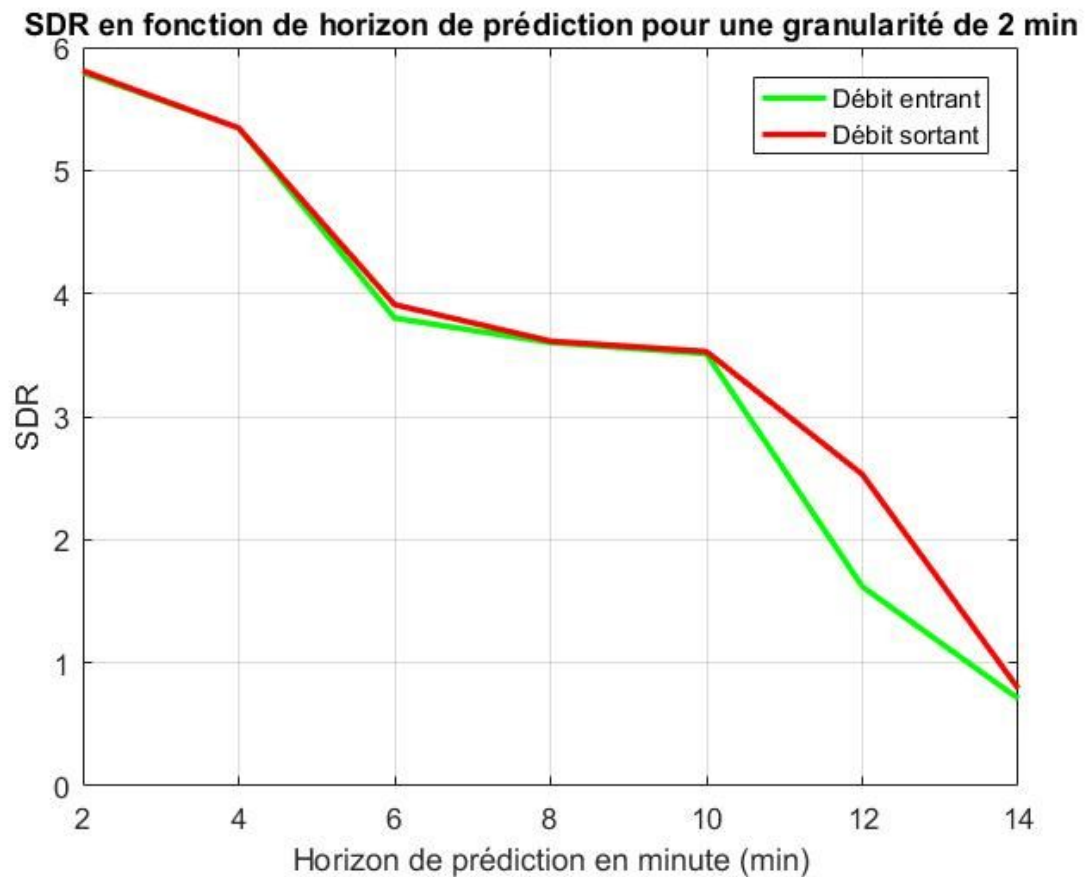
## **Scénario 3**

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	5.2071	9.3619e+05	3.3425	5.2063	9.3609e+05	3.3420
AR	5.4008	9.0580e+05	3.2340	5.4029	9.0554e+05	3.2329
ARMA	5.7948	8.6665e+05	3.0942	5.8099	8.6575e+05	3.0909
NN	-2.6783	2.4568e+06	8.7717	-5.6367	3.4177e+06	12.2018
AR NN	5.1783	1.0711e+09	3.1843	5.6737	1.0054e+09	2.9890

On voit bien donc que l'algorithme de réseau de neurones ne fonctionne pas très bien, car comme on vu précédemment le signal prédit obtenu par cette méthode devient quasiment constant au bout d'un certain temps, par contre dans ce scénario on a un nombre variable d'utilisateur, et l'enveloppe du

signal peut augmenter ou diminuer. Par contre les autres algorithmes sont proches en termes de performances, ils fournissent de bon résultats de SDR, et un bon pourcentage de l'erreur vu la valeur du MAD, les méthodes qui semblent les mieux adaptées à cette situation sont les méthodes linéaires AR et ARMA.

On procède donc de la même manière, et on trace le SDR en fonction de l'horizon de prédiction en utilisant l'algorithme ARMA.



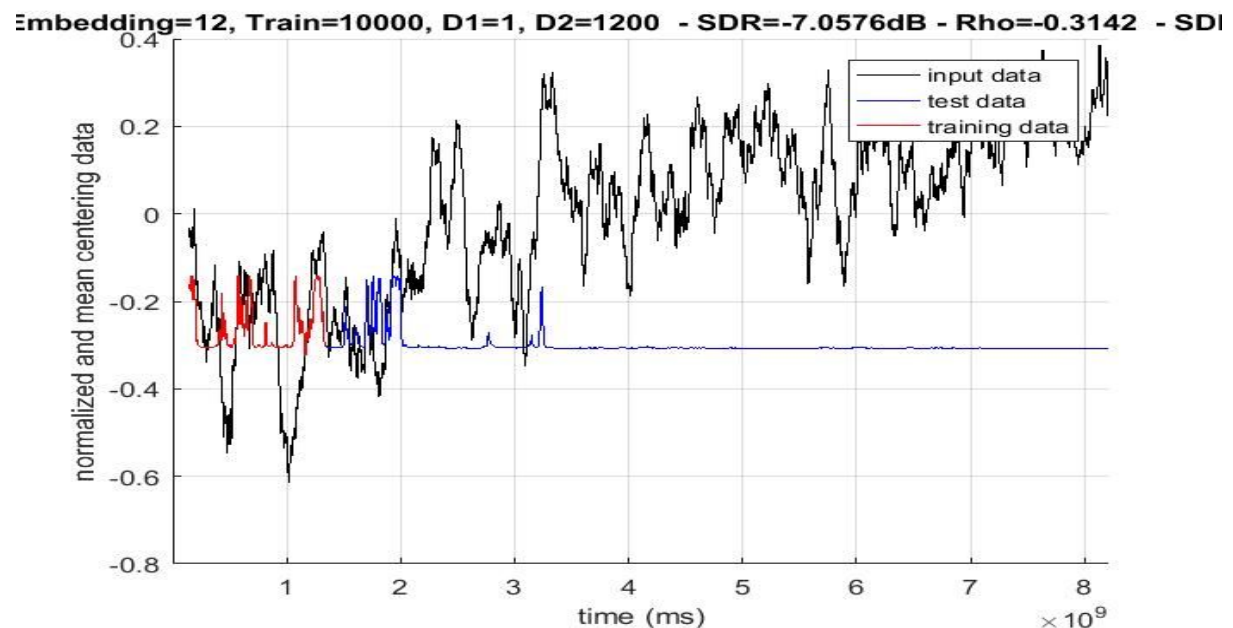
On voit donc qu'on peut faire la prédiction jusqu'à l'échelle de 12 min, par contre la valeur de SDR commence à décroître brutalement à partir de l'horizon de 14 min.



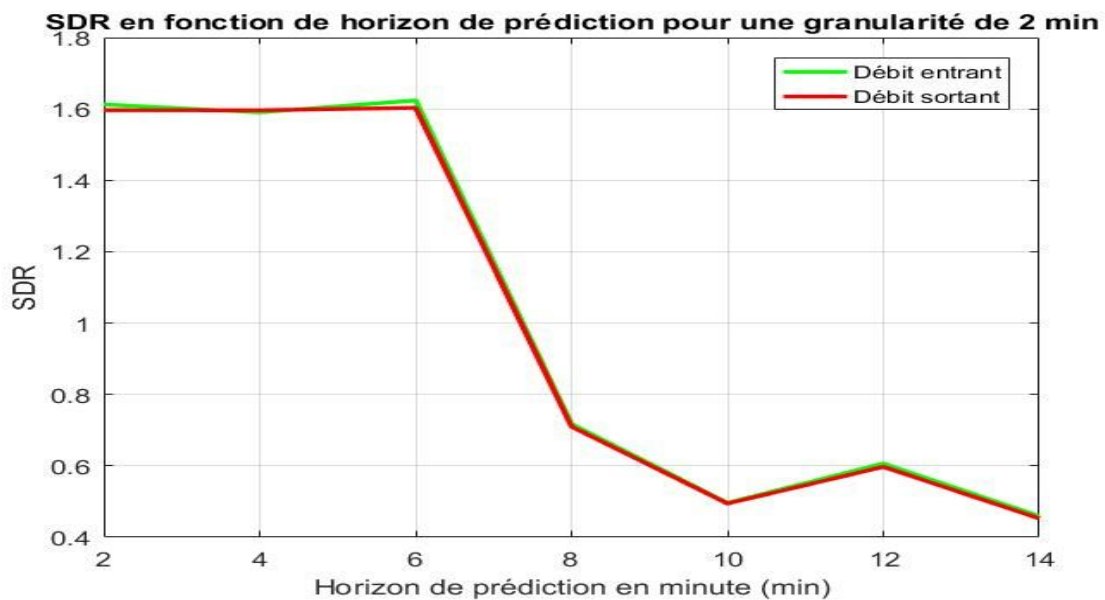
## Scénario 4

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	0.6785	8.7922e+05	2.1666	0.6562	8.7962e+05	2.1676
AR	1.6125	8.0399e+05	1.9812	1.5965	8.0443e+05	1.9823
ARMA	1.6012	8.0502e+05	1.9838	1.5849	8.0547e+05	1.9848
NN	-7.0576	2.5027e+06	6.1672	-6.0287	2.2122e+06	5.4514
AR NN	-1.3516	1.1465e+09	2.3514	-0.8002	1.0701e+09	2.1946

Dans ce scénario on voit que l'algorithme de réseau de neurones fournit une valeur très petite de SDR, car comme précédemment l'algorithme produit un signal d'enveloppe à peu près constant au cours du temps ( comme le montre la figure suivante ( on choisit les données du débit entrant par exemple vu qu'il présente à peu près les même performance que les données du débit sortant))



Le baseline fournit une prédiction très faible, par contre les deux autres algorithmes linéaires AR et ARMA donnent la meilleure prédiction, on va donc procéder de la même manière, et tracer le SDR en fonction de la distance de prédiction pour l'algorithme AR vu qu'il permet d'avoir la valeur la plus élevée de SDR, et un pourcentage d'erreur très petit.



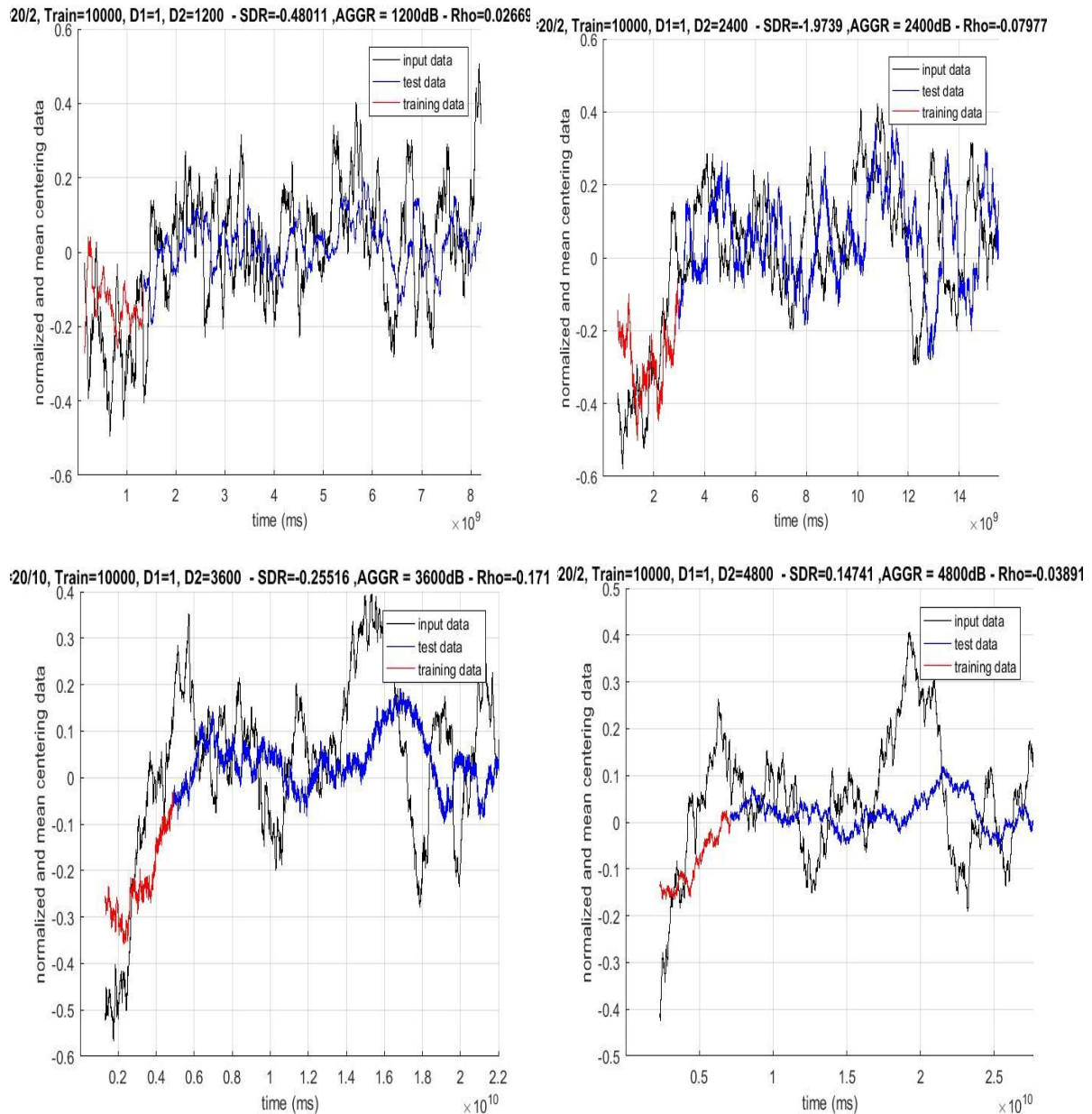
On remarque donc que l'algorithme AR ne permet pas d'avoir une bonne prédiction si on veut estimer à l'échelle de 10 minutes, la valeur maximale de la distance de prédiction qu'on peut atteindre est à peu près 6 minutes.

## Scénario 5

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-2.2846	1.2992e+06	2.2310	-2.2833	1.3016e+06	2.2352
AR	-1.0214	1.1257e+06	1.9332	-1.0140	1.1276e+06	1.9363
ARMA	-0.4801	1.0593e+06	1.8191	-0.4695	1.0612e+06	1.8223
NN	-7.9222	2.7897e+06	4.7905	-9.7437	3.4531e+06	5.9298
AR NN	-2.6691	1.2682e+09	1.8146	-2.8156	1.3117e+09	1.8768

On voit donc que tous ces algorithmes fournissent des valeurs négatives de SDR, et donc on ne peut pas avoir une bonne prédiction si on augmente la valeur de la distance de prédiction, et donc on peut penser à agréger les données on choisit donc l'algorithme ARMA qui possède la plus grande valeur du SDR pour faire augmenter la granularité et voir si on peut améliorer les performances prédictives.

les figures suivantes représentent le signal prédit obtenu, on choisit de ne représenter que le débit entrant comme précédemment, dans chaque étape nous allons augmenter la granularité de 2 min, il en est de même pour l'horizon de prédiction. Dans un premier temps on prend une granularité de 2 min, et on va prédire à l'échelle de 2 min.

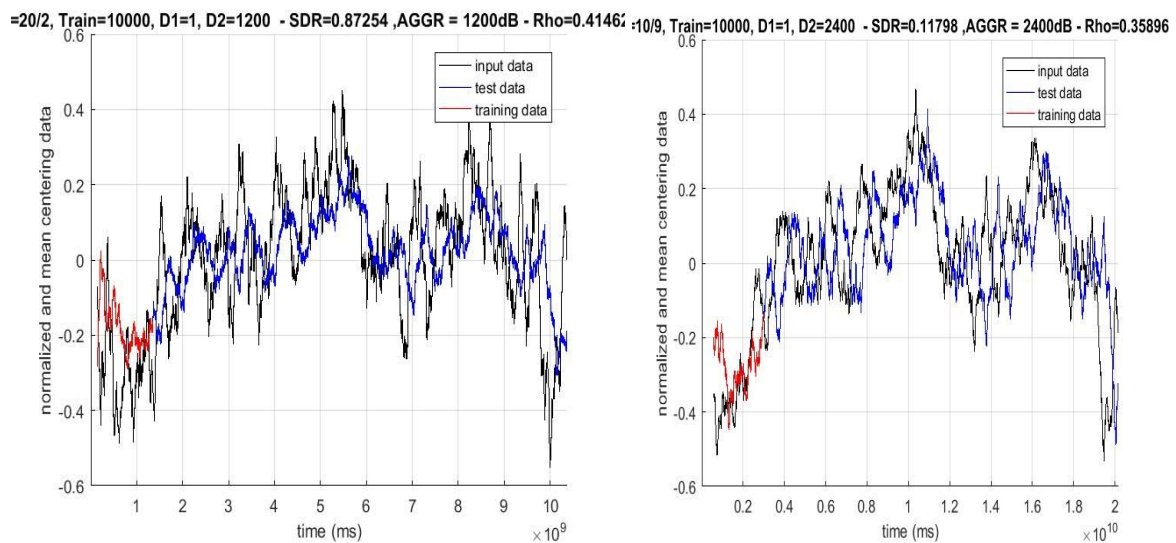


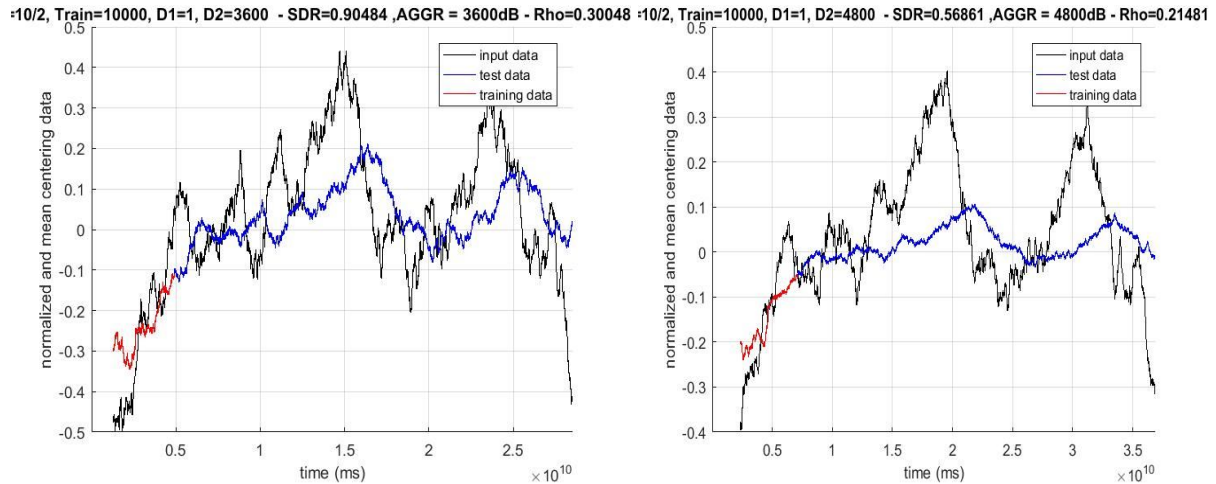
Dans les deux premiers cas où l'agrégat de flux est égale à 2 minutes et 4 minutes, le signal prédit suit éventuellement l'allure des données d'origine mais l'algorithme ne permet d'avoir la prédiction attendue, il fournit des valeurs très faibles de SDR, ce qui est pire dans le cas de l'augmentation de flux à 6, et 8 minutes ou le signal prédit est complètement différent au débit.

## Scénario 6

	Débit entrant	Débit entrant	Débit entrant	Débit sortant	Débit sortant	Débit sortant
Méthodes de prédiction	SDR	Erreur absolu	MAD	SDR	Erreur absolu	MAD
Baseline	-0.5829	1.2943e+06	2.1158	0.3860	1.1660e+06	1.9060
AR	0.4115	1.1647e+06	1.9039	-0.5934	1.2943e+06	2.1157
ARMA	0.8725	1.1135e+06	1.8202	0.8562	1.1149e+06	1.8225
NN	-9.3645	4.1748e+06	6.8244	-9.8620	4.4036e+06	7.1984
AR NN	-2.2716	1.1995e+09	1.7162	-1.7512	1.6681e+09	2.2710

Comme précédemment, la valeur du SDR pour les réseaux de neurones, et la méthode baseline, et AR NN est négative, par contre l'algorithme ARMA fournit comme même un résultat mais qui n'est pas satisfaisante, cette valeur de SDR est susceptible de décroître si on augmente l'horizon de prédiction, nous allons donc choisir cette méthode et agréger les données comme on a fait dans le scénario 5 pour visualiser les performances de l'algorithmes.





On peut donc tirer la même conclusion que dans le cas du scénario 5, dans tous les cas la méthode ARMA ne fournit pas de bon résultat.

#### **4-Conclusion.**

D'après ce qui précède, dans le cas d'un nombre fixe d'utilisateurs, on remarque que dans tous les scénarios, les performances de la méthode de base (baseline) ne sont pas satisfaisantes, car elle fournit des valeurs négatives pour le SDR, et des valeurs élevées pour le pourcentage de l'erreur. En revanche, dans le cas d'une granularité de 100 ms, les performances des autres méthodes sont très proches, mais ils ne fournissent pas de bonne prédiction, on était donc obligé à augmenter la granularité, pour faire face aux variations brutales des données. Et puis on a pu remarquer une augmentation de la valeur de SDR surtout pour la méthode AR NN qui nous a permis d'obtenir une forte corrélation entre le signal prédit et le signal de référence. Par contre, le fait de passer à une granularité égale à 10s, les performances des algorithmes ont dégradés, les algorithmes linéaires, et le réseau de neurones ne fonctionnent pas, tandis que la méthode hybride AR et réseau de neurones fournit une valeur très petites de SDR par rapport au cas d'une granularité de 1s, il en est de même pour une granularité de 1 min, et de 10 min, ceci peut être justifié par le fait qu'on a pas suffisamment de données pour faire une telle agrégation.

Par contre dans le cas d'un nombre variable d'utilisateurs, on constate que les algorithmes linéaires AR et ARMA sont plus efficaces en terme de performances par rapport aux réseaux de neurones et AR NN dans les scénarios 1 jusqu'à 4, par contre dans les scénarios 5 et 6 on a bien vu que toutes les méthodes fournissent des valeurs négatives ou bien quasiment nulles du SDR, il faut peut être envisager d'autres algorithmes d'apprentissages pour faire la prédiction.