

Juillet 2018



Rapport de Contrat

TéSA

par

**Urtzi Ayesta, Aurélien Delrieu, Riadh Dhaou, David Pradas,
Jean-Yves Tourneret et Herwig Wendt**

**MODÈLES DE PRÉDICTION DE TRAFIC
RÉFÉRENCE R-S17/TC-0008-031**

Table des matières

1	Introduction	1
2	Résultats sur débits instantanés agrégés	3
2.1	Méthodes de prédiction	4
2.1.1	Méthodes linéaires	4
2.1.2	Méthodes non-linéaires	5
2.1.3	Méthodes hybrides	5
2.1.4	Méthode de référence - “baseline”	5
2.2	Mesures de performances.	6
2.3	Estimation des paramètres par validation croisée	6
2.4	Résultats	7
3	Nouvelles traces	17
3.1	Description de la plateforme de test	18
3.1.1	Génération de trafic et types d’applications	19
3.1.2	Métriques collectées	20
3.2	Voie retour seulement	21
3.3	Prédiction croisée avec traces aller et retour	22
3.4	Prédictions avec la méthode AR vectorielle (VAR)	24
3.5	Prédiction croisée avec traces file d’attente et retour	25
4	Conclusion générale	27

Chapitre 1

Introduction

Dans une première partie de cette étude, nous avons fait un état de l'art des méthodes de prédiction de séries temporelles avec leurs application pour la prédiction de trafic réseau. A la fin de cette première partie, nous avons analysé les performances de prédiction obtenues à l'aide de séries temporelles simulées, ce qui nous a permis de sélectionner les méthodes qui semblent les plus pertinentes pour la prédiction du trafic réseau. L'objet de ce livrable est de présenter les résultats de prédiction obtenus sur des traces de trafic réelles. Les algorithmes choisis ont été détaillés dans le rapport de l'état de l'art (Lot 1) de cette étude. Nous résumerons leur fonctionnement lorsque cela s'avère nécessaire mais le lecteur est invité à consulter le rapport de l'état de l'art pour plus de détails.

Chapitre 2

Résultats sur débits instantanés agrégés

Dans un premier temps, nous avons étudié la sensibilité des algorithmes de prédiction au niveau d'agrégation et à la portée de la prédiction (appelée dans la suite distance de prédiction). Cette première étude, se fonde sur deux scénarios de trafic décrits ci-dessous

Sc. 1 Débit instantané échantillonné à 100 ms de flux agrégés simulé au niveau de la GW satellite (forward) pour un mélange de trafic pour 500 terminaux (28000 observations).

Sc. 2 Débit instantané échantillonné à 100 ms de flux agrégés simulé au niveau de la GW satellite (forward) pour un mélange de trafic pour 750 terminaux (28000 observations).

Les données ont été agrégées sur des fenêtres glissantes de tailles variables appartenant à l'ensemble $\{100, 200, 300, 400, 500\}$ ms.

La prédiction est ensuite effectuée pour la valeur des données agrégées à une distance de prédiction de 100, 200, ..., 1000 ms au delà de la dernière observation. Dans la suite, on ne représentera que les résultats obtenus pour une distance de prédiction supérieure à la taille des fenêtres d'agrégation. La figure suivante illustre le principe de l'agrégation et de la prédiction.

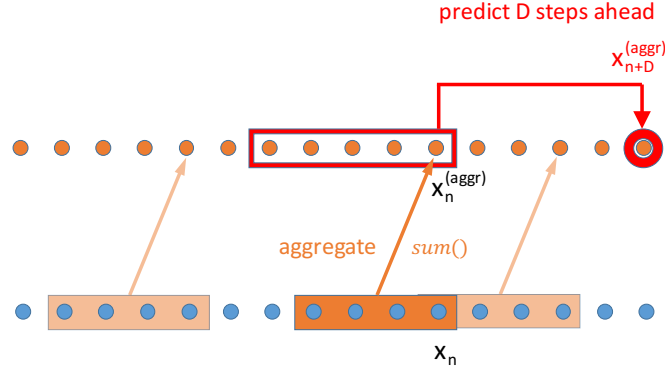


Schéma de prédiction à $D = 5$ pas dans le futur (distance de prédiction $D = 5$, correspondant à 500 ms dans le Sc. 1 et Sc. 2) pour des données agrégées sur des fenêtres de taille 4 (correspondant à 400 ms dans le Sc. 1 et Sc. 2).

2.1 Méthodes de prédiction

Nous rappelons ici brièvement les méthodes de prédiction utilisées dans la suite afin de rappeler les données d'entrée de ces méthodes et le vecteur de paramètres θ qui doit être réglé pour chaque méthode (se référer au rapport "ÉTAT DE L'ART SUR LES TECHNIQUES DE PRÉDICTION DE TRAFIC - RÉFÉRENCE R-S17/TC-0008-031" pour plus de détails).

2.1.1 Méthodes linéaires

Les méthodes linéaires AR et ARMA prennent en entrée des vecteurs de la forme $\mathbf{x}_n = (x_n, x_{n-1}, \dots, x_{n-p+1})^T$ et donnent en sortie la valeur prédite du débit x_{n+D} , où D est la distance de prédiction. Les paramètres libres de ces méthodes sont l'ordre p du modèle AR ou les ordres p et q du modèle ARMA, i.e., $\theta = p$ (pour le modèle AR) et $\theta = (p, q)$ pour le modèle ARMA.

Les modèles AR vectoriels, dénotés VAR dans la suite, prennent en entrée des matrices de forme $(\mathbf{x}_n, \mathbf{y}_n)^T$, construits à partir de deux séries temporelles x et y , et donnent en sortie la valeur prédite du vecteur $(x_{n+D}, y_{n+D})^T$. Comme pour les modèles AR, le seul paramètre à choisir est l'ordre du modèle, $\theta = p$. Nous n'avons pas considéré les modèles ARMA vectoriels car le fait de considérer une partie MA n'a pas donné d'amélioration

significative dans la prédiction.

2.1.2 Méthodes non-linéaires

Les réseaux de neurones (NN) et machines à vecteurs supports (SVM) fournissent des prédictions scalaires pour x_{n+D} à partir des données vectorielles d'entrée $\mathbf{x}_n = (x_n, x_n, \dots, x_{n-p+1})^T$ ou $(\mathbf{x}_n^T, \mathbf{y}_n^T)^T$.

Pour les réseaux de neurones, le vecteur de paramètres θ regroupe les éléments suivants

- la taille p des vecteurs d'entrée
- le nombre d'entrées N_E
- le nombre de couches L
- le nombre de neurones N_l , $l = 1, \dots, L$, pour chaque couche.

Nous considérons des machines à vecteurs supports avec un noyau polynomial d'ordre K , ou un noyau gaussien de largeur σ . Les paramètres θ à régler sont donc

- la taille p des vecteurs d'entrée
- les paramètres K ou σ du noyau
- les paramètres C (borne sur les multiplieurs de Lagrange), λ (paramètre de conditionnement) et ϵ (largeur de tube) de l'algorithme SVM.

2.1.3 Méthodes hybrides

Deux méthodes hybrides ont été considérées dans cette étude : un modèle AR suivi d'un réseau de neurones, et un modèle AR suivi d'une machine à vecteurs supports. Les paramètres de la partie linéaire et non-linéaire des méthodes hybrides sont réglés de façon indépendante.

2.1.4 Méthode de référence - “baseline”

Pour obtenir une idée de la difficulté de la tâche de prédiction, nous calculons également une prédiction “naive”, que nous appelons approche “baseline”, qui consiste à utiliser l'observation précédente comme prédiction, i.e., telle que

$$\hat{x}_n = x_{n-1}.$$

Il n'y a pas de paramètre à régler pour cette méthode.

2.2 Mesures de performances.

Pour quantifier et comparer la qualité des prédictions, nous proposons d'utiliser les mesures suivantes

- le coefficient de corrélation de Pearson pour le signal x et sa prédiction \hat{x}

$$\rho = \frac{(\mathbb{E}[x] - \mu_x)(\mathbb{E}[\hat{x}] - \mu_{\hat{x}})}{\sigma_x \sigma_{\hat{x}}}.$$

Plus ce coefficient ρ est proche de 1, meilleure est la prédiction ;

- le rapport signal-distortion (SDR) défini comme

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

qui sont deux mesures couramment utilisées pour évaluer les performances d'une méthode de prédiction. Nous étudions également la différence entre la valeur de SDR de la prédiction considérée et de la prédiction "baseline" (qui est parfois utilisée comme méthode de référence) : plus cette différence est grande, meilleur est le gain obtenu entre cette méthode de prédiction et la méthode de référence "naïve".

Finalement, pour les scénarios simulés plus réalistes qui seront traité dans le chapitre suivant, nous proposons également de quantifier l'erreur de prédiction en utilisant la déviation absolue moyenne normalisée entre le signal x et sa prédiction \hat{x} (en %)

$$\text{MAD} = 100 \left(\frac{\|x - \hat{x}\|_1}{\|x\|_1} \right)$$

où $\|x\|_1 = \sum_{i=1}^n |x_i|$.

2.3 Estimation des paramètres par validation croisée

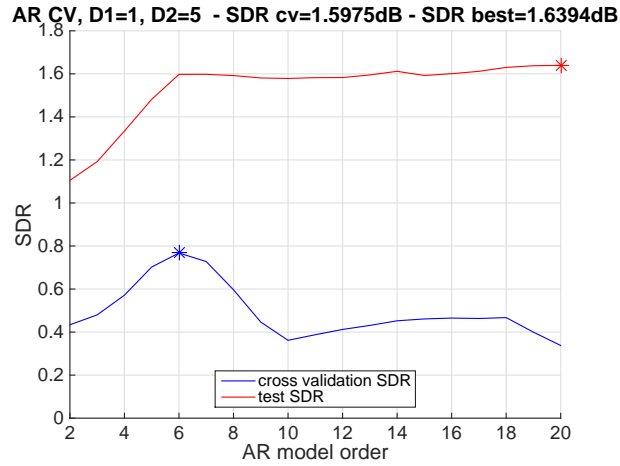
Nous supposons dans la suite que nous disposons de $N_{train} = 500$ échantillons pour estimer / apprendre les modèles de prédiction, et notre tâche consiste à prédire les $N_{test} = N - N_{train} = 27500$ échantillons restants supposés inconnus. Il est connu que la qualité de la prédiction peut varier fortement avec le réglage du vecteur de paramètres θ de la méthode de prédiction. Pour cette raison, nous proposons d'estimer θ en utilisant une procédure de validation croisée sur l'ensemble de $N_{train} = 500$ échantillons supposés connus :

1. Les premiers $N_{train,cv} = 250$ échantillons des $N_{train} = 500$ échantillons qui sont disponibles sont utilisés pour estimer / apprendre les modèles de prédiction pour un large ensemble de valeurs possibles de θ .

2. Les modèles obtenus pour différentes valeurs de θ sont testés sur les $N_{test,cv} = 250$ derniers échantillons disponibles.
3. Les paramètres θ^* pour laquelle la valeur la plus élevée de SDR est obtenu est retenu comme estimé pour la valeur optimale de θ .

Le modèle final est ensuite estimé / appris sur les $N_{train} = 500$ valeurs de débits disponibles en utilisant θ^* .

Le principe de la validation croisée est illustré dans la figure suivante.



Validation croisée de l'ordre d'un modèle AR pour le premier scénario (Sc. 1) : la courbe bleue correspond à la valeur de SDR obtenu avec les 250 débits issus de la validation croisée. Cette courbe est maximale pour le modèle d'ordre $p = 6$, qui est donc retenu par la procédure CV. La courbe en rouge montre le SDR qui est obtenu sur les 27500 débits qui ne font pas partie des données d'entraînement. On observe que la valeur $p = 6$ estimée par CV fournit une valeur de SDR proche de la valeur optimale pour ces données, comme espéré.

2.4 Résultats

Les figures présentées sur les quatre pages suivantes résument les résultats obtenus pour les deux scénarios (Sc. 1 pages 1-2 et Sc. 2 pages 3-4) et sont organisées de la façon suivante :

En haut sont représentées les données (à gauche), un zoom sur les données (au centre) et la fonction d'auto-corrélation (à droite).

Les lignes suivantes montrent les mesures de performance obtenues avec les différentes méthodes de prédiction : SDR (à gauche), coefficient de corrélation (au centre) et différence de SDR entre la méthode de prédiction considérée et la méthode "baseline" (à droite).

Chaque courbe correspond à un niveau d'agrégation différent et est représentée en fonction de la distance de prédiction D .

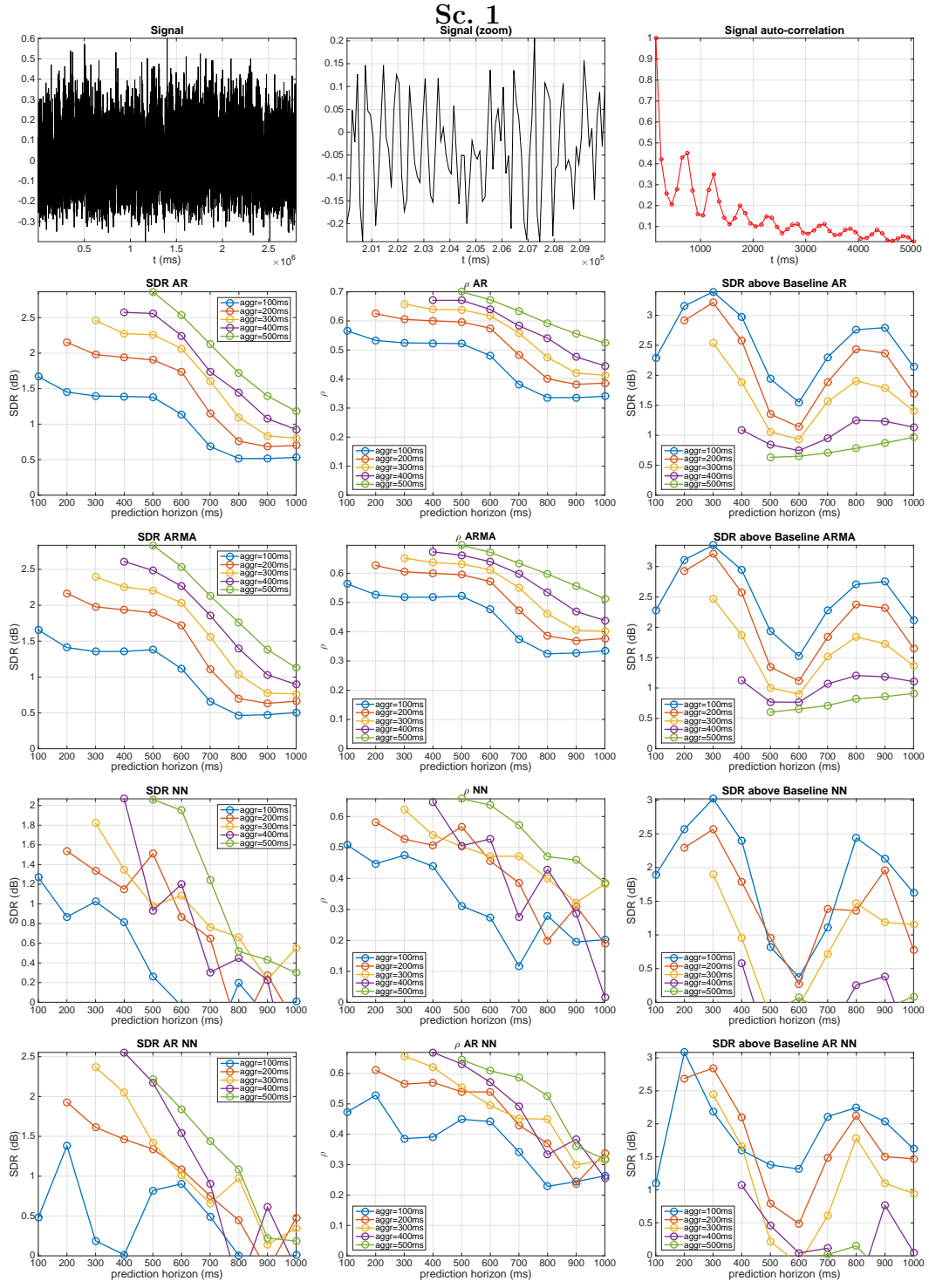
Les résultats amènent aux conclusions générales suivantes.

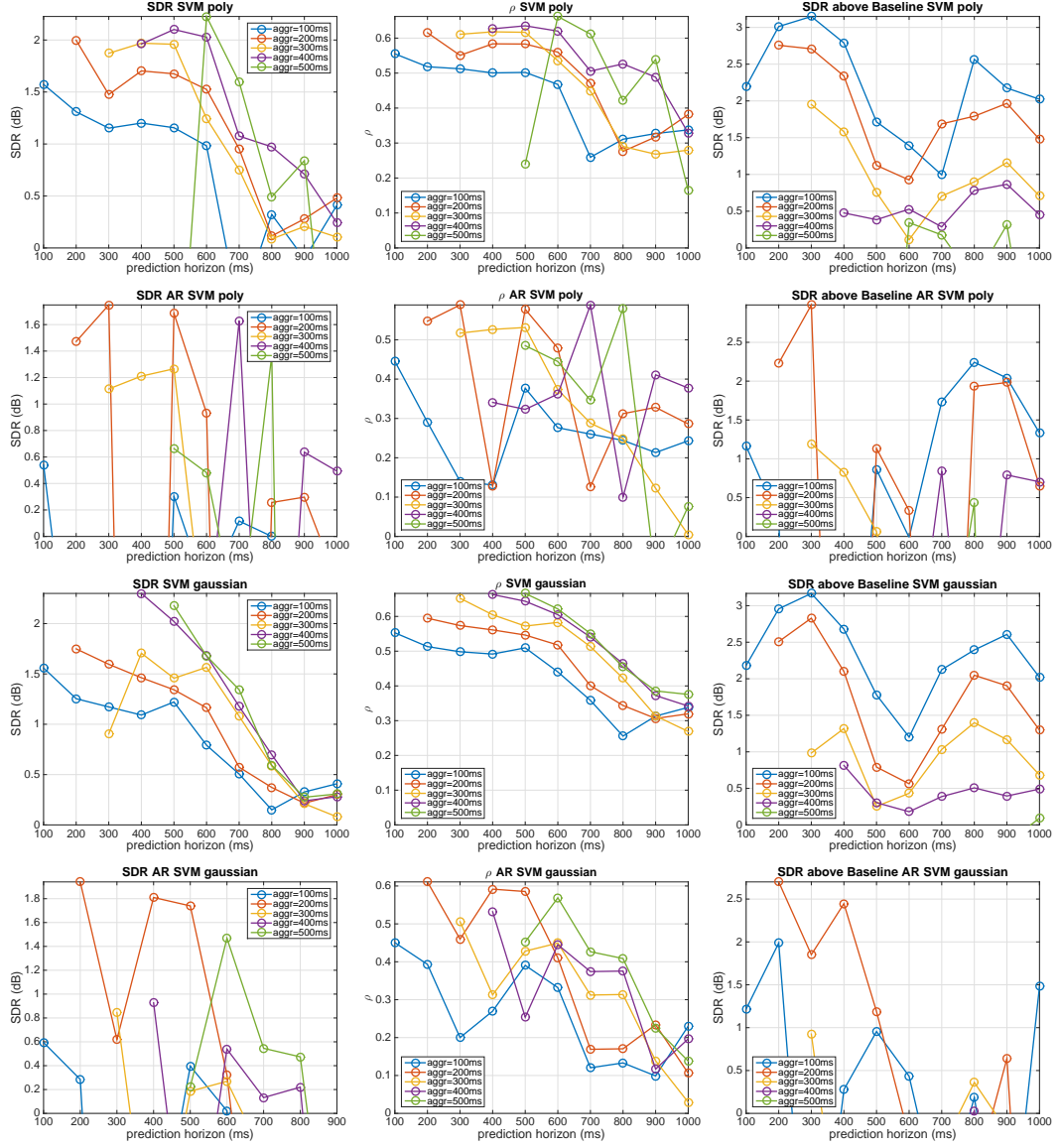
- La fonction de corrélation est caractérisée par des maxima locaux aux multiples de 600 ms, ce qui correspond aux multiples d'un RTT. On en déduit que la prédiction sera plus facile pour une distance de prédiction en dessous de 600 ms qu'au delà de 600 ms (et de même, plus facile en dessous d'un multiple $D \leq k \cdot 600$ ms qu'en dessus) : En effet, pour une distance de prédiction de $D < 600$ ms, on disposera d'une observation à $600 - D$ ms dans le passé qui est fortement corrélée avec la valeur à prédire.
- Le problème de prédiction est d'autant plus facile que les fenêtres d'agrégation sont longues : en effet, plus les données sont agrégées, plus les valeurs de SDR et de ρ sont élevées, et ce pour toutes les méthodes considérées.
- La performance de prédiction *relative* par rapport à la performance d'une méthode naïve "baseline" (colonne de droite) est plus élevée pour les données avec un niveau d'agrégation faible (de 100 ms).
- La prédiction est plus difficile quand la distance de prédiction D est grand.
- La prédiction est plus facile pour le premier scénario Sc. 1 que pour le second Sc. 2.

En ce qui concerne les performances relatives entre les différentes méthodes de prédiction, nous obtenons les conclusions suivantes.

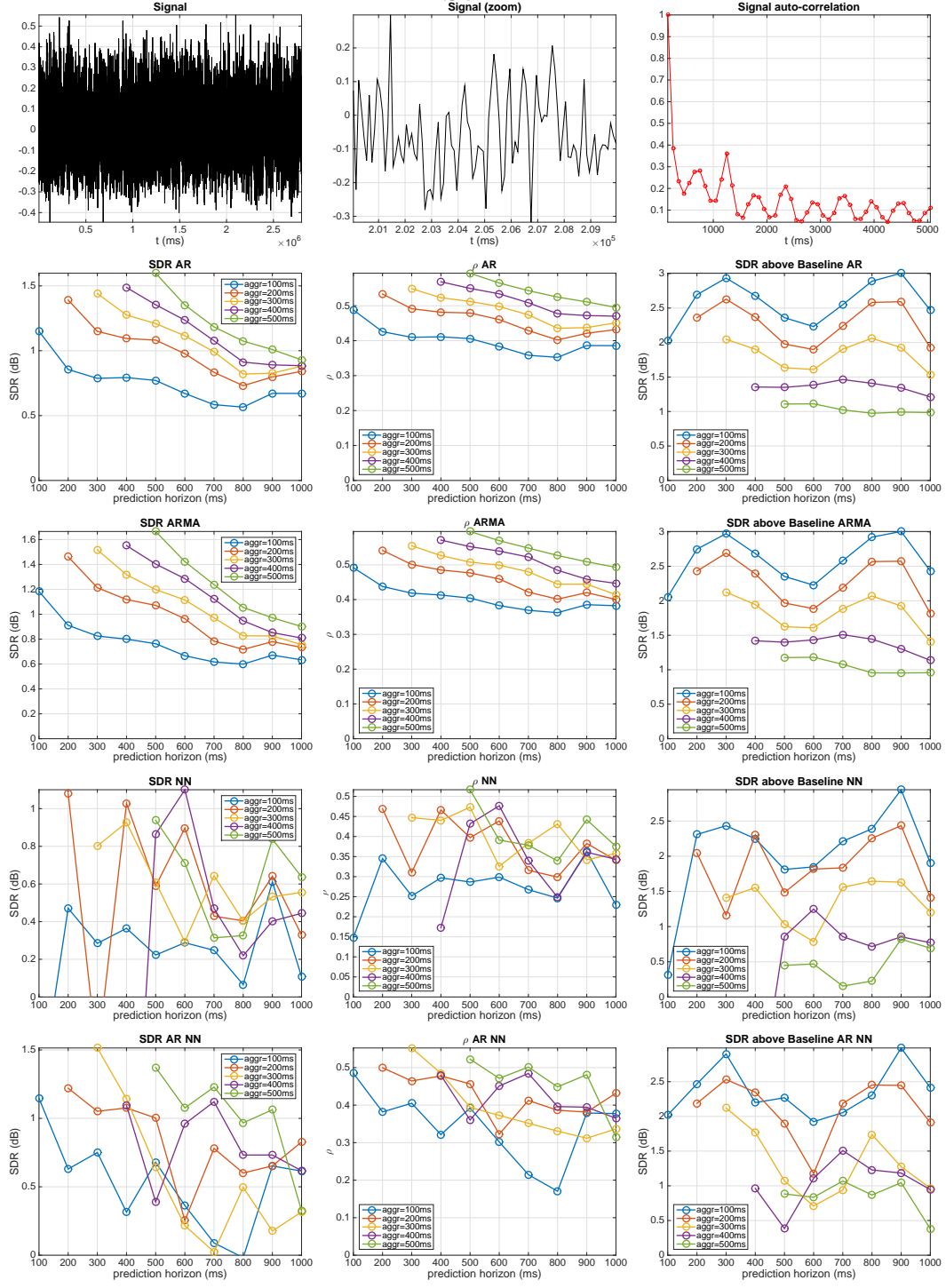
- Les modèles AR et ARMA donnent des très bons résultats avec des valeurs de SDR dépassant de 3 dB celles obtenues avec la prédiction "baseline".
- La méthode NN donne des résultats similaires aux modèles AR et ARMA. Cependant, les résultats obtenus avec les réseaux de neurones sont plus difficiles à interpréter à cause du nombre élevé de paramètres à estimer, ce qui amène à des courbes de performances moins lisses et cohérentes (nous rappelons que les paramètres du réseau sont estimés indépendamment pour chaque point des courbes).
- SVM donne aussi de bons résultats, mais le réglage des paramètres s'avère encore plus délicat et coûteux. La même conclusion est obtenue pour les méthodes hybrides.

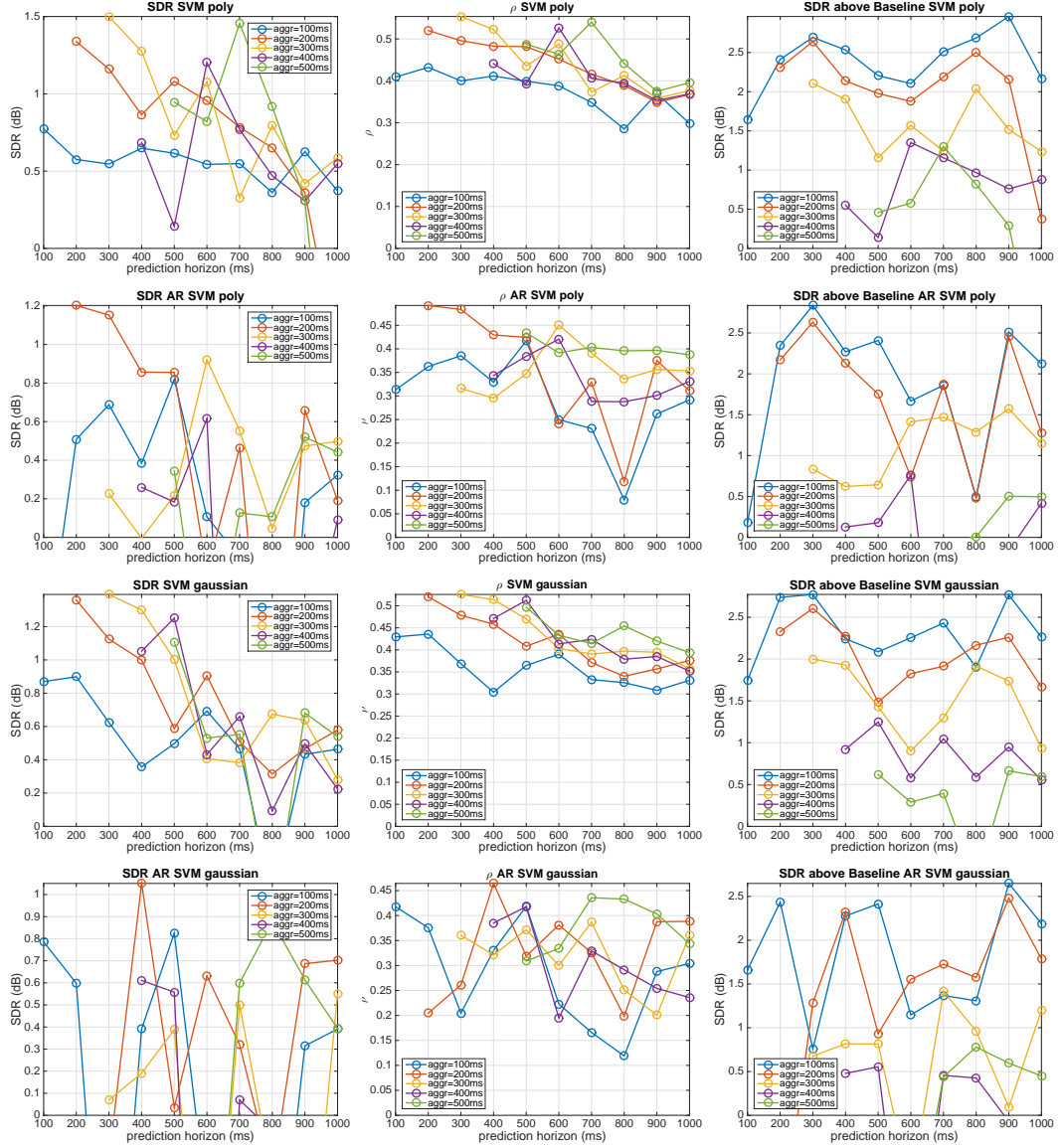
Nous en concluons que les modèles AR et ARMA ainsi que les réseaux de neurones sont les méthodes les plus prometteuses pour la suite de l'étude.





Sc. 2



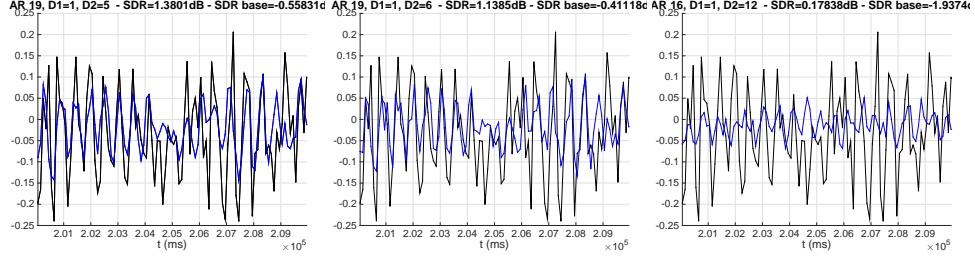


Afin de pouvoir apprécier l'allure des prédictions dont la qualité a été quantifié et comparée dans les figures précédents, les deux pages suivantes présentent des exemples de prédictions obtenus avec la méthode AR pour les deux scénarios (Sc. 1 et Sc. 2) et pour une distance de prédiction égal à $D = 5$ (colonnes de gauche), $D = 6$ (colonnes du centre) et $D = 12$ (colonnes de droite), pour des niveaux d'agrégation allant de 100 à 500 ms (de haut en bas). Les valeurs $D = 6$ et $D = 12$ ont été choisies car elles correspondent à un et à deux RTT, respectivement.

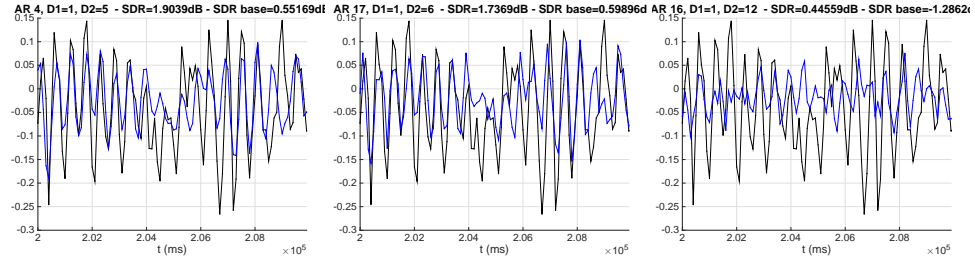
Sc. 1 : illustration des prédictions obtenues avec la méthode AR

 $D = 5$ $D = 6$ $D = 12$

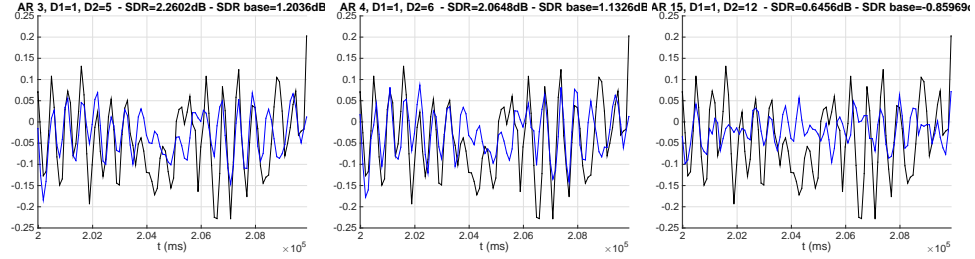
aggr=100ms



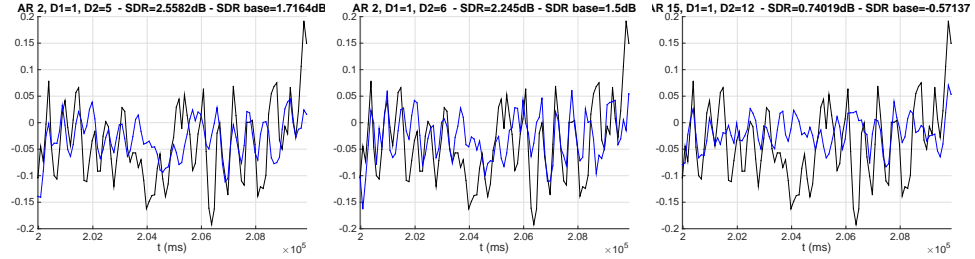
aggr=200ms



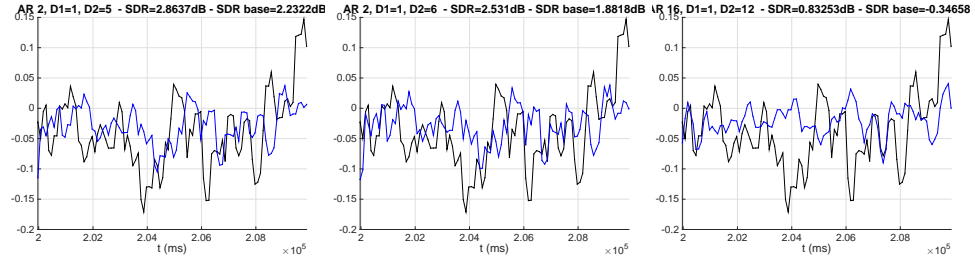
aggr=300ms



aggr=400ms



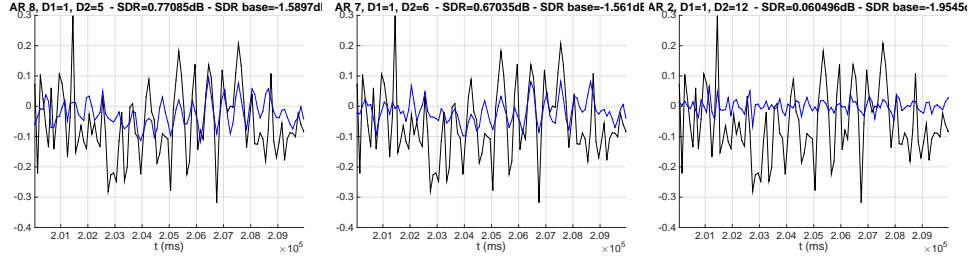
aggr=500ms



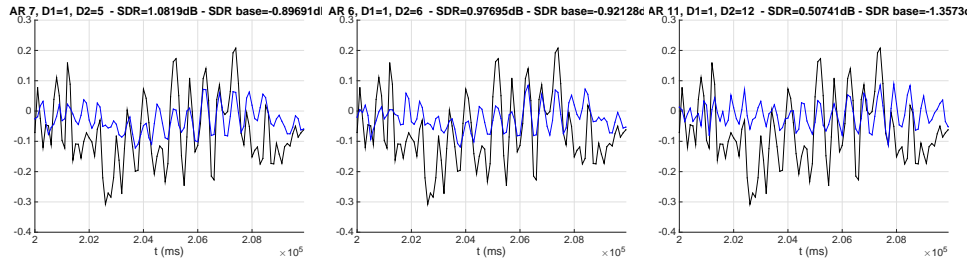
Sc. 2 : illustration des prédictions obtenues avec la méthode AR

 $D = 5$ $D = 6$ $D = 12$

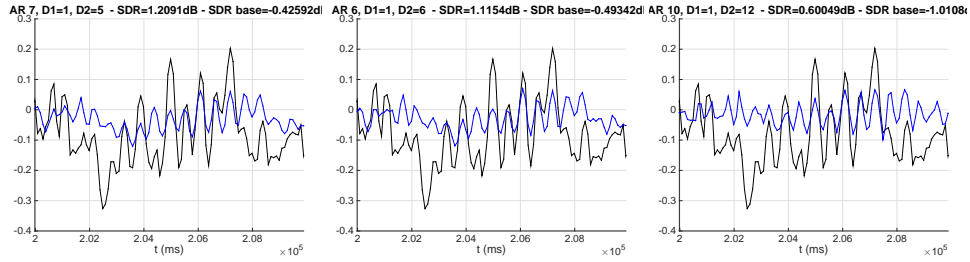
aggr=100ms



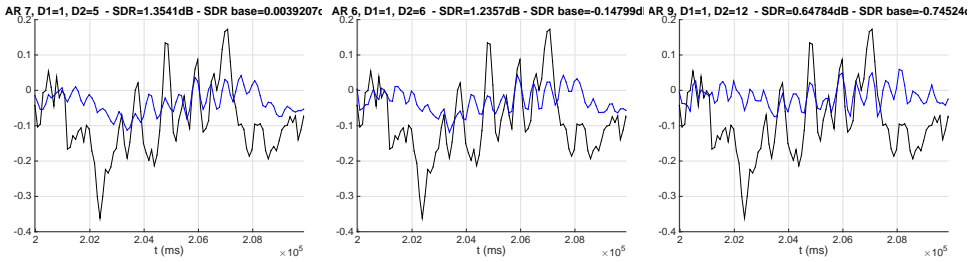
aggr=200ms



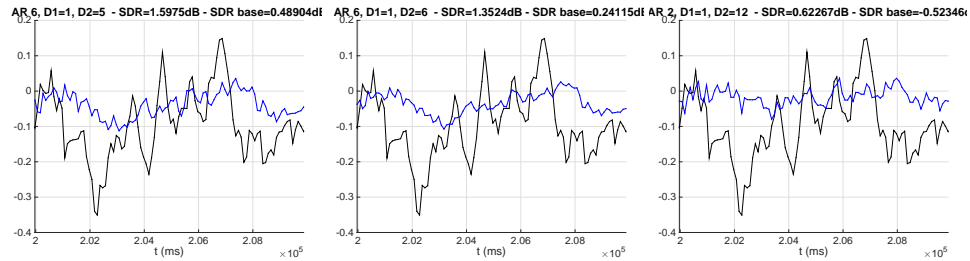
aggr=300ms



aggr=400ms



aggr=500ms



Chapitre 3

Applications des méthodes sélectionnées aux nouvelles traces

Nous utilisons deux méthodes représentatives des techniques linéaires et non-linéaires basées respectivement sur le modèle AR et les réseaux neuronaux. Il avait été conclu dans le rapport du lot 1 [1] que la méthode linéaire AR donnait des résultats satisfaisants et avait l'avantage d'être simple d'utilisation avec une complexité limitée permettant d'obtenir des temps de calcul très satisfaisants.

Dans notre étude, nous analysons également l'utilité d'utiliser des prédictions croisées, c'est à dire, utiliser les traces de débit réalisées sur la voie aller et la taille de la file d'attente, spécifiques à un terminal, pour améliorer la prédiction de la voie retour. Pour ce cas particulier, la méthode AR a été écartée, ne pouvant pas prédire l'état d'une variable à partir d'une autre variable (par exemple, elle ne peut effectuer une prédiction du débit sur la voie retour à partir de celui de la voie aller). Après plusieurs essais, nous avons décidé d'utiliser la méthode basée sur les réseaux de neurones et avons écarté les méthodes basées sur les machines à vecteurs supports (méthode SVM). D'un côté, comme présenté précédemment, la méthode SVM requiert un choix de paramètres large et complexe. De plus, avec les réseaux de neurones, il semble assez simple de trouver une structure raisonnablement performante par validation croisée. Nous avons testé l'utilisation de 2, 3 et 4 couches, chaque couche étant constitué d'un nombre de neurones variant entre 2 et une dizaine. La structure du réseau retenue correspond à 3 couches, avec un faible nombre de neurones. Sur ce point, il y a une

incohérence apparente avec la conclusion du rapport de lot 1 [1]. Cela s’explique par le fait que dans le rapport [1] nous avons fait seulement une partie des expérimentations avec des paramètres choisis de façon adéquate (par validation croisée) pour les différents méthodes. Après une étude plus systématique, Il s’avère que la méthode SVM est très sensible à ce choix des paramètres. Cette méthode pourrait être améliorée en effectuant une validation croisée plus fine. Mais cette procédure de validation croisée nécessiterait des temps de calcul prohibitifs. En outre, on constate que le réseau de neurones donne des résultats assez bons pour l’ensemble des tests que nous avons réalisés.

Dans une dernière partie de l’étude, nous avons étudié l’intérêt potentiel d’une méthode de prédiction de type AR vectorielle, permettant de prédire la voie aller (ou la voie retour) en fonction de mesures faites soit conjointement ou séparément sur les voies aller et retour.

3.1 Description de la plateforme de test

Dans cette section, nous décrivons la plateforme de tests que nous avons mis en place pour évaluer les performances des différentes méthodes de prédiction. La plateforme est orchestrée par le banc d’ingénierie OpenBACH (v1.0), permettant l’automatisation des tests via différents scénarios.

Ces scénarios permettent de :

- Lancer l’émulateur satellite OpenSAND avec la topologie de la Figure 3.1, comprenant un terminal ST1, une gateway GW et le satellite SAT. Différentes WorkStations (WS) sont disponibles dans le sous-réseau du ST1. L’accès à internet se fait via la GW, i.e., tout le trafic à destination/source l’Internet doit passer par OpenSAND.
- Préparer les routes afin que les Work Stations (WS) puissent communiquer via le réseau satellite et pour qu’elles aient accès à Internet.
- Lancer la capture du trafic (envoyé et reçu) sur les différentes interfaces “opensand-tun” de la GW et du ST (fichiers pcap)
- Génération de trafic sous deux formes
 - En tâche 1 de cette étude : Un scénario OpenBACH permettant de lancer différents services/applications (VoIP, Visioconférence, navigation web, vidéo streaming, etc...) de façon automatisée.
 - En tâche 2 de cette étude : Des personnes physiques derrière les WS permettant de générer du trafic plus réaliste (Netflix, Skype, emails, navigation web, Deezer,

Youtube, etc...).

- Nettoyer les routes et arrêter OpenSAND à la fin des tests.

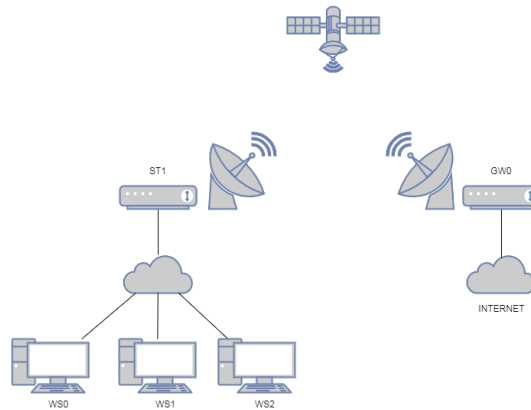


FIGURE 3.1 – Plateforme utilisée pour les tests.

Le terminal satellite ST1 est configuré avec une allocation constante de type CRA (Constant Rate Assignment) de 100 kb/s et une allocation dynamique maximal de type RBDC (Rate-Based Dynamic Capacity) de 5120 kb/s.

3.1.1 Génération de trafic et types d'applications

Pour cette tâche 2, nous proposons de générer du trafic de façon plus proche de la réalité qu'en tâche 1. Pour cela, nous avons placé de vrais utilisateurs derrière les WS0/1/2 permettant de générer des flux de différentes natures et plus représentatifs.

Voici le détail des applications :

- WS0 : L'utilisateur regarde une série Netflix (en qualité SD). Type de flux : streaming HTTP adaptatif.
- WS1 : Navigation web (site d'achat sur Internet type Amazon), envoi d'emails, et deux appels Skype.
- WS2 : Visualisation de vidéos sur Youtube pendant 10 min (résumé des match de la Coupe du Monde), puis navigation web (site l'Équipe), puis écoute de musique sur Deezer.

3.1.2 Métriques collectées

Pour chaque expérimentation, nous avons collecté les métriques suivantes :

- Sur ST1 :
 - Taille de la file BE (kb et paquets) - voie retour.
 - Les requêtes RBDC (kb/s) du terminal.
 - L'allocation total (CRA+RBDC) donnée au terminal satellite (kb/s)
- Sur GW : Taille de la queue BE (kb et paquets) - voie aller.
- Sur SAT : Le débit (kb/s) de la voie aller ainsi que de la voie retour.

Nous avons collecté environ 40000 échantillons pour chaque métrique, avec une période d'échantillonnage de 50ms. Des exemples de métriques de débit et de taille de file sont montrés dans la Figure 3.2. Nous avons également collecté les requêtes de capacité envoyées par le terminal ainsi que l'allocation réelle allouée au terminal (voir Figure 3.3). Cette figure montre que le terminal peut dépasser les requêtes de capacité max (5120 kb/s) car seulement la GW est au courant de cette limitation. En revanche, nous voyons que l'allocation réelle donnée par la GW ne dépasse pas le CRA+RBDCmax. Un léger dépassement peut être observé au niveau de l'allocation car la GW alloue avec une granularité en timeslots (de taille 67 kb/s pour le MODCOD plus robuste), et elle est donc obligée de lui allouer un timeslot entier même si le ST n'a demandé qu'une petite partie.

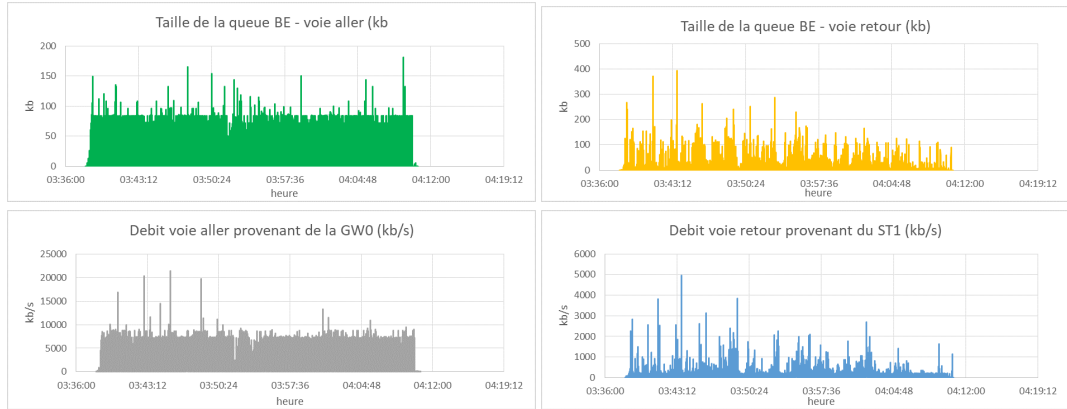


FIGURE 3.2 – Métriques collectées : débits et tailles de file.

Remarque : Sur les mesures de débit au niveau du SAT, le temps de propagation étant appliqué à la transmission et à la réception des équipements terrestres (ST et GW), 125 ms sont déjà appliquées lors de la mesure des débits au niveau du satellite.

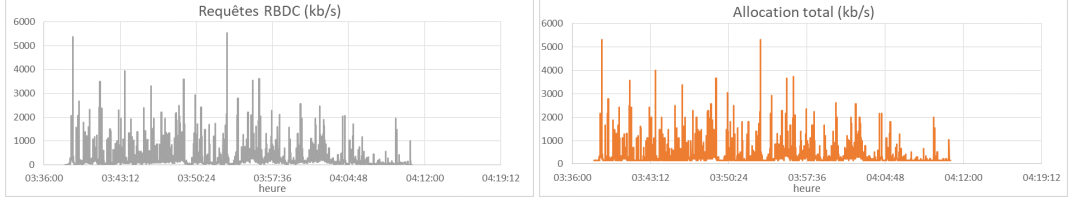
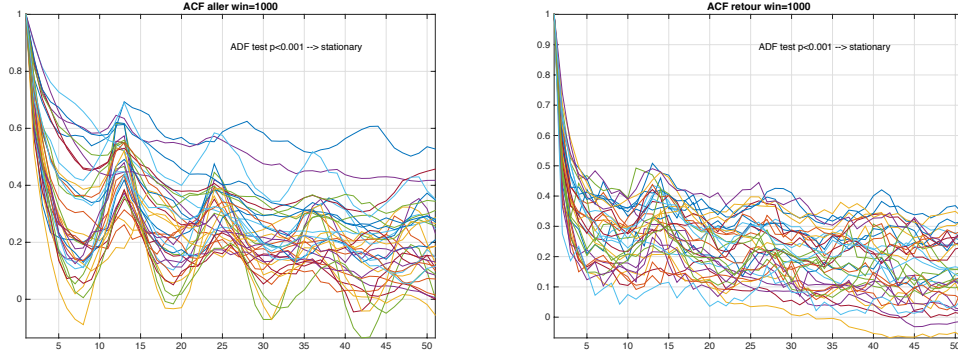


FIGURE 3.3 – Métriques collectées : allocation et requêtes du ST1

Analyse de stationarité : débit voie aller et débit voie retour

Les figures représentées ci-dessous ont été produites pour les voies aller (gauche) et retour (droite) qui montrent les fonctions d'auto-correlation estimées sur des fenêtres de longueur 1000. Le fait que les différentes courbes se superposent assez bien est un indicateur heuristique de la stationnarité des signaux considérés. Les résultats obtenus à l'aide d'un test de stationnarité classique - Augmented Dickey-Fuller (ADF) test - rejettent l'hypothèse selon laquelle les séries temporelles sont non-stationnaires et confirment donc cette observation. Cette hypothèse de stationnarité est importante car elle est requise pour certaines méthodes de prédiction comme la méthode de prédiction autorégressive (méthode AR).



3.2 Voie retour seulement

Dans un premier temps, nous analysons la qualité des prédictions via l'évolution temporelle du débit sur la voie retour seulement (voir Tables 3.1 et 3.2). Le tuple $(5000, aggr = 0, D = 1)$ indique que nous avons utilisé 5000 échantillons pour faire l'entraînement, le niveau d'agrégation, et D indique la distance temporelle (ou horizon de prédiction) à laquelle on prédit. Le terme “embedding dimension” indique la longueur des vecteurs utilisés pour

la prédiction.

Embedding dimension 12			
(taille entr, agrég, distance)	prediction	prediction	valeur précédente
(5000, aggr=0, D=1) :	1.96		0.6
(5000, aggr=0, D=6) :	1.2		-1.4
(5000, aggr=0, D=12) :	1.2		-1.15
(5000, aggr=3, D=6) :	1.9		-0.4
(5000, aggr=3, D=12) :	1.8		-0.1
Embedding dimension 24			
(5000, aggr=0, D=12)	1.12		-1.15
(5000, aggr=3, D=12)	1.8		-0.1

TABLE 3.1 – Prédictions avec voie retour seulement et des réseaux neuronaux.

3.3 Prédiction croisée avec traces aller et retour

Nous avons analysé la possibilité d'améliorer la prédiction en utilisant les traces de débit dans le sens aller (le débit étant agrégé sur une fenêtre temporelle donnée par le paramètre aggr où l'unité représente une centaine de millisecondes). Dans la Figure 3.4, nous analysons les corrélations entre les signaux sur les voies aller et retour. Ces traces ont été capturées au niveau du satellite, mais puisque que notre but est de faire une prédiction au niveau de la gateway, nous décalons la voie aller d'un demi RTT. La figure montre que les deux traces sont corrélées avec une périodicité égale au RTT (corrélation par fenêtre de RTT). Comme le montre cette figure, la corrélation est maximale à l'instant 0, tout décalage entre la voie aller et la voie retour décale le point qui donne la valeur maximale.

Dans les Tables 3.3 et 3.4, nous montrons les résultats des prédictions que nous avons obtenus.

Embedding dimension 12			
(taille entr, agrég, distance)	prediction	prediction	valeur précédente
(5000, aggr=0, D=6) :	1.1		-1.4
(5000, aggr=0, D=12) :	1.0		-1.16
(5000, aggr=3, D=6) :	1.8		-0.4
(5000, aggr=3, D=12) :	1.6		-0.1

TABLE 3.2 – Prédictions avec voie retour seulement et méthode AR.

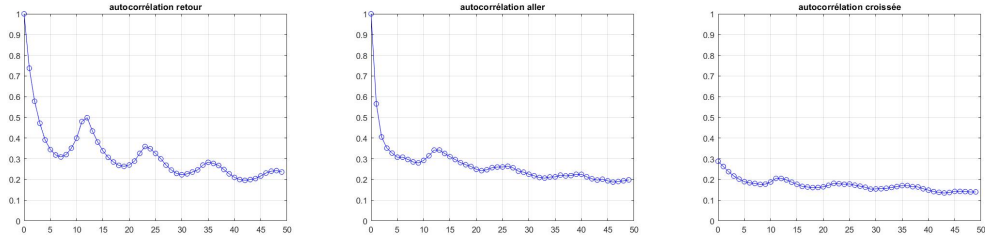


FIGURE 3.4 – Corrélation entre les traces aller et retour.

Embedding dimension 12	
(5000, aggr=0, D=1) :	1.9 0.6
(5000, aggr=0, D=6) :	0.5 -1.4
(5000, aggr=0, D=12) :	0.7 -1.15
(5000, aggr=3, D=6) :	0.6 -0.4
(5000, aggr=3, D=12) :	1.7 -0.1
Embedding dimension 6	
(5000, aggr=0, D=12)	1.1 -1.16 R only
(5000, aggr=0, D=12)	1.2 -1.16 R+A

TABLE 3.3 – Prédiction avec voie aller et retour et des réseaux neuronaux.

Embedding dimension 12	
(5000, aggr=0, D=1) :	1.96 0.6
(5000, aggr=0, D=6) :	1.2 -1.4
(5000, aggr=0, D=12) :	1.1 -1.15
(5000, aggr=3, D=6) :	2.0 -0.4
(5000, aggr=3, D=12) :	1.6 -0.1
Embedding dimension 24	
(5000, aggr=0, D=12)	0.9 -1.15
(5000, aggr=3, D=12)	1.6 -0.1

TABLE 3.4 – Prédiction avec voie aller et retour et des réseaux neuronaux. Signal décalé d'un demi RTT.

Taille du modèle 12						
Method	VAR				Basic	
voie prédite (entrée)	$P_A(A)$	$P_R(R)$	$P_A(A+R)$	$P_R(A+R)$	$P_A(A)$	$P_R(R)$
(5000, aggr=0, D=6) :	1.5 (78%)	1.1 (67%)	1.6 (76%)	1.4 (60%)	-1.4 (96%)	-1.4 (82%)
(5000, aggr=0, D=12) :	1.4 (79%)	1.0 (68%)	1.4 (78%)	1.1 (65%)	0.0 (73%)	-1.16 (76%)
(5000, aggr=3, D=1) :	11.7 (20%)	9.3 (21%)	11.7 (19%)	9.5 (19%)	8.2 (27%)	6.8 (25%)
(5000, aggr=3, D=6) :	2.0 (65%)	1.8 (53%)	2.1 (63%)	2.3 (46%)	-0.7 (82%)	-0.4 (64%)
(5000, aggr=3, D=12) :	1.8 (64%)	1.5 (54%)	1.8 (64%)	1.8 (52%)	0.7 (60%)	-0.1 (58%)
Taille du modèle 24						
voie prédite (entrée)	$P_A(A)$	$P_R(R)$	$P_A(A+R)$	$P_R(A+R)$	$P_A(A)$	$P_R(R)$
(5000, aggr=0, D=6) :	1.5 (78%)	1.1 (67%)	1.6 (76%)	1.4 (60%)	-1.4 (96%)	-1.4 (82%)
(5000, aggr=0, D=12) :	1.4 (79%)	1.0 (68%)	1.4 (78%)	1.1 (65%)	0.0 (73%)	-1.16 (76%)
(5000, aggr=3, D=1) :	11.7 (20%)	9.3 (21%)	11.7 (19%)	9.5 (19%)	8.2 (27%)	6.8 (25%)
(5000, aggr=3, D=6) :	2.0 (65%)	1.8 (53%)	2.1 (63%)	2.3 (46%)	-0.7 (82%)	-0.4 (64%)
(5000, aggr=3, D=12) :	1.8 (64%)	1.5 (54%)	1.8 (64%)	1.8 (52%)	0.7 (60%)	-0.1 (58%)

TABLE 3.5 – Prédictions avec les méthodes VAR et Basique.

3.4 Prédictions avec la méthode AR vectorielle (VAR)

Nous avons utilisé la méthode AR vectorielle pour prédire le débit sur la voie aller (A) ou la voie retour (R)

- soit uniquement en fonction du débit de la voie aller $P_X(A)$, ou de celui de la voie retour $P_X(R)$,
- soit en fonction des débits des voies aller et retour, utilisés conjointement $P_X(A+R)$.

Nous nous sommes placés à deux points d'observation différents :

- au niveau de la passerelle (pour prédire le débit de la voie retour)
- au niveau du terminal (pour prédire le débit de la voie aller).

Enfin, nous avons fait varier la valeur de l'horizon de prédiction de 6 à 12. La table 3.5 montre les résultats issus de ces expérimentations. Nous constatons que la dimension choisie pour le modèle (12 ou 24) n'a pas d'impact sur les résultats. De même, le point d'observation (terminal ou passerelle) n'a pas d'influence sur les résultats à horizon de prédiction équivalent. Nous avons également décalé dans le temps la voie retour par rapport à la voie aller d'un RTT, afin d'essayer de capturer l'impact potentiel des requêtes sur les réponses dans le sens inverse. Les résultats obtenus avec et sans décalage restent inchangés.

À partir de ces expérimentations, nous pouvons établir les conclusions suivantes

- Si on regarde le critère SDR, la méthode VAR donne de meilleurs prédictions que la méthode de base. Cette conclusion reste vraie si on regarde le critère Erreur moyenne, sauf dans quelques cas.
- L'agrégation améliore les résultats dans tous les cas
- La prise en compte conjointe des voies aller et retour améliore légèrement la prédiction sur chacune des deux voies
- L'amélioration de la prédiction apportée par la méthode VAR avec l'utilisation de l'aller et du retour est plus significative sur la voie retour (donc si on veut faire la prédiction coté GW) par rapport à une prédiction de la voie aller (coté terminal). Ceci est conforté par une prédiction sur la trame suivante, comme le montre le résultat obtenu pour les paramètres (5000, aggr=3, D=1).

Nous concluons que décaler le signal améliore la prédiction, mais les résultats obtenus sont comparable aux prédictions obtenues avec la voie retour seulement.

3.5 Prédiction croisée avec traces file d'attente et retour

Dans la Figure 3.5, nous analysons la corrélation entre les traces retour et celle de la file d'attente. Nous observons que la corrélation croisée est presque inexistante. En plus, nous voyons que l'autocorrélation de la file d'attente décroît de façon quasi-monotone. Au vu de ces résultats et du jeu de données utilisé, nous concluons que la file d'attente ne permet pas d'améliorer les prédictions du débit de la voie retour.

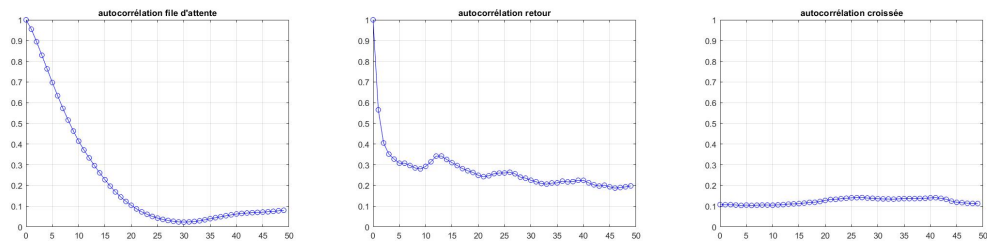


FIGURE 3.5 – Corrélation entre les traces retour et celle de la file d'attente

Chapitre 4

Conclusion générale

La sélection des méthodes de prédiction effectuée dans cette étude prend en compte les critères de simplicité d'utilisation, tout en laissant la possibilité de faire une prédiction croisée (à l'aide de la méthode AR vectorielle). Un gain en terme de précision des prédictions peut être observé lorsque le nombre de paramètres du modèle de prédiction augmente mais cette augmentation peut nécessiter une optimisation plus délicate.

Les méthodes testées sur les données réelles basées sur les modèles autorégressifs (AR) et les réseaux de neurones (NN) permettent d'apporter un gain significatif par rapport à une méthode naïve qui consiste à utiliser l'état courant pour prédire l'avenir.

Les prédictions effectuées sont fondées sur des traces de profils de trafics sans adaptation de l'allocation réalisée. Nous pourrions comparer les résultats de performances sur des méthodes d'allocation avec et sans prédiction de trafic. L'implémentation de la méthode AR et/ou ARMA pour influencer le calcul des requêtes de capacité du terminal satellite OpenSAND est donc envisagé pour la tâche 3. Cela permettrait d'évaluer l'impact de la rétroaction sur les profils de trafic et sur l'ensemble de la prédiction du trafic.

Partant des traces utilisées en tâche 2, nous avons évalué ci-dessous les éventuels besoins de traces pour la tâche 3. Il est clair que tout ne pourra être fait, il serait donc intéressant de fixer les priorités conjointement avec le CNES.

- Ajouter de trafics de différentes natures (BE, EF, AF) dans la voie retour et qui seront pris en compte par les files OpenSAND de cette même priorité. Nous ne pouvons pas modifier le DSCP/ToS pour les service de type Netflix/Skype/Deezer, car il est difficile de différencier ces applications par port ou type d'application/protocole (car encrypté ou pas connus). Nous pouvons néanmoins différencier/marker les services

par adresse source de la Work Station qui envoie/demande ce type de service. Nous proposons par exemple de faire que : la WS1 soit limitée à utiliser que du Netflix et du Deezer (trafic AF moyennement prioritaire), la WS2 fasse que des appels Skype (trafic EF prioritaire) et la WS3 fasse que de la navigation web, Youtube et envoi d'emails (trafic BE non-prioritaire).

- La prédiction a été jusqu'à présent faite au niveau paquet. Les prédictions croisées, effectuées au niveau paquet en taille de file, ne permettent pas d'améliorer la précision. Nous pourrions envisager de faire les prédictions sur des paramètres d'état de niveau flux (nombre, taille, débit). Comme indiqué précédemment, les flux sont souvent encryptés et les ports des applications sont très variables pour Netflix/Skype/Deezer, etc. Pour cela, il faudrait identifier les début/fin des flux (nous pouvons en effet noter ces dates à la main et/ou avec des outils de développement web pour les services utilisant un navigateur) puis analyser le pcap et extraire les informations par flux. Cette tâche peut devenir complexe s'il faut extraire les informations pour toutes les connexions/flux. Pour simplifier la tâche, nous pouvons revenir aux tests de tâche 1 avec des flux "non propriétaires" et maîtrisés (c-à-d : vidéo streaming HTTP adaptatif à la place de Netflix, un appel VoIP avec Twinkle à la place de Skype, navigation web sur des sites web locaux/maîtrisés à la place d'aller sur Internet), et le tout orchestré par OpenBACH.
- Réaliser des nouvelles simulations sur SNS3 pour obtenir de nouvelles traces de la voie aller. Il semble que le modèle 3GPPP est disponible sur SNS3. Cela permettrait de reproduire les traces générées avec SMACSAT en faisant varier certaines conditions. Attention au fait qu'il faut valider le fonctionnement du modèle.

Bibliographie

- [1] J. Y. Tourneret and H. Wendt, “Etat de l’art sur les méthodes de prédiction,” in *Rapport de Lot 1*, Toulouse, France, May 2018.