RAPPORT SUR LA TÂCHE T4

Méthodes à noyaux pour la prédiction des crues subites

K. Boukharouba, P. Roussel-Ragot, G. Dreyfus

1. INTRODUCTION

Le présent rapport porte sur la tâche T4 « Machines à vecteurs supports dynamiques », dont le responsable était le laboratoire SIGMA (SIGnaux, Modèles, Apprentissage statistique) de l'ESPCI ParisTech, partenaire 3 du consortium FLASH.

Cette tâche comportait deux sous-tâches :

- T4.1 : « The goals of this sub-task are (i) to improve the modeling of the hydro-meteorological chain using SVMs, and (ii) to increase the knowledge and improve the design methodology of kernel-based dynamical models ». C'est sur cette tâche qu'a porté l'essentiel de nos efforts. Nous avons modélisé la relation pluie-hauteur d'eau à l'aide de différents types de machines à vecteurs supports dynamiques. Ceci nous a conduit notamment à proposer une méthode originale de conception de modèle combinant classification non supervisée et régression non linéaire.
- T4.2 : « The issue that is addressed in this subtask is the integration of prior knowledge into dynamic support vector machines ». Plusieurs types de connaissances expertes ont été introduites dans nos modèles. La méthode de filtrage spatial des pluies est celle qui a permis l'amélioration la plus notable des performances de prédiction.

Les sections 2 et 3 présentent respectivement le bassin versant d'Anduze auquel ont été appliquées nos méthodes, et la base de données dont nous disposions. Les modèles dynamiques que nous avons mis en œuvre sont décrits dans la section 4 ; nous présentons essentiellement le modèle postulé (modèle NARX avec bruit d'état) et les modèles prédictifs qui en découlent. Ces derniers utilisent la régression à vecteurs supports (SVR), la régression Least-Squares Support Vector Machines (LSSVM), et la régression linéaire par morceaux (PieceWise linear Regression ou PWR). La modélisation, à l'aide des modèles précédents, de la relation pluie-hauteur d'eau du bassin versant du Gardon d'Anduze à l'exutoire d'Anduze est décrite dans la section 5. Cette section est divisée en trois sous-sections :

- la conception de modèles prédictifs globaux sous l'hypothèse NARX avec bruit d'état, et les résultats obtenus,
- la conception de modèles spécifiques par classification non supervisée des événements de crues, toujours sous l'hypothèse NARX avec bruit d'état, et les résultats obtenus,
- la conception de modèles prédictifs sous l'hypothèse NARX avec bruit de sortie et les résultats obtenus.

Enfin, la dernière section présente les conclusions et les perspectives ouvertes par cette étude.

2. LE BASSIN VERSANT D'ANDUZE

L'objectif de la tâche T4 était de mettre en œuvre des Machines à Vecteurs Supports (SVM pour Support Vector Machines) pour la prédiction des crues rapides, et de comparer leurs performances estimées avec celles des réseaux de neurones dynamiques qui étaient mis en œuvre par le partenaire 1 (EMA) du consortium. Il était donc indispensable que les deux partenaires travaillent sur les mêmes données.

Le choix s'est porté sur le bassin versant du Gardon d'Anduze (Figure 1), qui avait déjà fait l'objet d'une étude commune des partenaires. Ce système hydrologique est célèbre pour ses « épisodes cévenols », qui sont souvent meurtriers, et causent toujours des dégâts matériels coûteux.

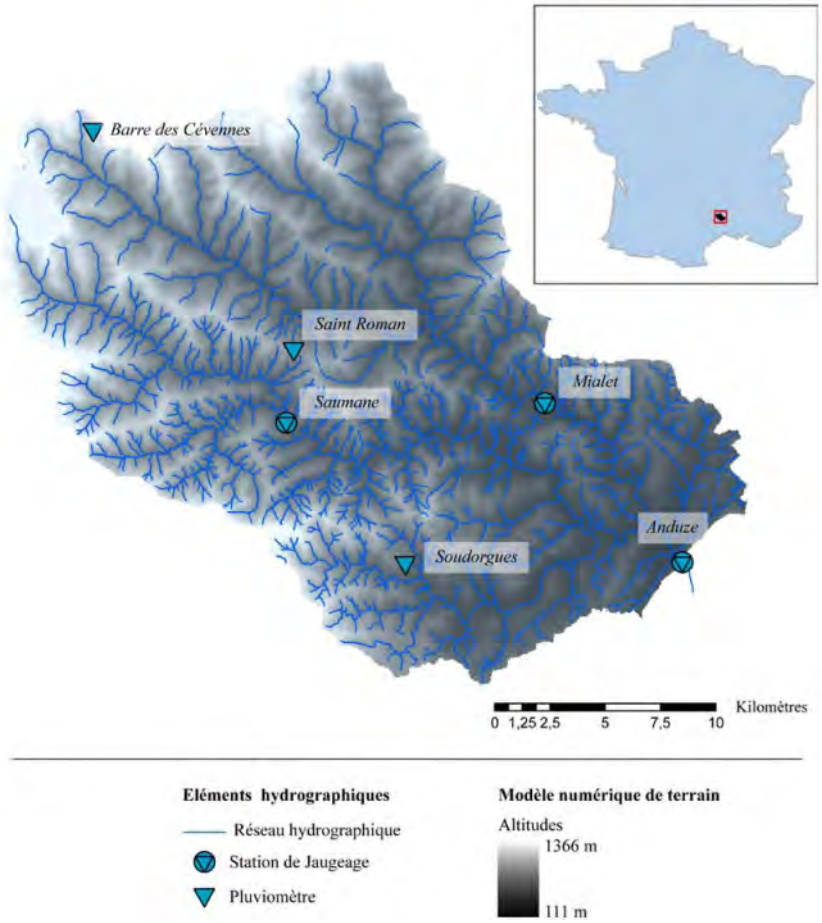

Figure 1. Le bassin versant du Gardon d'Anduze

Le bassin versant, d'une superficie de 524 km², comprend deux sous-bassins : le Gardon de Mialet et le Gardon de Saint Jean. Il est équipé de 6 pluviomètres, situés à Anduze, Barre des Cévennes, Mialet, Saint Roman, Saumane et Soudorgues. La hauteur d'eau qui doit être prédite par les modèles est mesurée par un limnimètre situé à l'exutoire du bassin versant, à Anduze.

3. BASE DE DONNÉES

3.1 ÉCHANTILLONNAGE

Depuis 2001, les données fournies par les pluviomètres et par le limnimètre sont échantillonnées avec une période de cinq minutes. La résolution de la mesure de hauteur de pluie est de 0,5 mm : en conséquence, même s'il pleut de façon constante, un pluviomètre n'enregistre pas de précipitation pendant une période d'échantillonnage si la hauteur de pluie tombée pendant cette période est inférieure à 0,5 mm. Pour réduire l'influence de ce bruit de quantification, six données consécutives fournies par chaque pluviomètre sont sommées pour fournir la

précipitation cumulée pendant 30 mn. De même, les mesures de hauteur d'eau fournies par le limnimètre sont échantillonnées avec une période de 30 mn.

3.2 DONNÉES UTILISÉES

Les données de précipitations et de hauteurs d'eau que nous avons utilisées dans notre étude ont été collectées de 1993 à 2008. Dix-sept événements significatifs ont été enregistrés depuis 1993 (Figure 2). Le Tableau 1 résume tous les événements utilisés pour l'apprentissage, la validation et le test. La date, la durée et le niveau d'eau maximum sont reportés. L'événement le plus intense de la base de données est l'événement n°19 (2002), dont le niveau d'eau maximum a atteint 9.7 m.

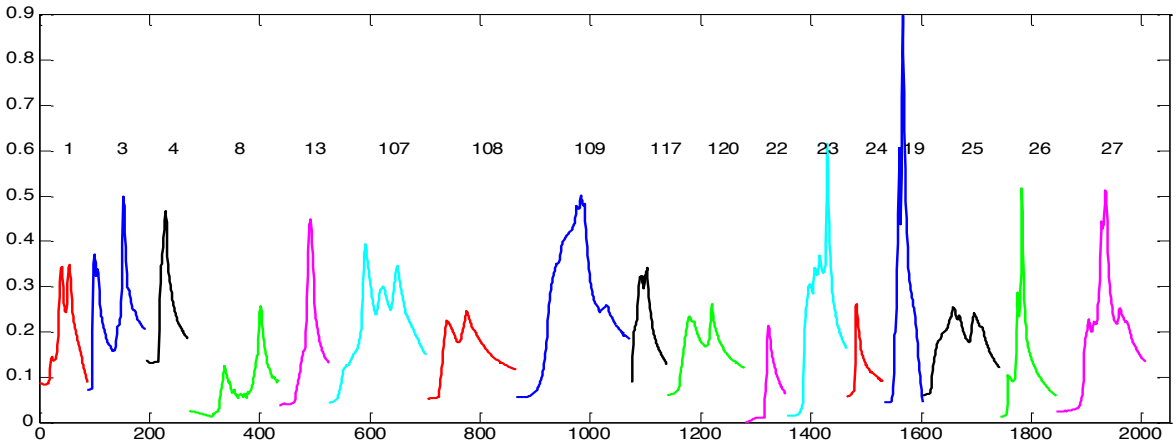

Figure 2
Les hauteurs sont représentées ici après normalisation selon la méthode décrite dans la section 3.3 ; les hauteurs réelles sont indiquées dans le Tableau 1.

3.3 NORMALISATION DES DONNÉES

3.3.1 Normalisation de la hauteur d'eau à Anduze (grandeur à prédire)

La normalisation de la hauteur d'eau y a été effectuée par la méthode adoptée par le partenaire 1 lors d'études précédentes : l'application de la transformation affine

$$y \leftarrow 0{,}9\frac{y-\min(y)}{\max(y)}.$$

3.3.2 Normalisation des données des pluviomètres (variables du modèle)

Soit x une hauteur de précipitation observé ; sa normalisation est effectuée par la transformation linéaire

$$x \leftarrow 0{,}9\frac{x}{\max(x)}$$

où $\max(x)$ est la hauteur de précipitation maximale présente dans la base de données.

Tableau 1 : Liste des événements collectés entre 1993 et 2008

Numéro	Date	Durée (heures)	Niveau d'eau maximum (m)
1	21-24 septembre 1994	35	3,71
3	4-5 octobre 1995	54	5,34
4	13-14 octobre 1995	92	5
8	10-12 novembre 1996	82	2,71
107	5-7 novembre 1997	74	4,2
108	26-27 novembre 1997	66	2,58
109	18-19 décembre 1997	104	5,37
117	20-21 octobre 1999	34	3,64
13	28-29 septembre 2000	46	4,8
120	12-14 novembre 2000	71	2,77
19	8-9 septembre 2002	29	9,71
22	24-25 septembre 2006	23	2,24
23	19-20 octobre 2006	55	6,61
24	17-18 novembre 2006	34	2,75
25	20-23 novembre 2007	70	2,69
26	21-23 octobre 2008	43	5,57
27	31 octobre – 3 novembre 2008	81	5,53

4. MODÈLES DYNAMIQUES MIS EN ŒUVRE

4.1 MODÈLE POSTULÉ

Rappelons que l'objectif de notre étude est la prédiction de la hauteur d'eau à Anduze en fonction des précipitations et des hauteurs d'eau passées, *en l'absence de prévisions de pluies*, à l'aide de méthodes d'apprentissage statistique. Nous avons postulé un modèle de type NARX (Nonlinear AutoRegressive with eXogenous inputs, voir par exemple [Ljung, 1999]) en temps discret avec « bruit d'état » : nous supposons que le processus de crue subite peut être décrit de manière satisfaisante, dans le domaine de variations des variables du modèle, par une relation du type

$$y\left(k+h_p\right)=f_{h_p}\left(\varphi\left(k\right)\right)+d\left(k\right),$$

où

- h_p est l'horizon de prédiction
- $k \in N^+$ est un entier positif qui désigne le temps discret kT, où T est la période d'échantillonnage,

- $f_{h_p}(.)$ est une fonction non linéaire inconnue appelée fonction de régression, relative à l'horizon de prédiction h_p,
- $d(k)$ est une réalisation d'une variable aléatoire d'espérance mathématique nulle qui modélise l'ensemble des bruits et des perturbations,
- $y(k)$ est la hauteur d'eau observée à Anduze à l'instant kT,
- $\boldsymbol{\varphi}(k) \in \mathbb{R}^n$ est le vecteur des variables du modèle postulé, défini par

$$\boldsymbol{\varphi}(k) = \left[y(k), y(k-1), \cdots, y(k-n_a), \mathbf{u}(k)^T, \cdots, \mathbf{u}(k-n_b)^T \right]^T,$$

- $\mathbf{u}(k)$ est le vecteur des précipitations mesurées par les six pluviomètres à l'instant kT,
- $n_a \in N^+$ désigne l'ordre (inconnu) du modèle,
- $n_b \in N^+$ désigne la longueur (inconnue) de la fenêtre des précipitations passées pertinentes pour prédire les hauteurs futures en l'absence d'estimation des précipitations futures.

L'objectif est de concevoir, à partir des données disponibles, un modèle prédictif de y(k), c'est-à-dire une fonction paramétrée $g_{h_p}\left[\boldsymbol{\varphi}(k), \mathbf{w}\right]$ *qui soit aussi proche que possible de la fonction de régression inconnue, dans le domaine de variation des variables* $\boldsymbol{\varphi}(k)$. On peut démontrer que si le modèle est parfait, c'est-à-dire si $g_{h_p}\left[\boldsymbol{\varphi}(k), \mathbf{w}\right] = f_{h_p}\left(\boldsymbol{\varphi}(k)\right)\ \forall k$, la variance de l'erreur de prédiction est égale à la variance du bruit $d(k)$, ce qui est caractéristique du prédicteur optimal, ou prédicteur à variance minimale.

4.2 MODÈLES PRÉDICTIFS

Nous avons mis en œuvre trois méthodes d'apprentissage de modèles prédictifs : la régression par les machines à vecteurs supports (SVR pour Support Vector Regression, voir par exemple [Smola, 2004]), la régression par les LSSVM (Least Squares Support Vector Machines, [Suykens 1999]) et la régression par les modèles dynamiques affines par morceaux (PWR pour PieceWise affine Regression, [Boukharouba, 2009]).

4.2.1 Régression SVR

L'objectif de la régression par les machines à vecteurs supports (SVR) est de trouver un modèle de la forme

$$g_{h_p}\left[\boldsymbol{\varphi}(k), \mathbf{w}, b\right] = \mathbf{w}^T \boldsymbol{\eta}\left[\boldsymbol{\varphi}(k)\right] + b,$$

où $\boldsymbol{\eta}\left[\boldsymbol{\varphi}(k)\right]$ est le vecteur obtenu par application de la transformation vectorielle $\boldsymbol{\eta}$ au vecteur $\boldsymbol{\varphi}(k)$, et où $\mathbf{w}$ et b sont les paramètres du modèle. Celui-ci est donc linéaire en ses paramètres.

w et *b* sont estimés à partir d'un ensemble de données expérimentales qui constituent l'ensemble d'apprentissage ; celui-ci est constitué de *N* couples $\left\{y\left(i+h_p\right),\boldsymbol{\varphi}\left(i\right),i=1\ldots N\right\}$.

Lorsque l'on cherche à estimer les paramètres de modèles à partir d'une quantité finie de données, on est confronté au problème du *risque structurel* [Vapnik, 1995] : si le vecteur $\boldsymbol{\eta}$ (donc le vecteur **w**) est de dimension trop grande, le modèle est trop « souple » et s'adapte finement aux données d'apprentissage au détriment de sa capacité de généralisation, c'est-à-dire de sa capacité à fournir des prédictions satisfaisantes dans des situations qui ne sont pas présentes dans l'ensemble d'apprentissage (« surajustement »). Inversement, si la dimension de $\boldsymbol{\eta}$ est trop petite, le modèle est incapable de s'adapter aux données. Pour minimiser le risque de surajustement, on peut exprimer le problème de l'estimation des paramètres du modèle comme un problème d'optimisation sous contraintes :

$$\text{Minimiser} \quad \frac{1}{2}\left\|w\right\|^2$$

$$\text{sous les contraintes} \quad \begin{cases} y(i+h_p)-\mathbf{w}^T\boldsymbol{\eta}\left[\boldsymbol{\varphi}\left(i\right)\right]-b\le\varepsilon \\ \mathbf{w}^T\boldsymbol{\eta}\left[\boldsymbol{\varphi}\left(i\right)\right]+b-y(i+h_p)\le\varepsilon \end{cases} \quad \forall i\in\left[1,N\right]$$

où $\varepsilon > 0$ est la tolérance que l'on admet sur l'erreur de modélisation.

Il va de soi que, pour une transformation non linéaire $\boldsymbol{\eta}$ et une précision ε données, ce problème n'admet pas nécessairement une solution. Si c'est le cas, on peut chercher une autre transformation linéaire, ou augmenter la tolérance. On peut aussi contourner ce problème en relâchant les contraintes afin d'autoriser quelques erreurs supérieures à la tolérance, en nombre aussi petit que possible. On introduit alors des « variables ressorts » qui constituent de nouveaux paramètres à estimer par apprentissage, et l'on reformule le problème de la façon suivante :

$$\text{Minimiser} \quad \frac{1}{2}\left\|\mathbf{w}\right\|^2+C\sum_{i=1}^{N}(\xi_i+\xi_i^*)$$

$$\text{sous les contraintes} \quad \begin{cases} y(i+h_p)-\mathbf{w}^T\boldsymbol{\eta}(\boldsymbol{\varphi}(i))-b\le\varepsilon+\xi_i \\ \mathbf{w}^T\boldsymbol{\eta}(\boldsymbol{\varphi}(i))+b-y(i+h_p)\le\varepsilon+\xi_i^* \quad \forall i\in\left[1,N\right] \\ \xi_i,\xi_i^*\ge 0 \end{cases}$$

On minimise ainsi conjointement

- la norme des paramètres afin de régulariser le modèle,
- la somme des variables ressorts afin de minimiser le nombre d'exemples pour lesquels la tolérance ε aux erreurs de modélisation est dépassée ; seules les erreurs de modélisation supérieures en valeur absolue à la tolérance sont pénalisées.

Le paramètre de régularisation *C* exprime l'importance que l'on accorde à la régularisation par rapport à la tolérance aux erreurs de modélisation : si l'on cherche à obtenir un modèle pour lequel la tolérance ε est respectée, au détriment du risque de surajustement, on choisit une valeur élevée pour *C* ; au contraire, si l'on est prêt à sacrifier la tolérance aux erreurs de modélisation pour minimiser le risque de surajustement, on choisit une petite valeur de *C*.

En pratique, on préfère résoudre numériquement la « formulation duale » du problème précédent. Le résultat essentiel du passage à la forme duale est le suivant : le modèle peut s'écrire sous la forme d'une combinaison linéaire de produits scalaires

$$g_{h_p}\left[\boldsymbol{\varphi}(k),\mathbf{w},b\right] = \sum_{i=1}^{N}\alpha_i\left(\boldsymbol{\eta}^T\left[\boldsymbol{\varphi}(i)\right]\boldsymbol{\eta}\left[\boldsymbol{\varphi}(k)\right]\right)+b$$
$$= \sum_{i=1}^{N}\alpha_i K\left(\boldsymbol{\varphi}(i),\boldsymbol{\varphi}(k)\right)+b$$

où la somme porte sur tous les exemples de l'ensemble d'apprentissage, et où $K(.,.)$ est appelée fonction noyau. L'objectif de l'optimisation est alors l'estimation des paramètres α_i et b.

Une fonction noyau doit se comporter comme un produit scalaire, ce qui introduit des contraintes sur son choix. En pratique, des fonctions noyaux d'usage général ont été établies, et, pour certaines catégories de problèmes, des fonctions noyaux spécifiques ont été développées. Dans toute la suite, nous utiliserons les noyaux gaussiens d'usage général

$$K\left(\boldsymbol{\varphi}(i),\boldsymbol{\varphi}(k)\right) = \frac{\left\|\boldsymbol{\varphi}(i)-\boldsymbol{\varphi}(k)\right\|^2}{\sigma^2}$$

où σ est un hyperparamètre dont la valeur doit être choisie par le concepteur, ou déterminée par une procédure de sélection de modèle.

Le modèle recherché est donc une combinaison linéaire de gaussiennes multidimensionnelles, centrées sur les points de l'ensemble d'apprentissage, et de largeur σ.

4.2.2 Régression par les LSSVM

Comme la méthode SVR décrite dans le paragraphe précédent, la méthode LSSVM (Least squares support vector machines) cherche un modèle paramétré de la forme

$$g_{h_p}\left[\boldsymbol{\varphi}(k),\mathbf{w},b\right] = \mathbf{w}^T\boldsymbol{\eta}\left[\boldsymbol{\varphi}(k)\right]+b$$

dont on cherche à estimer les paramètres $\mathbf{w}$ et b à partir d'un ensemble d'apprentissage constitué d'exemples pour lesquels la grandeur d'intérêt observée $y(k)$ et le vecteur de variables $\boldsymbol{\varphi}(k)$ sont connus.

L'estimation des paramètres du modèle est effectuée en résolvant le problème de minimisation quadratique sous contraintes suivant :

$$\text{Minimiser} \quad \frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{\gamma}{2}\sum_{i=1}^{N}e(i)^2$$
$$\text{sous les contraintes} \quad y(i+h_p)=\mathbf{w}^T\boldsymbol{\eta}(\boldsymbol{\varphi}(i))+b+e(i) \quad \forall i\in\left[1,N\right]$$

Comme dans le cas des SVR avec variables élastiques, on cherche à minimiser une combinaison linéaire de la norme des paramètres et de l'erreur de modélisation. Contrairement au cas des SVR, on ne définit pas de tolérance pour l'erreur de modélisation : toute erreur de modélisation est pénalisée, quelle que soit sa valeur.

De plus, les contraintes sont ici des contraintes d'égalité, alors que celles des SVR sont des contraintes d'inégalité.

Le passage à la forme duale met le modèle sous une forme analogue à celle des SVR

$$g_{h_p}\left[\boldsymbol{\varphi}(k),\mathbf{w},b\right]=\sum_{i=1}^{N}\alpha_i\left(\boldsymbol{\eta}^T\left[\boldsymbol{\varphi}(i)\right]\boldsymbol{\eta}\left[\boldsymbol{\varphi}(k)\right]\right)+b$$
$$=\sum_{i=1}^{N}\alpha_i K\left(\boldsymbol{\varphi}(i),\boldsymbol{\varphi}(k)\right)+b$$

où les α_i et b sont les solutions du système d'équations linéaires suivant :

$$\begin{bmatrix} 0 & \mathbf{1}_N^T \\ \mathbf{1}_N & \Omega+\gamma^{-1}\mathbf{I}_N \end{bmatrix}\begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix}=\begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}$$

où $\mathbf{y}=\left[y(1),\ldots y(N)\right]^T$, $\boldsymbol{\alpha}=\left[\alpha_1,\ldots\alpha_N\right]^T$, $\mathbf{1}_N$ est le vecteur dont les N composantes sont égales à 1, $\mathbf{I}_N$ est la matrice identité de dimension N, et $\boldsymbol{\Omega}$ est la matrice carrée de dimension N dont les éléments sont

$$\left\{\Omega_{ij}=\boldsymbol{\eta}^T\left[\boldsymbol{\varphi}(i)\right]\boldsymbol{\eta}\left[\boldsymbol{\varphi}(j)\right]=K\left(\boldsymbol{\varphi}(i),\boldsymbol{\varphi}(j)\right),\ i=1\ldots N,\ j=1\ldots N\right\}.$$

4.2.3 Régression PWR

Dans le cas de la régression PWR (PieceWise linear Regression), le modèle prédictif recherché est affine par morceaux. L'espace des variables est partitionné en s domaines R_i, et le modèle est de la forme :

$$g_{h_p}\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]=\begin{cases} \mathbf{w}_1^T\boldsymbol{\varphi}(k) & \text{si } \boldsymbol{\varphi}(k)\in R_1 \\ \vdots & \\ \mathbf{w}_s^T\boldsymbol{\varphi}(k) & \text{si } \boldsymbol{\varphi}(k)\in R_s, \end{cases}$$

où $\mathbf{w}_i$ est le vecteur des paramètres relatifs au sous-modèle affine.

Les modèles cherchés étant affines, les vecteurs φ contiennent une composante constante égale à 1.

Pour estimer les paramètres des modèles affines par morceaux, il faut effectuer une classification non supervisée de données qui permet de déterminer la partition de l'espace des variables, préalablement à l'estimation des vecteurs de paramètres des sous-modèles affines de chacune des régions définies par la partition. Le nombre de régions définies par la partition n'étant pas connu a priori, on met en œuvre une procédure itérative d'affectation des données aux régions et d'estimation des paramètres des sous-modèles de chaque région.

Initialisation

Le nombre de sous-modèles étant inconnu, on partitionne initialement l'espace des variables en autant de régions qu'il y a d'exemples dans l'ensemble d'apprentissage. Le modèle associé à chaque exemple est obtenu en appliquant la méthode des moindres carrés à cet exemple et à ses k plus proches voisins. On obtient ainsi un ensemble de N vecteurs de paramètres initiaux $\mathbf{w}^0_i$, i = 1…N.

Réaffectation

Les données ayant été initialement affectées à *N* classes, cette étape de la procédure a pour objectif de réduire le nombre de classes jusqu'à trouver un nombre minimal de classes disjointes. Chaque donnée *i* à classer est décrite par le vecteur $\mathbf{x}_i$ obtenu par concaténation des variables de l'exemple *i* et de la grandeur mesurée correspondante :

$$\mathbf{x}_i = \begin{bmatrix} \boldsymbol{\varphi}(i) & y(i+h_p) \end{bmatrix}^T .$$

Deux données $\mathbf{x}_i$ et $\mathbf{x}_j$ suffisamment proches (au sens de la distance euclidienne) sont susceptibles d'appartenir à la même classe, donc d'être modélisées par le même sous-modèle affine. D'autre part, il faut que les sous-modèles soient suffisamment nombreux pour que le modèle puisse rendre compte des données. Il faut donc définir un indice de ressemblance entre les données qui tienne compte de ces deux considérations ; une fois cet indice calculé, il faut l'utiliser pour réaffecter les données aux classes de façon à diminuer le nombre de classes tout en modélisant correctement les données.

Considérons une itération de l'algorithme de classification, avant laquelle *c* classes ont été définies, et les paramètres $\mathbf{w}_n$ ($1 \le n \le c$) des *c* sous-modèles correspondants ont été estimés par la méthode des moindres carrés.

L'indice de ressemblance entre une donnée *i* et une donnée *j* qui

- appartient à l'ensemble des *k* plus proches voisins de *i*,
- appartient à la classe C_n ($1 \le n \le c$)

est défini par la relation

$$\phi_j^i = \exp\left(-\frac{\left\|\mathbf{x}(i)-\mathbf{x}(j)\right\|^2}{\frac{1}{2\left|C_n\right|^2}\sum_{l\in C_n}\sum_{m\in C_n}\left\|\mathbf{x}(l)-\mathbf{x}(m)\right\|^2} \right) \exp\left(-\frac{\left(y(i+h_p)-\mathbf{w}_n^T\boldsymbol{\varphi}(i)\right)^2}{\frac{1}{\left|C_n\right|}\sum_{m\in C_n}\left(y(m)-\mathbf{w}_n^T\boldsymbol{\varphi}(m)\right)^2} \right)$$

où $\left|C_n\right|$ désigne le nombre d'exemples présents dans la classe *n* à l'itération considérée.

Les deux exemples *i et j* sont donc d'autant plus semblables que

- leur distance euclidienne est petite devant la distance moyenne entre exemples appartenant à la classe de l'exemple *j*,
- l'erreur quadratique de modélisation commise si l'on prédit $y(i+h_p)$ à l'aide du modèle de la classe à laquelle appartient *j* est petite devant l'erreur quadratique moyenne commise par ce modèle sur les exemples de cette classe.

On peut ainsi calculer, pour chaque exemple *i*, les indices de ressemblance avec ses *k* plus proches voisins. La décision d'affectation de cet exemple à l'une des *c* classes existantes est fondée sur l'indice d'appartenance A_i^q de l'exemple *i* à la classe *q* :

$$A_i^q = \frac{\sum_m \phi_m^i}{\sum_l \phi_l^i},\ 1 \le q \le c$$

où la somme au numérateur porte sur les *k* plus proches voisins de l'exemple *i* qui appartiennent à la classe *q*, et la somme au dénominateur porte sur tous les plus proches voisins de l'exemple *i*. Cet indicateur est compris entre 0 (si aucun des plus proches voisins de *i* n'appartient à la classe *q*) et 1 (si tous les plus proches voisins de *i* appartiennent à la classe *q*).

Le critère d'affectation de l'exemple *i* est alors : *l'exemple i est affecté à la classe r pour laquelle l'indice d'appartenance de i est maximum.* Le nombre d'exemples de la classe *r* est augmenté de 1, celui de la classe de l'exemple *i* est diminué de 1. Si la classe de l'exemple *i* ne contenait qu'un exemple, elle disparaît.

Une fois que l'affectation de tous les exemples a été examinée, les paramètres des modèles affines sont mis à jour par la méthode des moindres carrés.

Critère d'arrêt

La procédure est arrêtée lorsque les paramètres ne varient plus significativement entre deux itérations successives ; dans le présent travail, nous avons choisi d'arrêter la procédure si la variation du module du vecteur **w** devient inférieure à 10^{-5}.

5. MODÉLISATION DU BASSIN VERSANT DU GARDON D'ANDUZE

Cette section décrit l'application des méthodes décrites dans la section 4 à la modélisation du bassin versant du Gardon d'Anduze.

La première partie est consacrée à l'application des méthodes de régression, présentées dans la section précédente, à la prédiction des crues à l'aide d'un modèle unique, dit « modèle global » construit à partir de l'ensemble des données disponibles et prévues à cet effet.

La deuxième partie décrit une approche différente, qui permet de déterminer des classes d'événements et de construire un modèle spécifique à chaque classe.

Pour fixer la terminologie,

- nous désignons par *apprentissage* l'ensemble des algorithmes et procédures d'estimation des paramètres des modèles prédictifs,
- nous désignons par *validation* l'ensemble des algorithmes et procédures d'estimation de l'erreur de généralisation des modèles prédictifs après apprentissage, aux fins de sélection du meilleur modèle,
- nous désignons par *test* l'ensemble des algorithmes et procédures d'estimation de l'erreur de généralisation du modèle final.

5.1 CONCEPTION DE MODÈLES GLOBAUX

5.1.1 Objectif

L'objectif est de concevoir, par apprentissage statistique à partir de l'ensemble des événements d'apprentissage (voir section suivante), des modèles prédictifs des niveaux d'eau à Anduze, en l'absence de prédiction de pluies, à des horizons de prédiction 30 mn, 1 h, 2 h, …5 h (h_p = 1, 2, 4, …, 10).

Rappelons que les modèles prédictifs cherchés sont de la forme :

$$g_{h_p}\left(\boldsymbol{\varphi}(k),\mathbf{w}\right) = g_{h_p}\left(y(k),y(k-1),\cdots,y(k-n_a),\mathbf{u}(k)^T,\cdots,\mathbf{u}(k-n_b)^T,\mathbf{w}\right),$$

où $y(k)$ est la hauteur d'eau mesurée à Anduze à l'instant kT (T = 30 mn), $\mathbf{u}(k)$ est le vecteur dont les composantes sont les hauteurs de pluie mesurées à l'instant k par les pluviomètres, et $\mathbf{w}$ est le vecteur des paramètres du modèle, estimés par apprentissage. Les valeurs de n_a et n_b sont déterminées par validation croisée partielle comme indiqué dans le paragraphe 5.1.3.

5.1.2 Données

Dans un souci de cohérence avec les travaux du partenaire 1, nous utilisons les mêmes ensembles d'apprentissage, de validation et de test. Sauf indication contraire, les événements {1, 3, 4, 8, 107, 108, 109, 117, 120, 22, 23, 24, 25} sont utilisés pour l'apprentissage et la validation. Les événements {13, 19, 26, 27} sont utilisés pour le test.

5.1.3 Sélection de modèles et d'hyperparamètres par validation croisée partielle

La méthode de validation croisée partielle a été utilisée pour sélectionner simultanément

- la constante de régularisation de l'apprentissage C et, pour la régression SVR, la quantité ε (section 4.2.1),
- l'ordre du modèle n_a et l'horizon d'observation des pluies n_b, définis dans la section 4.1,
- l'hyperparamètre σ du noyau gaussien (section 4.2.1).

Une fois ces cinq quantités fixées, le modèle obtenu est unique. La validation croisée partielle permet alors d'estimer les performances de ce modèle.

Mécanisme de validation croisée partielle

La validation croisée partielle utilise l'ensemble des 13 événements E = {1, 3, 4, 8, 107, 108, 109, 117, 120, 22, 23, 24, 25}. Pour chaque horizon de prédiction, l'apprentissage d'un modèle est effectué sur 12 événements, et les performances de ce modèle sont estimées en l'appliquant au 13ème événement. Cette opération est effectuée quatre fois avec les ensembles d'événements d'apprentissage E_{Ai} (i = 1…4), et les performances des quatre modèles obtenus sont estimées sur les événements de validation E_{Vi} (i = 1…4) :

- E_{A1} = {1, 4, 8, 107, 108, 109, 117, 120, 22, 23, 24, 25} E_{V1} = {3}
- E_{A2} = {1, 3, 8, 107, 108, 109, 117, 120, 22, 23, 24, 25} E_{V2} = {4}
- E_{A3} = {1, 3, 4, 8, 107, 108, 117, 120, 22, 23, 24, 25} E_{V3} = {109}
- E_{A4} = {1, 3, 4, 8, 107, 108, 109, 117, 120, 22, 24, 25} E_{V4} = {23}

Une fois que les valeurs optimales de C, n_a, n_b, σ, et, le cas échéant, ε, sont déterminées pour chaque horizon de prévision, on réalise l'apprentissage des modèles sur les 13 événements de l'ensemble E. Les performances de ces modèles sont estimées à l'aide des événements de l'ensemble de test (section 5.1.2).

Indicateurs de qualité

Pour estimer les performances des modèles, nous avons utilisé l'erreur quadratique moyenne

$$EQM = \frac{1}{N}\sum_{k=1}^{N}\left(y(k+h_p) - g_{h_p}\left(\boldsymbol{\varphi}(k), \mathbf{w}\right)\right)^2$$

ainsi que deux indicateurs utilisés habituellement en hydrogéologie (voir par exemple [Krause, 2005]) :

- Le coefficient de Nash

$$C_N = 1 - \frac{\sum_{k=1}^{N}\left(y(k+h_p) - g_{h_p}\left(\boldsymbol{\varphi}(k), \mathbf{w}\right)\right)^2}{\sum_{k=1}^{N}\left(y(k) - \bar{y}\right)^2}$$

  où $\bar{y}$ désigne la hauteur d'eau mesurée moyenne sur la séquence observée. La prédiction est d'autant meilleure que le coefficient de Nash est plus proche de 1. Si le modèle prédit simplement que la hauteur d'eau future sera égale à la moyenne des hauteurs d'eau mesurées jusqu'à l'instant présent, le coefficient de Nash vaut 0. Une valeur négative de C_N indique que le modèle est de très mauvaise qualité.

- Le coefficient de persistance

$$C_p = 1 - \frac{\sum_{k=1}^{N}\left(y(k+h_p) - g_{h_p}\left(\boldsymbol{\varphi}(k), \mathbf{w}\right)\right)^2}{\sum_{k=1}^{N}\left(y(k+h_p) - y(k)\right)^2}$$

  Ce coefficient vaut 0 si le prédicteur est « naïf », c'est-à-dire s'il se contente de prédire que la hauteur d'eau future sera égale à la hauteur d'eau actuelle ; il vaut 1 si la prédiction est toujours exacte. Une valeur négative de C_p indique que le modèle est moins performant que le modèle naïf.

Filtrage spatial des pluies : réduction de la dimension du vecteur des variables par application des polygones de Thiessen :

La tâche T4 comportait une sous-tâche qui consistait à introduire des connaissances physiques dans les modèles conçus par apprentissage. La contribution la plus notable de cette sous-tâche à l'efficacité de nos prédicteurs est décrite ici.

Rappelons que les variables exogènes de nos modèles sont les précipitations mesurées par les 6 pluviomètres, échantillonnées avec une période d'échantillonnage de 30 mn, sur une fenêtre d'observation pouvant aller jusqu'à cinq heures : en effet, le temps d'écoulement de l'eau entre le pluviomètre le plus éloigné de l'exutoire et le point de mesure des hauteurs d'eau est estimé à 5 heures. Pour faire une prévision une heure à l'avance par exemple, il est donc inutile de prendre comme variable les précipitations tombées plus de cinq heures plus tôt. Néanmoins, il reste donc une soixantaine de variables de précipitations (6 pluviomètres x 5 heures).

Afin de réduire cette dimension, une « précipitation moyenne pondérée », calculée par la méthode des polygones de Thiessen [Thiessen, 1911], a été mise en œuvre afin d'agréger les données des six pluviomètres en une seule donnée représentative des précipitations. Les polygones de Thiessen sont déterminés en traçant toutes les médiatrices des segments qui joignent les pluviomètres deux à deux ; on affecte à chaque pluviomètre un poids correspondant à la fraction de l'aire totale du bassin versant qui est recouverte par le plus petit polygone à l'intérieur duquel il se trouve.

La précipitation moyenne pondérée est alors

$$u(k) = \frac{1}{6}\sum_{i=1}^{6} p_i u_i(k)$$

où $u_i(k)$ est la précipitation mesurée à l'instant k par le pluviomètre i et p_i est le poids affecté à ce pluviomètre. La superficie du polygone de Thiessen associé à chaque pluviomètre du bassin versant d'Anduze, et le poids correspondant, sont indiqués dans le Tableau 2.

Tableau 2

Pluviomètre	**Aire du polygone (km²)**	**Poids**
Anduze	48	9,16
Barre des Cévennes	90	17,18
Mialet	96	18,32
Saint Roman	142	27,10
Saumane	62	11,83
Soudorgues	86	16,41
Surface totale	**524**	**100**

La Figure 3 montre, pour chacun des trois types de modèles prédictifs (SVR, LSSVM, PWR), l'évolution du coefficient de Nash et du coefficient de persistance en fonction de l'horizon de prédiction pour les modèles qui utilisent les précipitations réelles et ceux qui utilisent les précipitations moyennes pondérées (ou « pluies agrégées »). L'utilisation des pluies agrégées permet d'améliorer notablement les performances en termes de coefficient de Nash et de coefficient de persistance (sauf pour les prédictions SVR à 4 et 5 h). Les courbes mettent également en évidence la tendance, commune à tous les modèles, à une dégradation du coefficient de Nash avec l'horizon de prédiction, ce qui est normal puisque les modèles ne disposent pas de prédiction de pluies.

Compte tenu de cette observation, les modèles présentés dans la suite de ce rapport ont tous pour variables exogènes les pluies agrégées.

La Figure 4 présente les résultats obtenus, sur les événements choisis pour les tests, par les méthodes de régression présentées dans la section 4.2. Outre les hauteurs d'eau réelles et prédites, les précipitations agrégées ont été reportées.

Rappelons que l'événement 19 est très particulier : c'est l'événement le plus important présent dans la base de données ; les mesures ont été en grande partie reconstruites a posteriori, les appareils de mesure ayant été mis hors d'usage pendant la crue.

Outre les méthodes de régression non linéaire, nous avons porté sur les mêmes figures les prédictions d'un modèle linéaire obtenu par la méthode des moindres carrés (LS pour least squares).

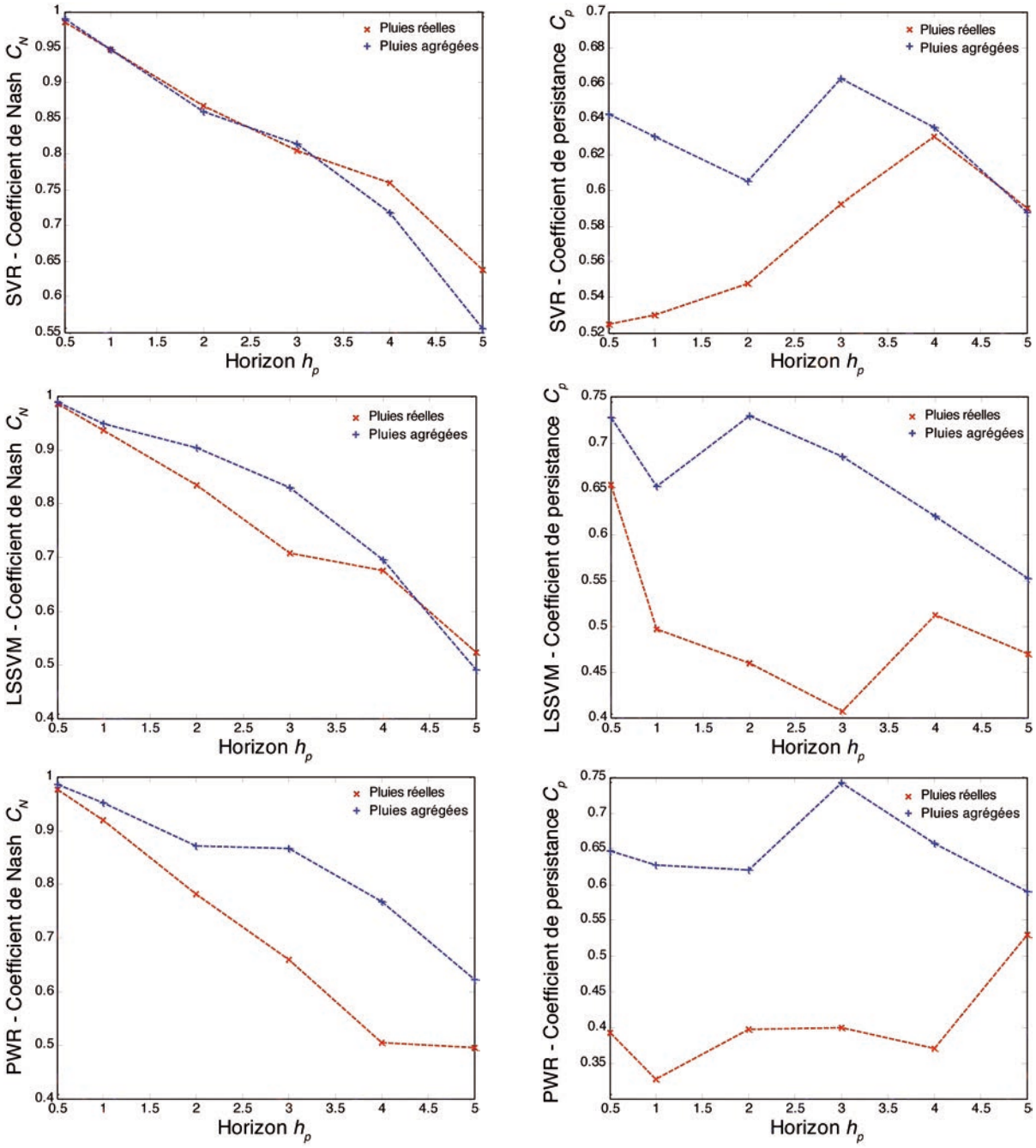
SVR - Coefficient de Nash C_N
Pluies réelles
Pluies agrégées
Horizon h_p
SVR - Coefficient de persistance C_p
Pluies réelles
Pluies agrégées
Horizon h_p
LSSVM - Coefficient de Nash C_N
Pluies réelles
Pluies agrégées
Horizon h_p
LSSVM - Coefficient de persistance C_p
Pluies réelles
Pluies agrégées
Horizon h_p
PWR - Coefficient de Nash C_N
Pluies réelles
Pluies agrégées
Horizon h_p
PWR - Coefficient de persistance C_p
Pluies réelles
Pluies agrégées
Horizon h_p

Figure 3

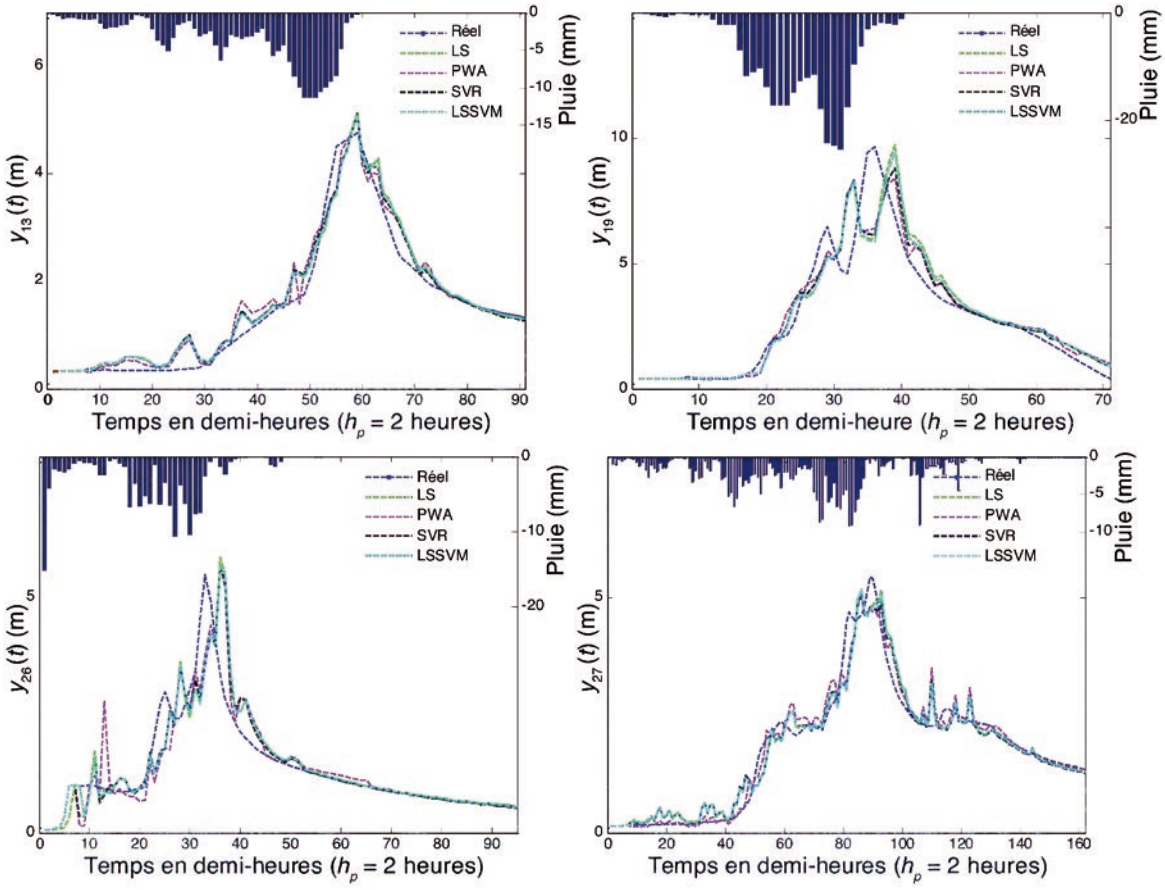

Figure 4

On observe que les pics sont correctement prédits en temps et en amplitude pour les événements 13 et 27. L'amplitude de l'événement 26 est correctement prédite, mais avec un retard. Comme on pouvait le prévoir, l'événement 19 est le plus mal prédit : (i) le pic à 15 h est surestimé et prédit avec retard, (ii) le pic principal est prédit avec retard, et sous-estimé sauf par le modèle linéaire.

À titre illustratif, la

Figure 5 montre les prédictions des modèles sur les mêmes événements pour un horizon de prédiction de 4 heures. La perte de précision lorsque l'horizon de prédiction augmente est clairement mise en évidence, surtout pour l'événement 19.

Afin de présenter de manière synthétique une estimation des performances des différents modèles en fonction de l'horizon de prédiction, nous avons reporté sur la Figure 6 les erreurs quadratiques moyennes de modélisation sur les événements de test.

Pour tous les événements de test, l'erreur de modélisation se dégrade lorsque l'horizon de prédiction augmente ; elle est évidemment d'autant plus grande que cet horizon approche du temps de transit de l'eau, estimé à partir de données hydrogéologiques, entre les pluviomètres et l'exutoire. Les prédictions à court terme sont excellentes pour tous les modèles (erreur moyenne de 10 à 60 cm). À l'horizon de 5 h, l'erreur moyenne est de l'ordre 40 à 70 cm pour les événements 13 et 27, de 80 à 90 cm pour l'événement 26, et de 1,50 à 2 m pour l'événement exceptionnel 19.

Tous les modèles sont à peu près équivalents en termes d'erreurs de prédiction moyenne pour les horizons courts (1/2 heure et 1 heure) ; des différences apparaissent pour les horizons de prédiction plus longs, mais elles ne sont pas très

importantes. Les modèles LSSVM et PWA sont les moins performants en moyenne sur les quatre événements de test.

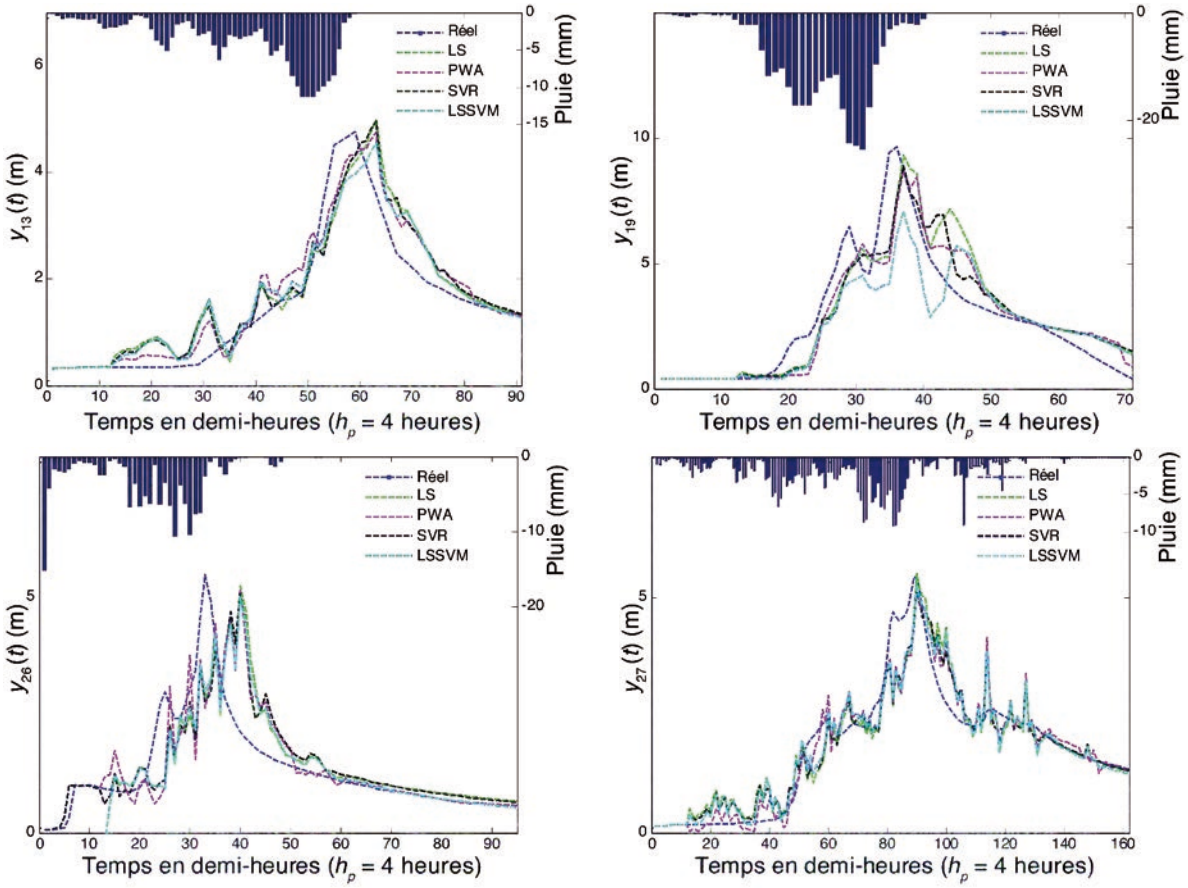

Figure 5

Les performances exprimées en termes de coefficient de persistance ne présentent pas de variation significative en fonction de l'horizon ou en fonction de la méthode de régression utilisée. Les coefficients de persistance varient de 10% à 20% pour un événement de test donné ; tous événements de test et tous modèles confondus, le coefficient de persistance le plus élevé vaut 0,95, et le plus faible vaut 0,55. Il est toujours positif, ce qui signifie que, pour tous les modèles et tous les horizons, les prédicteurs obtenus sont plus efficaces que le prédicteur naïf.

La constatation la plus significative est évidemment le fait que les modèles linéaires n'ont pas des performances sensiblement différentes, en termes d'erreur de prédiction moyenne, que les modèles non linéaires. Cela suggère que l'écart-type du noyau gaussien, déterminé par validation croisée pour obtenir des performances de généralisation satisfaisantes, est suffisamment grand pour que les modèles qui en résultent soient à peu près linéaires. On bénéficie donc peu des avantages des modèles de régression non linéaires.

La section suivante présente une démarche qui permet de surmonter cette difficulté.

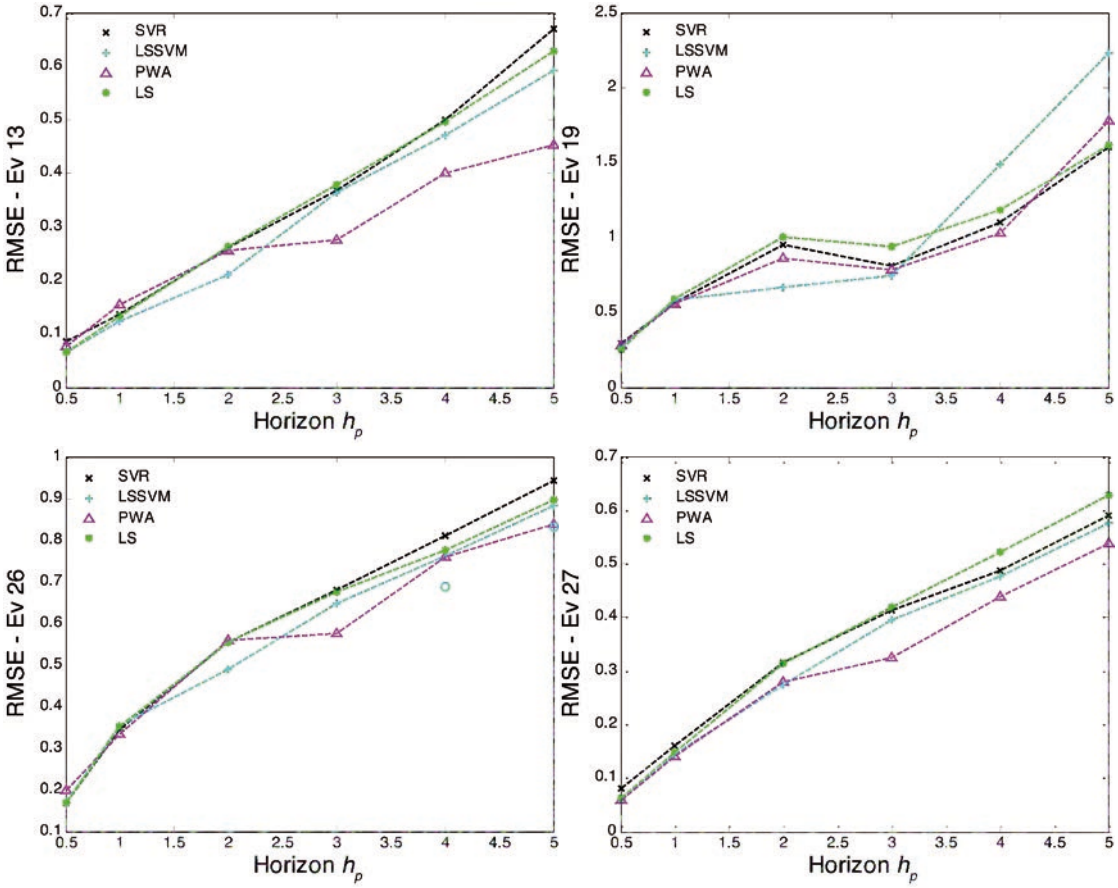

Figure 6

5.2 CLASSIFICATION DES ÉVÉNEMENTS ET CONCEPTION DE MODÈLES LOCAUX

5.2.1 Introduction

Nous avons mis en évidence, dans le chapitre précédent, les limites de la stratégie consistant à réaliser un modèle global par apprentissage statistique à partir de tous les exemples disponibles (hors exemples de test). Nous présentons ici une stratégie originale qui consiste à regrouper les événements en classes, puis à réaliser un modèle par classe. Chaque modèle est conçu à partir d'un nombre d'exemples plus restreint que le modèle global, mais la variabilité de ces exemples est moins grande, ce qui permet de tirer le meilleur parti de la non-linéarité des méthodes mises en œuvre.

5.2.2 Matrice de ressemblance entre événements

Pour grouper les différents événements en classes, il faut d'abord définir une « distance » entre événements.

La première étape consiste à construire des « modèles locaux » (par opposition aux « modèles globaux » décrits dans la section 5.1) : à cette étape, ce sont des modèles construits par apprentissage d'un seul événement. Désignons par M_i le modèle obtenu par apprentissage sur le seul événement i. Nous définissons la ressemblance entre deux événements à partir du comportement de leurs modèles locaux : deux

événements *i* et *j* sont considérés comme proches si le modèle M_i peut prédire correctement l'événement *j* et si le modèle M_j peut prédire correctement l'événement *i*. Nous désignons par $RMSE_{ij}$ la racine carrée de l'erreur quadratique moyenne commise par le modèle M_i lorsqu'il prédit l'événement *j*, et nous appelons « matrice de ressemblance » la matrice dont l'élément général est $RMSE_{ij}$. Cette matrice n'est pas symétrique (Figure 7). Nous pouvons néanmoins définir une quantité qui possède la propriété de symétrie :

$$d_{ij} = \max_{i,j}\left(RMSE_{ij}, RMSE_{ji}\right).$$

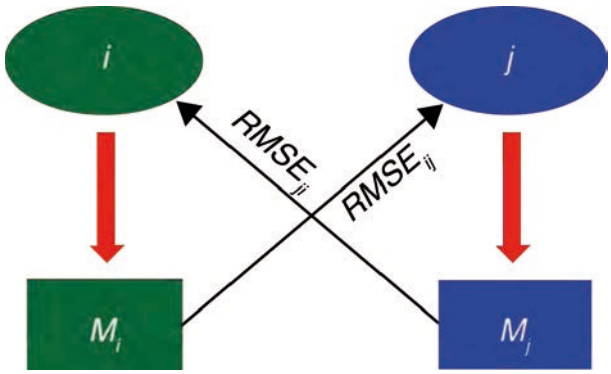

Figure 7

La Figure 8 présente les matrices de ressemblance pour l'ensemble des événements, modélisés par des modèles SVR, pour deux horizons différents. Le Tableau 3 répertorie les correspondances entre les numéros des événements (Tableau 1) et les numéros des colonnes et lignes des matrices de ressemblance.

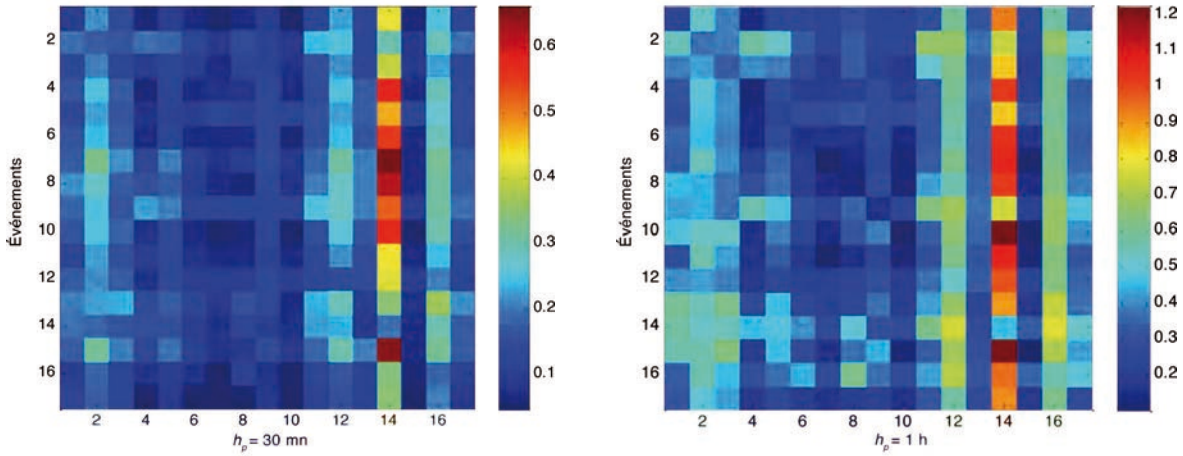

Figure 8
Matrices de ressemblance

Tableau 3

N° de colonne	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
N° d'événement	1	3	4	8	13	107	108	109	117	120	22	23	24	19	25	26	27

La ligne *i* de chaque matrice contient, en code de couleurs, les erreurs de modélisation commises par le modèle local construit par apprentissage de l'événement *i* lorsqu'on l'utilise pour prédire chaque événement. La colonne *j* de la matrice contient les erreurs de modélisation commises sur l'événement *j* lorsqu'il est prédit par chacun des modèles locaux.

On observe ainsi que l'événement exceptionnel 19 (colonne 14) est très mal prédit par tous les modèles locaux autres que son propre modèle local ; il en va de même, à un moindre degré, pour les événement 3, 23 et 26 (colonnes 2, 12 et 16 respectivement). On observe également (ligne 14) que le modèle construit par apprentissage sur l'événement 19 prédit assez bien les autres événements, à l'exception des événements 23 et 26 (colonnes 12 et 16).

Inversement les lignes comportant beaucoup de cases bleues correspondent à des modèles locaux qui généralisent bien, et les colonnes comportant beaucoup de cases bleues correspondent à des événements « facilement » modélisables.

5.2.3 Classification hiérarchique des événements

Une fois les matrices de ressemblance établies, une classification hiérarchique ascendante est mise en œuvre. À chaque itération, on agrège les deux événements les plus semblables, que nous définissons comme les événements *i* et *j* tels que

$$(i,j) = \arg\min_{m=1\ldots17,\, n=1\ldots17} (d_{mn}).$$

Les séquences de hauteurs d'eau et de précipitations des deux événements sont alors concaténées et ne forment plus qu'un événement. Un modèle local est construit par apprentissage sur cet événement, et la matrice de ressemblance (dont la dimension est diminuée de 1) est mise à jour. La procédure est résumée par le pseudo-code suivant :

Tant que le nombre n_e d'événements est supérieur à 1

Chercher le couple d'événements *i* et *j* tels que

$$(i,j) = \arg\min_{m=1\ldots n_e,\, n=1\ldots n_e} (d_{mn})$$

Concaténer les séquences de hauteurs et de pluies des deux événements

Construire un modèle par apprentissage sur cette séquence

Mettre à jour la matrice de ressemblance, décrémenter n_e.

Fin

La Figure 9 présente un dendrogramme obtenu pour un horizon de prédiction de 30mn. Il montre que le modèle de l'événement 19 (14 sur le dendrogramme) a un comportement différent de celui des autres événements. Comme indiqué plus haut, il s'agit d'un événement exceptionnellement violent, pour lequel les données disponibles ont été reconstruites *a posteriori*.

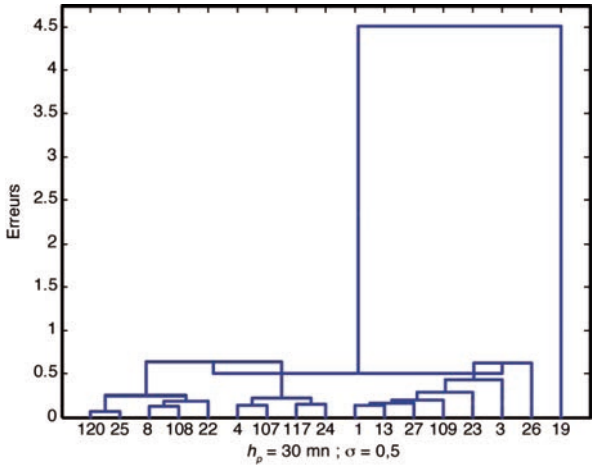

Figure 9

5.2.4 Conception des modèles locaux

Dans la section 5.1, nous avons montré que la validation croisée conduit, pour des modèles globaux, à une valeur σ de la largeur des gaussiennes qui est trop grande pour que l'on puisse bénéficier de la non-linéarité des modèles, en raison de la grande variabilité des événements disponibles, et de leur petit nombre. En revanche, pour la conception de modèles locaux, on peut s'attendre à ce que la valeur de σ soit plus petite. À titre illustratif, la Figure 10 montre la distribution des données si l'on considère un modèle du premier ordre avec une fenêtre d'observation de pluie d'une heure, pour l'événement 1. On voit qu'une largeur de gaussienne de l'ordre de 0,1 permettrait de recouvrir une fraction raisonnable du nombre de points, alors qu'une largeur de 0,01 donnerait des gaussiennes très locales, donc un fort risque de surajustement, et qu'une largeur de 0,4 donnerait des gaussiennes qui recouvriraient toutes les données, conduisant à un modèle quasiment linéaire. Ces ordres de grandeur sont à comparer à la valeur σ = 50 typiquement trouvée par validation croisée pour les modèles globaux.

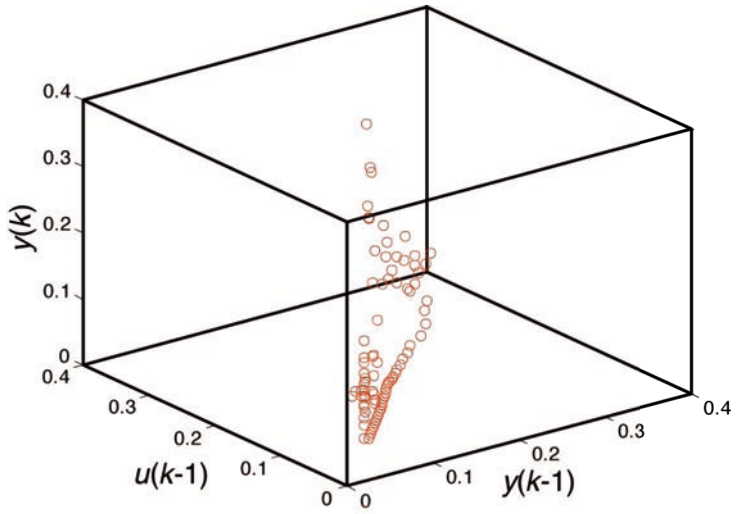

Figure 10

La valeur de σ a évidemment une influence sur la classification hiérarchique des modèles. La Figure 11 montre les dendrogrammes obtenus pour des valeurs de σ comprises entre 0,1 et 20, pour un horizon de prédiction de 30 mn.

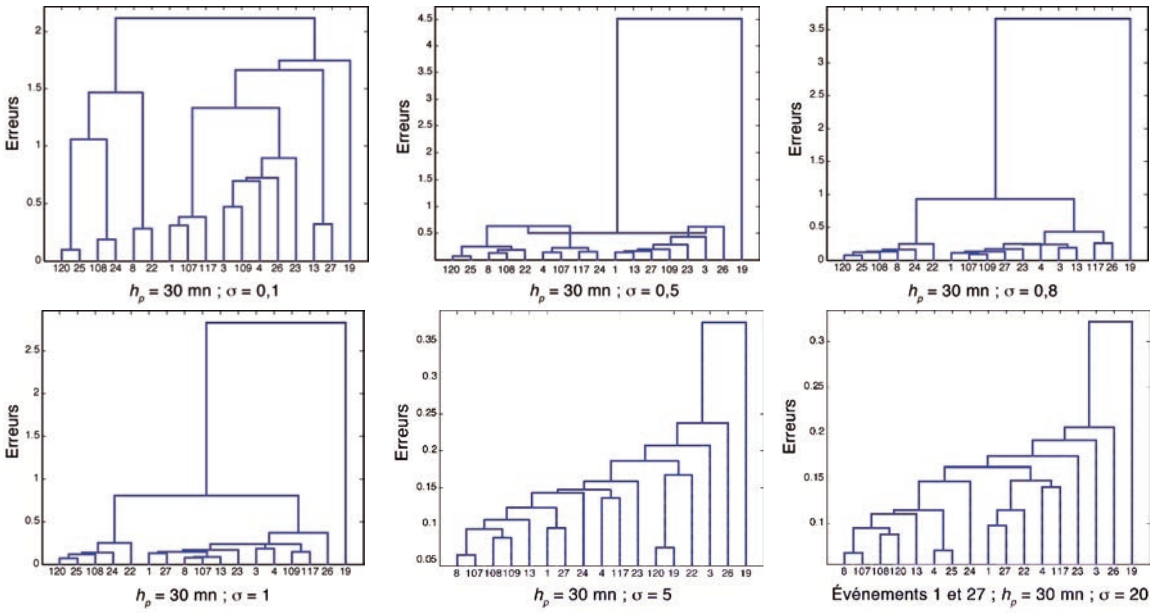

Figure 11

Trois types de dendrogammes apparaissent :

- pour σ = 0,1, les modèles obtenus à partir d'événements uniques généralisent très mal, donc se ressemblent peu au sens de la ressemblance définie à la section 5.2.2. : par exemple, la fusion des événements 120, 25, 108 et 24 produit un modèle qui commet une erreur de prédiction moyenne de 1 m, un ordre de grandeur au-dessus de l'erreur de prédiction commise par les modèles globaux pour cet horizon (Figure 6) ;
- pour σ = 0,5, σ = 0,8 et σ = 1, on voit apparaître des groupes bien individualisés ; le caractère exceptionnel de l'événement 19 est bien mis en évidence par le mécanisme de classification ;
- pour σ = 5 et σ = 20, chaque itération conduit à la fusion d'un ou deux événements au groupe existant à l'itération précédente, ce qui ne permet pas de distinguer des groupements de modèles.

5.2.5 Résultats (modèles SVR)

La Figure 12 présente les groupes obtenus pour chaque horizon de prédiction, avec σ = 0,5. Chaque ligne montre l'évolution d'un groupe en fonction de l'horizon de prédiction. Il n'est pas surprenant que les groupes évoluent en fonction de l'horizon de prédiction, mais ils restent assez stables en ce qui concerne les groupes les plus nombreux, présents dans les trois premières lignes.

Le Tableau 4 présente une comparaison entre les performances des modèles spécifiques et celles des modèles globaux. Outre l'erreur quadratique moyenne *RMSE*, le coefficient de Nash et le coefficient de persistance, le pourcentage des exemples utilisés comme vecteurs supports (*VS*) est indiqué.

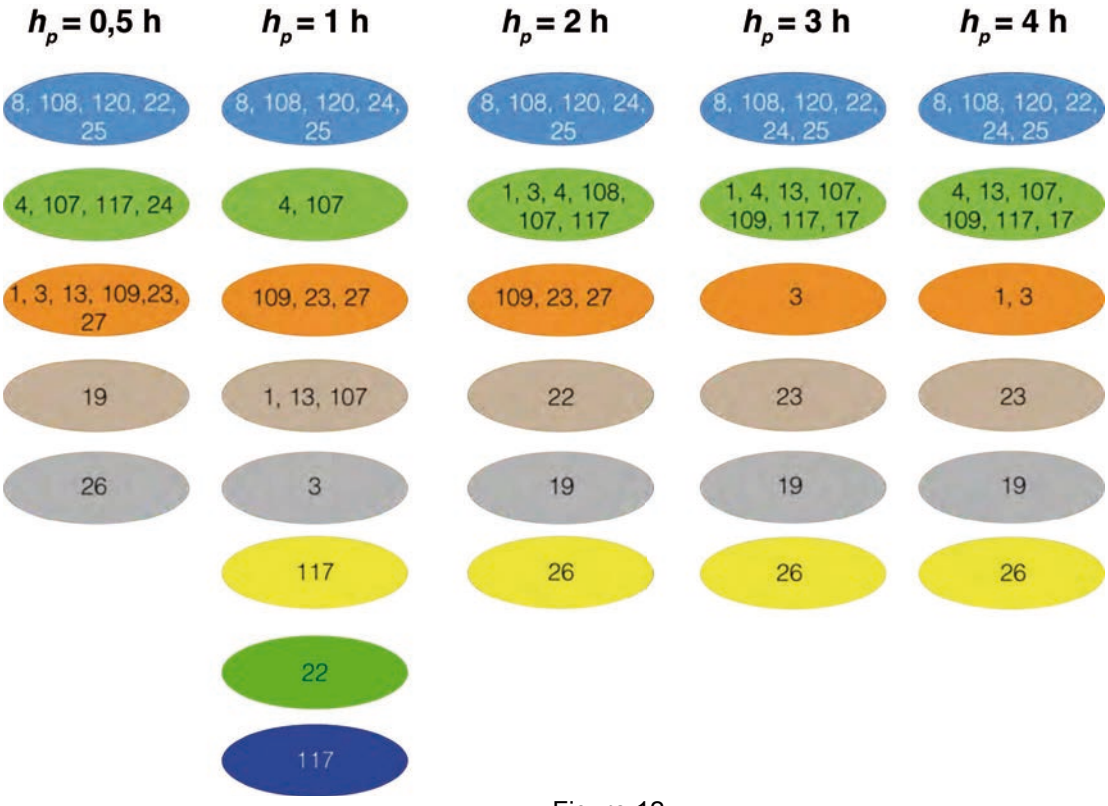

Figure 12

Tableau 4

		h_p=0,5 h	h_p =1 h	h_p =2 h	h_p =3 h	h_p =4 h
Modèle global	*VS* (%)	0,09	0,26	0,48	0,58	0,63
	RMSE	0,08	0,16	0,27	0,34	0,4
	C_N	0,99	0,98	0,94	0,91	0,87
	C_p	0,69	0,65	0,65	0,68	0,68
Modèle 1	*VS* (%)	0,08	0,09	0,21	0,36	0,39
	RMSE	0,06	0,06	0.10	0,17	0,18
	C_N	0,99	0,99	0,98	0,95	0,93
	C_p	0,52	0,83	0,85	0,81	0,85
Modèle 2	*VS* (%)	0,10	0,18	0,51	0,48	0,65
	RMSE	0,07	0,11	0,22	0,19	0,27
	C_N	1,00	0,99	0,95	0,98	0,95
	C_p	0,80	0,83	0,85	0,91	0,88
Modèle 3	*VS* (%)	0,24	0,26	0,43		0,82
	RMSE	0,15	0,13	0,25		0,52
	C_N	0,07	0,99	0,97		0,66
	C_p	1,00	0,84	0,77		0,78
Modèle 4	*VS* (%)		0,30			
	RMSE		0,16			
	C_N		0,98			
	C_p		0,83			

Ces groupes étant déterminés, les hyperparamètres des modèles spécifiques correspondants sont déterminés par validation croisée partielle : les groupes ne

comprenant qu'un seul élément ne sont donc pas pris en considération. Nous disposons finalement de trois modèles locaux pour les prédictions à 30 mn, 2 heures et 4 heures, de deux modèles locaux pour les prédictions à 3 heures, et de quatre modèles locaux pour les prédictions à 1 heure.

Afin de tester les modèles spécifiques et de les comparer au modèle global, six nouveaux événements de test ont été extraits de la base de données. Leurs caractéristiques sont indiquées dans le Tableau 5 ; l'évolution des hauteurs d'eau et des précipitations sont représentées sur la Figure 13.

Tableau 5

Numéro	Date	Durée (heures)	Niveau d'eau maximum (m)
20	28 avril - 2 Mai 2004	95	3,53
21	28-30 janvier 2006	55	3,16
30	30 mars - 4 avril 2004	143	3,39
31	1-5 decembre 2003	117	5,35
32	15-18 novembre 2003	95	3,8
33	1-2 octobre 2003	80	3,23

À titre d'exemple, nous présentons les résultats obtenus sur l'événement de test 32 à l'aide de modèles SVR.

Le Tableau 6 permet une comparaison entre les performances du modèle global et celles des meilleurs modèles spécifiques. On constate que les meilleurs modèles spécifique prédisent mieux l'événement 32 que le modèle global pour tous les horizons sauf h_p = 30 mn, horizon pour lequel les performances de tous les modèles sont satisfaisantes. On peut faire la même observation sur les trois événements de test.

La Figure 14 montre les précipitations et les hauteurs d'eau réelle et prédites par le modèle global et le meilleur modèle spécifique.

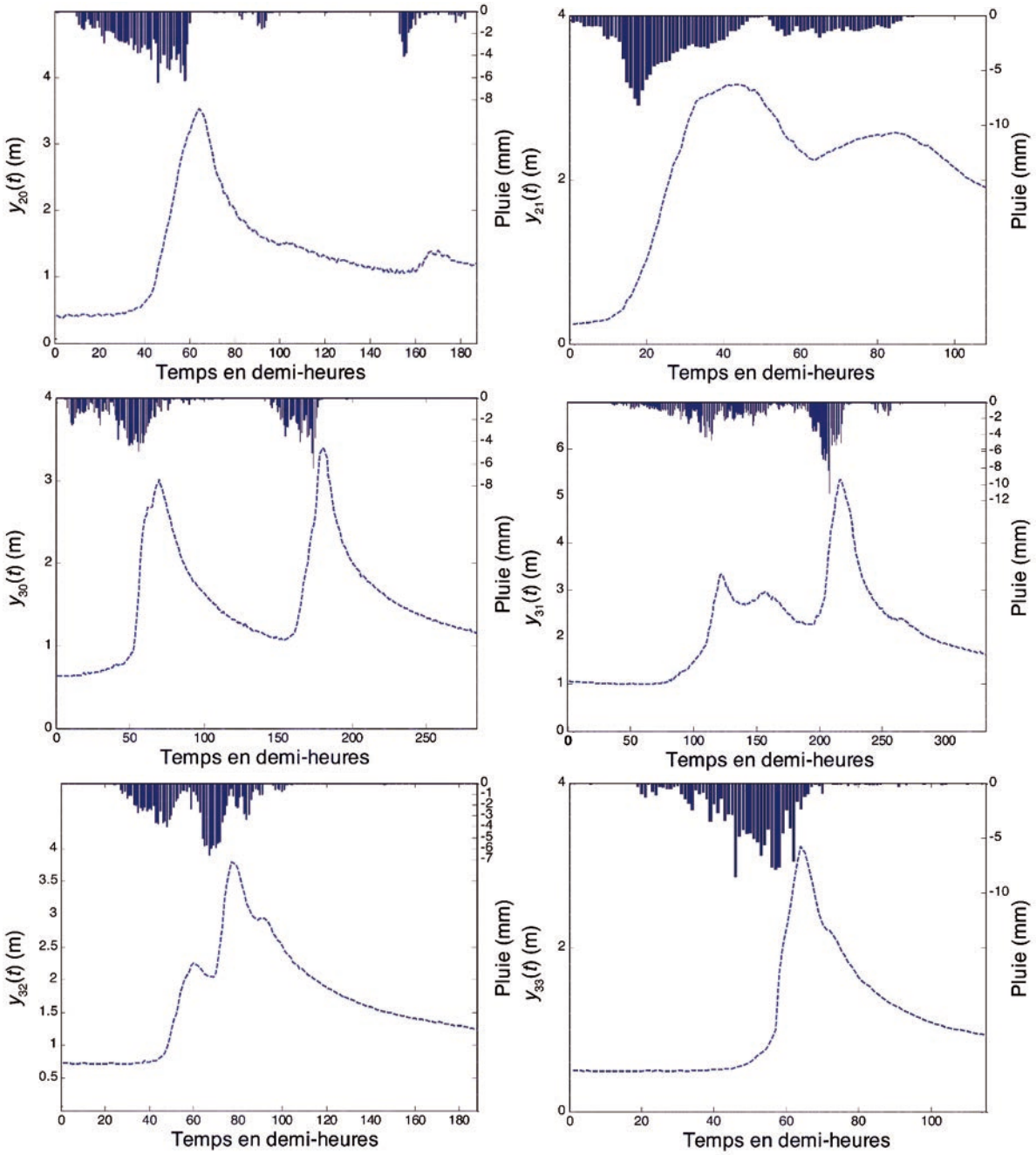

Figure 13

Tableau 6

		h_p = 0,5 h	h_p = 1 h	h_p = 2 h	h_p = 3 h	h_p = 4 h
Modèle global						
	RMSE	0,21	0,31	0,41	0,44	0,44
	C_N	1	0,99	0,95	0,94	0,94
	C_p	0,56	0,44	0,52	0,68	0,79
Meilleur modèle						
	RMSE	0,24	0,25	0,30	0,33	0,33
	C_N	0,99	0,99	0,99	0,98	0,98
	C_p	0,25	0,75	0,87	0,90	0,94

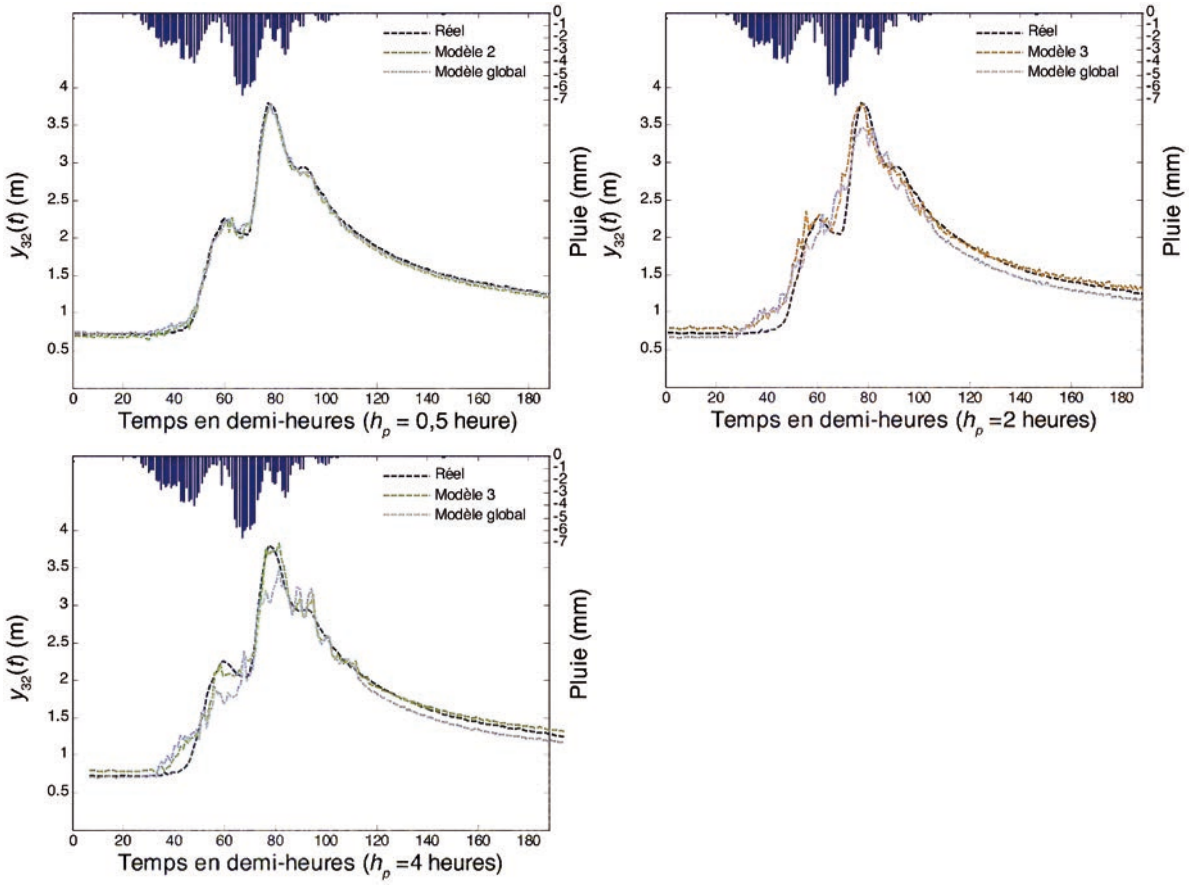

Figure 14

5.2.6 Résultats (modèles LSSVM)

La procédure hiérarchique, mise en œuvre dans la section précédente pour la conception de modèles SVR, a également été appliquée à la conception de modèles LSSVM. Le groupes d'événements trouvés à partir des modèles LSSVM (Figure 15) sont analogues à ceux qui ont été trouvés pour les modèles SVR. Comme dans le cas des SVR, les éléments qui constituent une classe à un seul élément sont les plus intenses de la base de données.

À titre d'illustration, nous présentons, comme pour les prédicteurs SVR, les résultats obtenus en test sur l'événement 32 (Tableau 7 et Figure 16).

Là encore, le meilleur modèle local prédit mieux que le modèle global, pour tous les horizons.

Afin de présenter une vue synthétique des résultats de test, nous montrons sur la Figure 17 les coefficients de Nash en fonction de l'horizon de prédiction, pour les modèles locaux et pour les modèles globaux, pour les exemples de test. On observe que les modèles locaux ont toujours des performances supérieures ou égales à celles des modèles globaux.

Enfin, afin de vérifier la validité de cette approche, nous avons effectué des groupements aléatoires d'événements, et avons vérifié que les modèles locaux créés par apprentissage à partir de ces groupes ne donnent pas de meilleures prédictions que les modèles globaux.

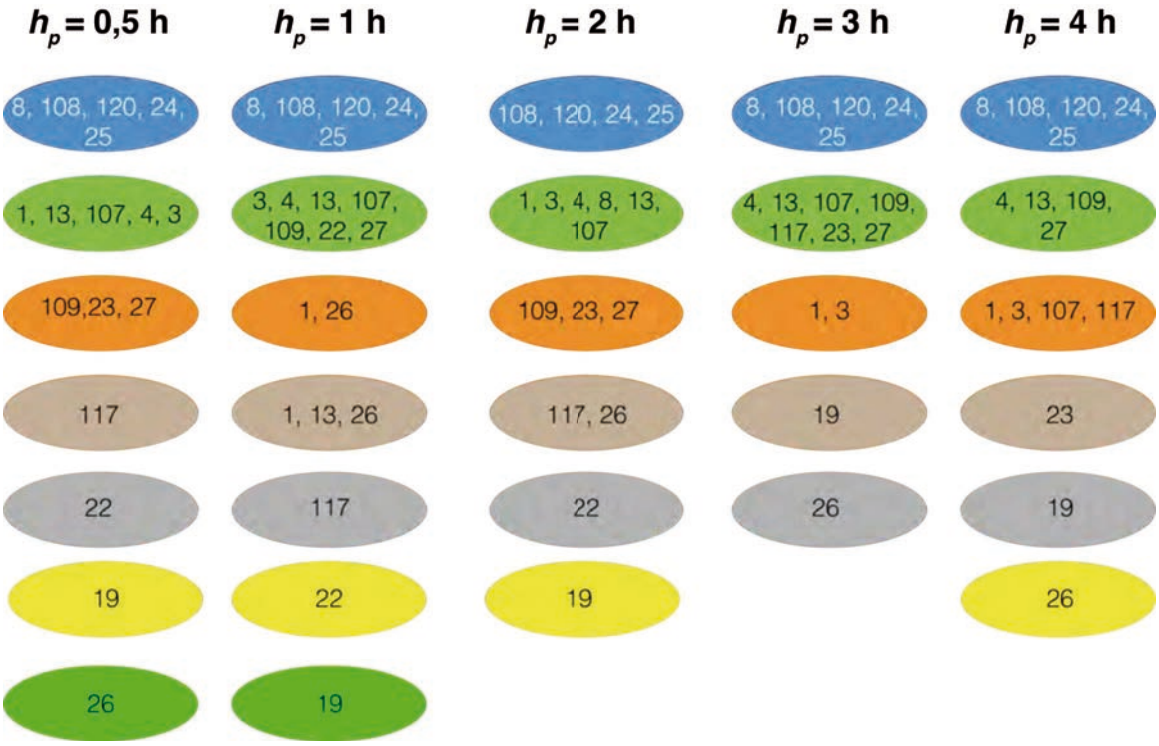

Figure 15

Tableau 7

		h_p = 0,5 h	h_p = 1 h	h_p = 2 h	h_p = 3 h	h_p = 4 h
Modèle global						
	RMSE	0,18	0,28	0,39	0,45	0,46
	C_N	1,00	0,99	0,96	0,93	0,93
	C_p	0,74	0,65	0,59	0,63	0,75
Meilleur modèle						
	RMSE	0,17	0,25	0,29	0,31	0,42
	C_N	1,00	0,99	0,99	0,98	0,95
	C_p	0,82	0,75	0,88	0,92	0,83

5.2.7 Conclusions

Nous avons montré que la conception de modèles spécifiques à des classes d'événements définies par une méthode de classification hiérarchique ascendante permet d'obtenir de meilleurs résultats que la conception d'un modèle global.

Il reste à trouver une méthode permettant de discerner le plus tôt possible, en cours d'événement, quel est le modèle prédictif le plus pertinent parmi tous les modèles spécifiques disponibles. Les premiers essais de mise en œuvre d'une telle démarche ont donné des résultats encourageants, mais de vrais tests opérationnels seront nécessaires, sur une variété d'événements à venir pour pouvoir définir une méthodologie rigoureuse.

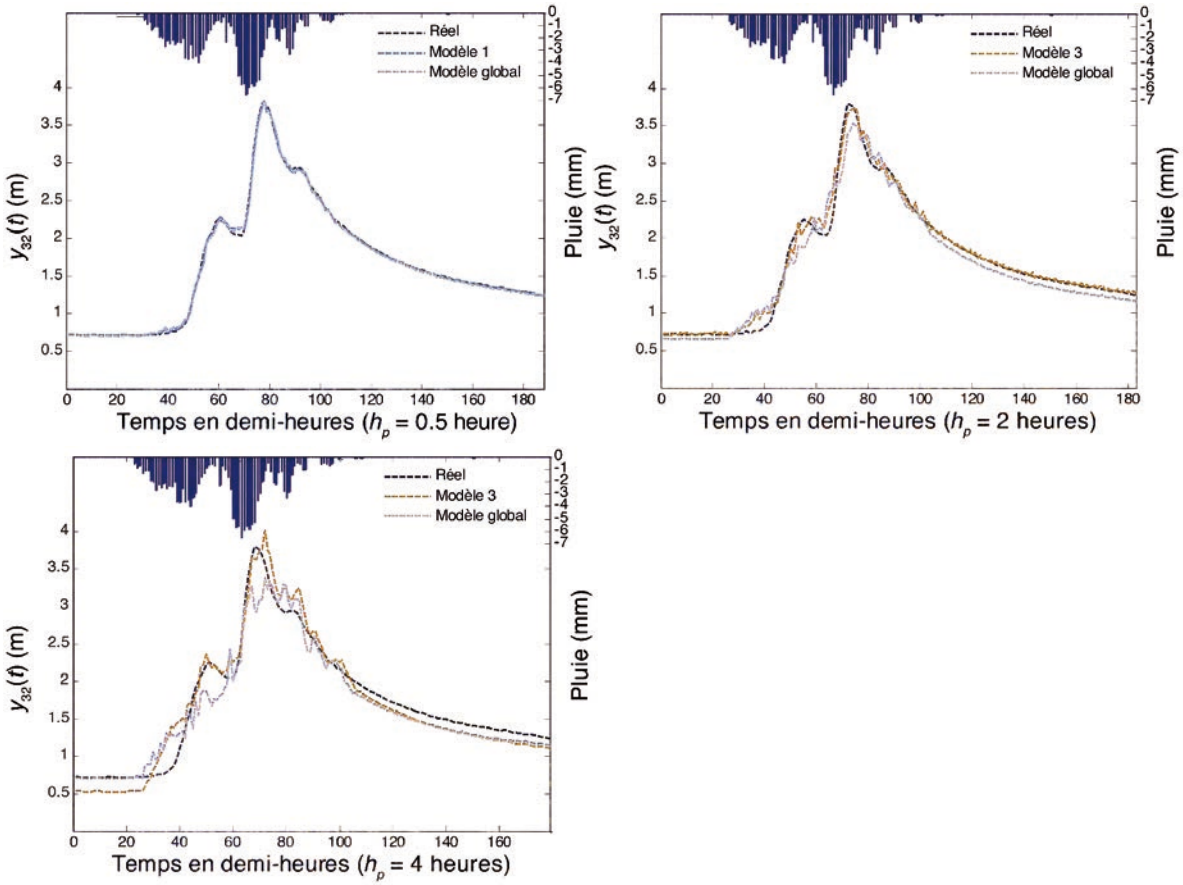
Réel
Modèle 1
Modèle global
$y_{32}(t)$ (m)
Pluie (mm)
Temps en demi-heures (h_p = 0.5 heure)
Réel
Modèle 3
Modèle global
$y_{32}(t)$ (m)
Pluie (mm)
Temps en demi-heures (h_p = 2 heures)
Réel
Modèle 3
Modèle global
$y_{32}(t)$ (m)
Pluie (mm)
Temps en demi-heures (h_p = 4 heures)

Figure 16

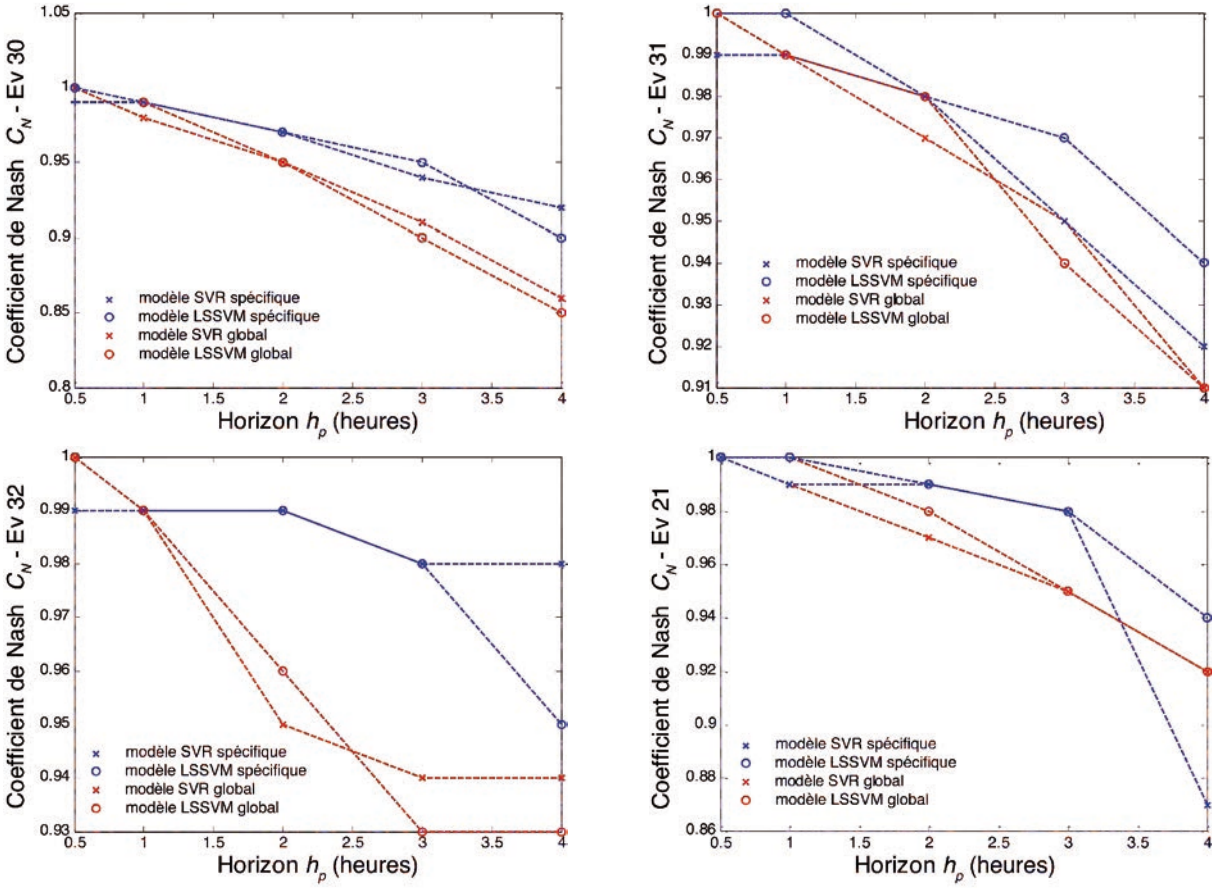

Figure 17

5.1 PRÉDICTEURS FONDÉS SUR D'AUTRES MODÈLES POSTULÉS

Rappelons que les résultats établis dans les sections précédentes sont fondés (section 4.1) sur l'hypothèse que le processus peut être modélisé par des modèles postulés de la forme

$$y\left(k+h_p\right)=f_{h_p}\left(\boldsymbol{\varphi}\left(k\right)\right)+d\left(k\right),$$

où $d\left(k\right)$ est une réalisation d'une variable aléatoire d'espérance mathématique nulle qui modélise l'ensemble des bruits et des perturbations, et où

$$\boldsymbol{\varphi}(k)=\left[y\left(k\right),y\left(k-1\right),\cdots,y\left(k-n_a\right),\mathbf{u}\left(k\right)^T,\cdots,\mathbf{u}\left(k-n_b\right)^T\right]^T.$$

Dans les sections précédentes, nous avons donc cherché des prédicteurs de la forme

$$g_{h_p}\left[\boldsymbol{\varphi}\left(k\right),\mathbf{w}\right]$$

où le vecteur des paramètres **w** est estimé, à partir des exemples disponibles, de telle sorte que le prédicteur soit aussi proche que possible de la fonction inconnue $f_{h_p}\left(\boldsymbol{\varphi}\left(k\right)\right)$.

Ces prédicteurs ne peuvent pas être utilisés comme simulateurs de crues : en effet, $\boldsymbol{\varphi}(k)$ contient les valeurs des hauteurs *mesurées* jusqu'à l'instant courant *k*. Si l'on

voulait, à l'instant k, effectuer des prédictions au-delà de l'horizon $k + h_p$, il faudrait disposer des mesures aux instants ultérieurs à k, ce qui n'est évidemment pas possible.

Supposons que nous souhaitions réaliser un prédicteur utilisable en simulateur avec un pas de 30 mn (h_p = 1 ; nous omettrons dans la suite l'indice h_p). Nous cherchons alors un prédicteur de la forme $g\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]$, avec à présent

$$\boldsymbol{\varphi}(k)=\left[g\left[\boldsymbol{\varphi}(k-1),\mathbf{w}\right],\cdots,g\left[\boldsymbol{\varphi}(k-n_a),\mathbf{w}\right],\mathbf{u}(k)^T,\cdots,\mathbf{u}(k-n_b)^T\right]^T$$

En remplaçant, dans l'expression de $\boldsymbol{\varphi}(k)$, les valeurs de hauteurs *mesurées* par les valeurs de hauteurs *prédites* aux instants précédents, nous rendons le prédicteur *récurrent* puisque la prédiction $g\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]$ effectuée à l'instant k dépend des prédictions effectuées aux n_a instants précédents. L'objectif de l'apprentissage est alors d'estimer les valeurs des paramètres **w** de telle sorte que le prédicteur g soit aussi proche que possible de la fonction inconnue f.

Il est important de remarquer que cette modification du vecteur $\boldsymbol{\varphi}(k)$ n'est pas une simple commodité d'écriture. Elle correspond en réalité à remplacer l'hypothèse « bruit d'état » (qui caractérisait notre modèle postulé dans les sections précédentes) par l'hypothèse « bruit de sortie » : nous remplaçons l'hypothèse selon laquelle le processus peut être correctement modélisé par

$$y(k+1)=f\left(\boldsymbol{\varphi}(k)\right)+d(k),\ \text{avec}\ \boldsymbol{\varphi}(k)=\left[y(k),y(k-1),\cdots,y(k-n_a),\mathbf{u}(k)^T,\cdots,\mathbf{u}(k-n_b)^T\right]^T$$

par l'hypothèse selon laquelle le processus peut être correctement modélisé par

$$y(k+1)=x(k)+d(k),$$

$$\text{avec}\ x(k)=f\left(\boldsymbol{\varphi}(k)\right)\ \text{et}\ \boldsymbol{\varphi}(k)=\left[x(k-1),\cdots,x(k-n_a),\mathbf{u}(k)^T,\cdots,\mathbf{u}(k-n_b)^T\right]^T$$

Dans la première hypothèse, le bruit $d(k)$ intervient directement dans la dynamique du processus (d'où le terme de « bruit d'état ») ; dans la seconde hypothèse, la dynamique est déterminée uniquement par $x(k)$, dans l'expression de laquelle le bruit n'intervient pas (d'où le terme de « bruit de sortie »).

En admettant la seconde hypothèse, il est naturel de chercher un prédicteur récurrent de la forme indiquée plus haut

$$g\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]\ \text{avec}\ \boldsymbol{\varphi}(k)=\left[g\left[\boldsymbol{\varphi}(k-1),\mathbf{w}\right],\cdots,g\left[\boldsymbol{\varphi}(k-n_a),\mathbf{w}\right],\mathbf{u}(k)^T,\cdots,\mathbf{u}(k-n_b)^T\right]^T$$

et d'estimer le vecteur des paramètres **w** de telle manière que la fonction g soit aussi proche que possible de la fonction inconnue f. En effet, si l'apprentissage était parfait, c'est-à-dire si l'on réussissait, par apprentissage, à obtenir le prédicteur

$$g\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]=f\left(\boldsymbol{\varphi}(k)\right)=x(k)\ \forall k$$

l'erreur de prédiction serait égale au bruit puisque $y(k+1)-g\left[\boldsymbol{\varphi}(k),\mathbf{w}\right]=d(k)\ \forall k$. Le prédicteur serait donc optimal.

En résumé :

- un prédicteur non récurrent tel que ceux qui ont été mis en œuvre dans les sections 5.1 et 5.2 peut être optimal (variance de l'erreur de prédiction égale à la variance du bruit) si l'hypothèse « bruit d'état » est vraie,
- un prédicteur récurrent tel que ceux dont la mise en œuvre est décrite dans cette section peut être optimal si l'hypothèse « bruit de sortie » est vraie.

Dans le second cas, le prédicteur peut être utilisé comme simulateur puisque son apprentissage est celui d'un prédicteur récurrent ; dans le premier cas, l'apprentissage du prédicteur est celui d'un prédicteur non récurrent, donc son utilisation en simulateur est risquée puisqu'elle nécessite de remplacer, dans les variables du modèle, les hauteurs d'eau mesurées par les hauteurs d'eau qui ont été prédites par un prédicteur qui n'a pas été conçu dans ce but.

Dans cette section, nous présentons l'apprentissage de prédicteurs récurrents obtenus par un apprentissage de type LSSVM.

Suivant la même procédure et avec les mêmes notations que dans la section 4.2.2 (LSSVM non dynamiques), la solution du problème d'optimisation correspond au point selle du Lagrangien suivant :

$$\begin{aligned} L(\mathbf{w},b,\mathbf{e};\boldsymbol{\alpha}) &= J(\mathbf{w},b,\mathbf{e}) + \sum_{i=1}^{N}\alpha_i\left[y_i - e_i - \mathbf{w}^T\boldsymbol{\varphi}(z_i) - b\right] \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{2}\sum_{k=1}^{N}e_k^2 + \sum_{i=1}^{N}\alpha_i\left[y_i - e_i - \mathbf{w}^T\boldsymbol{\varphi}(z_i) - b\right] \end{aligned} \tag{6}$$

où $z_{k+i} = \left[y_{k+i-1} - e_{k+i-1}, \ldots y_k - e_k, \ldots y_{k+i-n_a} - e_{k+i-n_a}, u_{i+k-1}\right]$

Les conditions nécessaires à l'obtention d'un minimum sont données par :

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{N}\alpha_i\boldsymbol{\varphi}(z_i) = 0 \qquad (a)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N}\alpha_i = 0 \qquad (b)$$

$$\frac{\partial L}{\partial e_k} = Ce_k - \alpha_k - \sum_{i=1}^{n_a}\alpha_{i+k}\frac{\partial}{\partial e_k}\left[\mathbf{w}^T\boldsymbol{\varphi}(z_{i+k}) - b\right] = 0, \quad k = 1,\ldots,N \qquad (c)$$

$$\frac{\partial L}{\partial \alpha_k} = y_k - e_k - \mathbf{w}^T\boldsymbol{\varphi}(z_k) - b = 0, \quad k = 1,\ldots,N \qquad (d)$$

(8)

La difficulté réside ici dans la contrainte (8.c). En effet, contrairement au cas des prédicteurs non récurrents, le vecteur $\boldsymbol{\varphi}(z_{i+k})$ dépend de e_k ; la dérivée $\frac{\partial}{\partial e_k}\mathbf{w}^T\boldsymbol{\varphi}(z_{i+k})$ n'est donc plus nulle, et son calcul n'est pas trivial. Pour résoudre ce problème, des hypothèses simplificatrices ont été proposées.

5.1.1 LSSVMs récurrentes non régularisés [Suykens, 2000]

Le calcul de $\frac{\partial L}{\partial e_k}$ étant très coûteux, le problème d'optimisation est simplifié en considérant que la constante de régularisation C est infinie. Le problème d'optimisation devient alors :

$$\underset{\mathbf{e},b;\boldsymbol{\alpha}}{\text{minimiser}} \frac{1}{2}\sum_{k=1}^{N} e_k^2$$

$$\text{sous les contraintes} \begin{cases} \sum_{i=1}^{N} \alpha_i = 0 \\ y_k^p - e_k - b - \sum_{i=1}^{N} \alpha_i K(z_i, z_k) = 0, \quad k = 1,2,.....N \end{cases} \tag{9}$$

5.1.2 LSSVMs récurrentes régularisées [Lucea, 2006]

En remplaçant l'équation (8.a) dans les équations (6) et (8.c), les équations (6) et (8) s'écrivent alors :

$$\underset{b,\mathbf{e};\boldsymbol{\alpha}}{\text{minimiser}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j K(z_i, z_j) - \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N} \alpha_i (y_i - e_i - b) \tag{10}$$

$$s.c \begin{cases} \sum_{i=1}^{N} \alpha_i = 0 & (a) \\ Ce_k - \alpha_k - \sum_{i=1}^{n_a}\sum_{j=1}^{N} \alpha_{i+k}\alpha_j \frac{\partial}{\partial e_k} K(z_j, z_{k+i}) = 0, \quad k = 1,2,.....N & (b) \\ y_k - e_k - \sum_{i=1}^{N} \alpha_i K(z_k, z_i) - b = 0, \quad k = 1,...,N & (c) \end{cases} \tag{11}$$

La contrainte (11.b) peut être approchée par la contrainte suivante :

$$Ce_k - \alpha_k - \sum_{i=1}^{N_a} a_{i+k} \sum_{j=1}^{N} a_j b_{k,i,j} \left[z_j(i) - y_k + e_k \right]$$

où $b_{k,i,j} = -\frac{2}{\sigma^2} e^{-\frac{\left\| z_j - z_{k+i} \right\|^2}{\sigma^2}}$.

5.1.3 Résultats : utilisation de modèles récurrents en simulateurs ou en simples prédicteurs

La mise en œuvre des LSSVM récurrentes régularisées pour la simulation des crues s'est révélée décevante en raison de temps de calcul très élevés (plusieurs heures pour modéliser un seul événement) et surtout en raison de résultats de très mauvaise qualité.

La mise en œuvre de ces mêmes modèles comme simples prédicteurs à un horizon de 30 mn n'a pas permis d'obtenir des résultats de qualité comparable à ceux qui ont été obtenus à l'aide des modèles prédicteurs non récurrents.

Rappelons que l'utilisation d'un prédicteur récurrent, quelle que soit sa nature (SVM, réseau de neurones, ou simplement prédicteur linéaire) est fondée sur l'hypothèse de la présence d'un bruit de sortie, qui n'a pas d'influence sur la dynamique du

processus. Si une telle situation est souvent réalisée pour des systèmes artificiels relativement simples, elle paraît peu vraisemblable pour un processus naturel aussi compliqué que celui des crues subites, où de nombreuses sources de perturbations, notamment sur l'état, sont certainement présentes. L'échec des tentatives de simulation des crues présentes dans notre base de données à l'aide de prédicteurs récurrents confirme cette conjecture.

5.1.4 Utilisation de modèles non récurrents en simulateurs

Rappelons que les modèles non récurrents sont conçus pour avoir comme variables les hauteurs d'eau mesurées, de sorte qu'ils ne peuvent pas être utilisés directement pour réaliser des simulateurs. On peut néanmoins essayer d'utiliser ces prédicteurs comme simulateurs, en remplaçant les hauteurs mesurées par les hauteurs prédites dans le vecteur des variables du modèle.

La Figure 18 montre les résultats obtenus sur quatre événements en utilisant comme simulateur le prédicteur non récurrent dont l'apprentissage a été fait sur ce même modèle, avec un horizon de 30 mn. La hauteur d'eau mesurée est utilisée comme variable pour la première prédiction ; pour les prédictions ultérieures, les hauteurs d'eau mesurées sont remplacées par les hauteurs d'eau prédites.

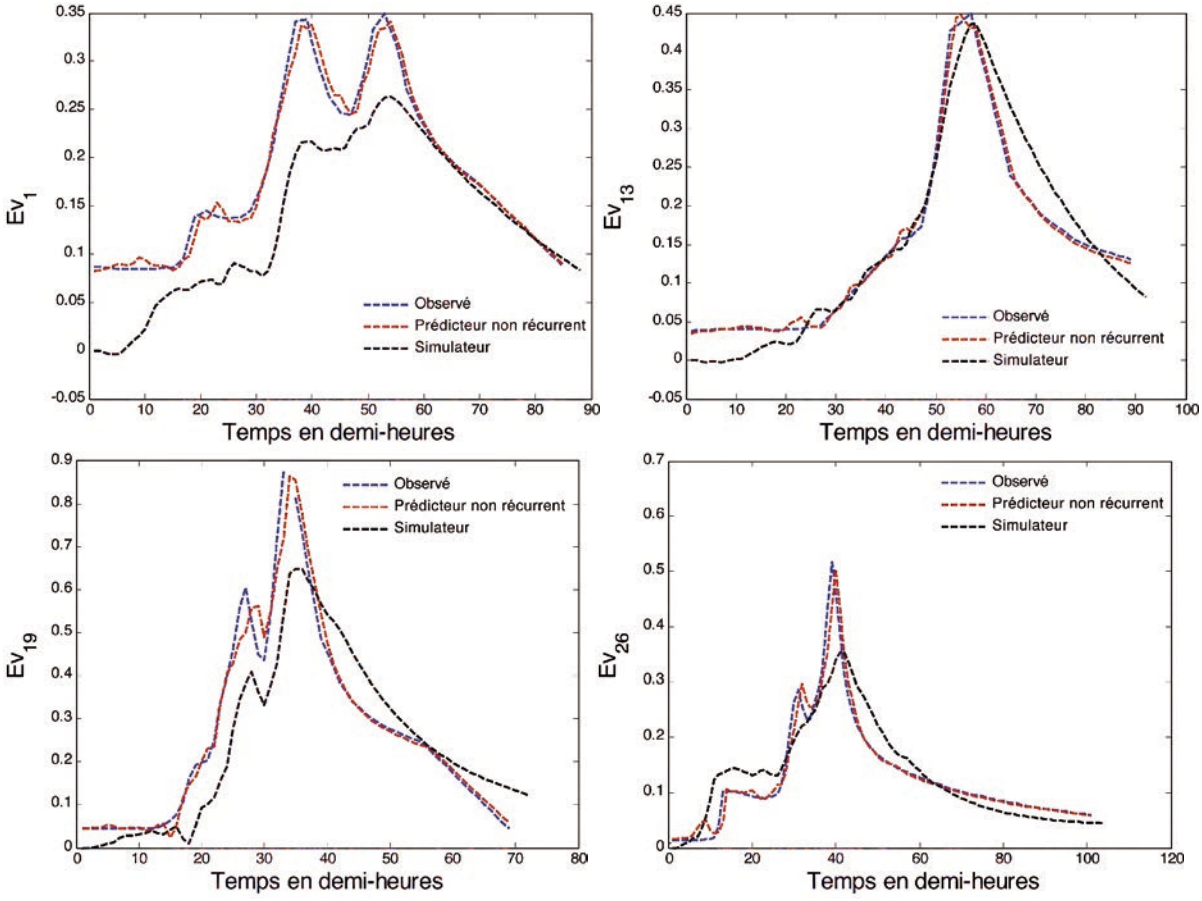

Figure 18

Les résultats obtenus ne sont pas aussi satisfaisants que ceux qui ont été décrits dans les sections 5.1 et 5.2, mais ils montrent que l'on peut obtenir des résultats certes médiocres, mais pas inacceptables, en simulation, à partir de prédicteurs non récurrents.

6. CONCLUSIONS ET PERSPECTIVES

L'objectif principal de la tâche T4 était la mise en œuvre de modèles dynamiques à noyaux, conçus par apprentissage statistique, pour la prédiction des crues en l'absence de prédiction de pluies. Le lieu d'application était le bassin versant du Gardon d'Anduze.

Une étude détaillée a été menée à l'aide de deux modèles postulés

- hypothèse NARX avec bruit d'état,
- hypothèse NARX avec bruit de sortie.

Les prédicteurs correspondants à ces deux hypothèses ont été construits, selon trois variantes de méthodes à noyaux :

- régression à vecteurs supports SVR,
- régression Least-Squares Support Vector Machines (LSSVM),
- régression linéaire par morceaux.

De plus, une méthode de filtrage spatial des pluies, inspirée des connaissances expertes des hydrogéologues, a été mise en œuvre avec succès : elle a contribué à diminuer la complexité des modèles et, en conséquence, à améliorer leurs capacités de généralisation.

Deux stratégies de modélisation ont été mises en œuvre :

- une stratégie conventionnelle, qui consiste à créer des modèles globaux à partir de tous les événements disponibles à l'exception de ceux qui étaient réservés pour le test,
- une stratégie originale, qui consiste à effectuer une classification non supervisée des événements de crues en différents groupes, puis à effectuer l'apprentissage de modèles locaux pour chacun de ces groupes.

L'application de ces modèles locaux à des événements de test réservés à cet effet a montré qu'il existe toujours au moins un modèle local qui est plus efficace que le modèle global.

Nous avons également montré que les modèles prédictifs construits à partir de l'hypothèse NARX avec bruit d'état sont beaucoup plus efficaces que les modèles prédictifs construits à partir de l'hypothèse NARX avec bruit de sortie. Ceci n'est pas très surprenant compte tenu de la grande complexité des phénomènes que nous cherchons à modéliser ici : il est très probable que les bruits et perturbations agissent directement sur l'état du processus, c'est-à-dire sur sa dynamique. Il était néanmoins important de confirmer expérimentalement cette conjecture.

Toutes les prédictions étant effectuées en l'absence de prévisions de pluie, on observe toujours une dégradation de la précision des prévisions avec l'horizon de prédiction. Les performances, exprimées par plusieurs indicateurs, restent très satisfaisantes jusqu'à un horizon de quatre à cinq heures, ce qui correspond à peu près au temps de transit estimé de l'eau de pluie dans le bassin versant : pour des prévisions à des horizons supérieurs, les données de pluie dont on dispose ne sont plus pertinentes puisque l'essentiel de l'eau de pluie tombée quatre à cinq heures plus tôt a déjà atteint l'exutoire.

Le principal problème qui reste ouvert est la sélection du modèle le plus approprié, parmi les modèles locaux disponibles, en temps réel pendant le déroulement d'un événement. Ceci nécessitera

- une interprétation physique aussi précise que possible, par des hydrogéologues, des groupes d'événements créés par classification non supervisée,
- une phase de tests opérationnels pour établir une méthodologie permettant de détecter le plus tôt possible le groupe d'événements auquel se rattache l'événement en cours.

7. RÉFÉRENCES

[Boukharouba, 2009] K. Boukharouba, L. Bako et S. Lecoeuche, *Identification of piecewise affine systems based on Dempster-Shafer theory*, IFAC Symposium on System Identification, Saint Malo, France, 2009.

[Krause, 2005] P. Krause, D.P. Boyle, *Comparison of different efficiency criteria for hydrological model assessment*, Advances in Geosciences, 5, 89–97, 2005.

[Ljung, 1999] L. Ljung, *System identification: theory for the user*. Prentice-Hall, 1999.

[Lucéa, 2006] M. Lucéa, *Modélisation dynamique par réseaux de neurones et machines à vecteurs supports : contribution à la maîtrise des émissions polluantes de véhicules automobiles*, Thèse de Doctorat de l'Université Pierre et Marie Curie, 2006.

[Smola, 2004] A.J. Smola, B. Schöllkopf, *A tutorial on support vector regression,* Statistics and Computing vol. 24, 199-222, 2004.

[Suykens, 1999] J.A.K. Suykens, J. Vandewalle, *Least squares support vector machine classifiers*, Neural Processing Letters vol. 9, 293-300, 1999.

[Suykens, 2005] J.A.K. Suykens, J. Vandewalle, *Recurrent least squares support vector machines*, IEEE Transactions on Circuits and Systems-I 47, 1109-1114, 2005.

[Thiessen, 1911] A. Thiessen, *Precipitation averages for large areas*, Mon. Weather Rev. 39, 1082–1084, 1911.

[Vapnik, 1995] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag,1995.

ANNEXE

FLASH FLOOD FORECASTING USING SUPPORT VECTOR REGRESSION: AN EVENT CLUSTERING BASED APPROACH

Khaled Boukharouba, Pierre Roussel, Gérard Dreyfus, Anne Johannet

IEEE INTERNATIONAL WORKSHOP
ON MACHINE LEARNING FOR SIGNAL PROCESSING,
Southampton, UK, 2013

2013 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2013, SOUTHAMPTON, UK

FLASH FLOOD FORECASTING USING SUPPORT VECTOR REGRESSION: AN EVENT CLUSTERING BASED APPROACH

Khaled Boukharouba[1], *Pierre Roussel*[1], *Gérard Dreyfus*[1], *Anne Johannet*[2]
(1) ESPCI Paristech, SIGnal processing and MAchine learning (SIGMA) Lab, 10 rue Vauquelin, 75005 Paris, France
(2) Ecole des Mines d'Alès, 6 av. de Clavières, 30319 Alès Cedex, France.

ABSTRACT

We present a new machine learning approach to flash flood forecasting in the absence of rainfall forecasts, based on the agglomerative hierarchical clustering of flood events. Each cluster contains events whose models have similar behaviors. Specific Support Vector Regression models are then trained from each cluster. The test results show that a specific model may be more accurate than a general model trained from all floods present in the training database.

Index Terms— Flash flood forecasting, Support vector regression, Hierarchical clustering, NARX model, Thiessen polygon.

1. INTRODUCTION

Flash floods are intense floods that occur within a few hours after a strong precipitation on small (some hundreds of km^2), high-slope watersheds, resulting in flows increasing by thousands of m^3/s in a very short time. These floods cause major damages each year in the south of France: the last flash flood of the Gardon in 2002 and the Var in 2010 killed nearly 50 persons and caused more than two billion euros of damage. Faced with this major risk, the French Ministry in charge of Sustainable Development created in 2003 the national center for flood forecasting and warning SCHAPI (Service Central d'Hydrométéorologie et d'Appui à la Prévision des Inondations), which is in charge of the "vigicrues" surveillance service available via the Internet (http://www.vigicrues.gouv.fr/). This warning system must be able to predict both the time of the occurrence of the event and its magnitude. Forecasting flash floods is a challenging problem that requires taking into account specific, coupled meteorological and hydrological phenomena. Moreover, the processes governing the relationships between rainfall and water flow (or level) are not well understood for these steep-sloped, heterogeneous basins. Therefore, black-box modeling can be a viable alternative to physics-based models for predicting these complex events.

Support Vector Machines (SVM) were seldom used for flash flood forecasting. Forecasts of slow and medium floods by SVMs were reported more frequently. Sivapragasam et al. [1] proposed a prediction technique based on Singular Spectrum Analysis (SSA) coupled with Support Vector Machine (SVM). This technique was applied to predict the Tryggevaelde catchment runoff data (Denmark) and the Singapore rainfall data as case studies. The same authors proposed in [2] to define three flow ranges (low, medium and high flow regions). An SVM-based flow forecasting model was then applied for each flow region. The relationship between rainfall and river discharge of the Fuji river was modeled by support vector radial basis function networks (SVRBFNs) [3]. Yu et al. [4] combined chaos theory with support vector machine (EC-SVM) for the analysis of chaotic time series; the performance of the EC-SVM was assessed on the Tryggevaelde catchment, Denmark and the Mississippi River at Vicksburg. Other reports of the use of SVM for flood forecasting can be found in [5] [6] [7] [8] [9]. Concerning flash floods, Yu et al. [10] established a real-time stage forecasting model based on support vector machine. A two-step grid search method with cross validation was used to tune the SVM model hyperparameters; the grid search method was first applied using a coarse grid search to determine the best region of the feasible parameter space, and then a finer grid search was performed in the feasible region to find the optimal parameters. The model was tested on flood events in Lan-Yang River, Taiwan, for 1h to 6h range forecasts. The authors claim that the validation results indicate good forecasting performance.

In this paper, we propose a new approach, which consists in clustering the flood events into different groups. Each cluster contains events such that a model trained on a single event of the cluster can predict satisfactorily the events of the same cluster. A specific regression model is trained from each group. All models thus obtained are tested on the same test set. We show that, in most cases, specific models provide more accurate predictions on the test set than a general model trained from all floods present in the training database.

The paper is organized as follows. In Section 2, we recall briefly the NARX hypothesis used for our predictive models, and we formulate the prediction problem and the event clustering problem. Section 3 presents the watershed under investigation and the available database. In Section 4, we recall briefly the support vector regression method.

Section 5 describes the clustering method. Finally, results are presented and discussed in Section 6.

2. PROBLEM STATEMENT

The purpose of this study is the modeling of the (nonlinear) rainfall-water level relationship. Nonlinear AutoRegressive eXogenous models (NARX) [11] are good candidates to perform the task. We assume that the dynamics of the process can be described satisfactorily by the discrete-time equation:

$$y(k+h) = f_h(x(k)) + d(k) \quad (1)$$

where f_h is the regression function, $x(k)$ is the regression vector at discrete time kT (T is the sampling period and k is a nonnegative integer), and $d(k)$ is a random variable that models noise and disturbances. The regression vector $x(k)$ contains the measured past values of the quantity of interest on a window of length n_a, and the past values of the exogenous variables $u(k)$ that control the quantity of interest, on a window of length n_b. In our case $y(k)$ is the water level, while $u(k)$ are the past rainfall values measured at six rain gauges; therefore, we assume that the water level can be predicted on horizon h in the absence of any waterfall prediction. Under this assumption, a predictive model of the form

$$\hat{y}(k+h) = g_h(x(k), w)$$

is designed, where g_h is a postulated function with parameter vector w. The parameters are estimated by training from the available data; ideally, if training was perfect, i.e. if function g_h was identical to the regression function f_h, the predictive model would provide the value of the water level that would be observed in the absence of noise and disturbances.

The problem that we address here can be summarized as follows: *given a collection of N data points $\{x(k), y(k+h\}$, estimate the parameters of function g_h (model M_h) for different prediction horizons, in the absence of rainfall forecasts.*

As mentioned in the introduction, our strategy aims at developing a finite number of specific models. Each model represents a specific behavior of the flash floods. This collection of models will hopefully span the relevant "operating points" of the flash floods of the considered area. Thus, the modeling problem becomes: *given a collection of N data points $\{x(k), y(k+h\}$, (i) cluster the events into different groups $i = 1,...C$, so that each group i contains events with similar behavior, (ii) estimate the parameters of the corresponding models $M_h(i)$ for different prediction horizons, in the absence of rainfall forecasts.*

In the present paper, we build the specific models $M_h(i)$ from the different groups of events, and we compare their performance to the performance of a global model M_h trained from the data of all the events of the training database.

3. THE GARDON D'ANDUZE WATERSHED

The watershed of interest in the present study is the Gardon d'Anduze, a sub-watershed of the Gardons, located in the south-east of France. The basin extends over an area of 524km^2 with a mean slope of 10%; its elevation varies from 111m to 1366m. The response time of this watershed is estimated to range from 2 to 4 hours. Rainfalls in that area are measured by six rain gauges; the water level is measured by one gauge station located at Anduze. Flash flood prediction for this watershed was performed previously by neural network models [12], [13].

3.1. Database

Rainfall and water level data were collected from 1993 to 2008. The resulting database contains 23 significant flood events. Table 1 and table 2 summarize all the events used in the present study for training and testing the models; the date, the duration and the peak water level are reported. The sampling period is 30 minutes. Table 1 shows that the most intense event of the database is event Ev_{14} of 2002, whose peak water level reached 9.7m. The hyperparameters of the models were estimated by cross-validation on the training/validation set.

3.2. Data preprocessing

The rain gauge data (exogenous variables) were normalized so as to have a maximum of 0.9 in each input vector: $u_n = (0.9 / \max(u)) \times u$, where u_n and u are the normalized and non-normalized variables respectively. The water level was normalized similarly, with the additional constraint of forcing the minimum normalized water level to be zero: $y_n = (0.9 / \max(y)) \times (y - \min(y))$.

In order to reduce the dimensionality of the input vector, the method of Thiessen polygons [14] was used, reducing the six variables provided by the rain gauges to a single variable: a "weighted average precipitation" is computed, based on the relative areas of each region in the Thiessen polygon (Table 3).

4. SUPPORT VECTOR REGRESSION (SVR)

Support vector regression is a kernel-based method with a built-in regularization mechanism, similar to that of support vector machine classifiers (see for instance [15]). The regression problem is expressed as the following constrained optimization problem:

$$\frac{1}{2}\|w\|^2 + C\sum_{k=1}^{N} L_\varepsilon\left(y(k), g(x(k), w)\right). \quad (3)$$

L_ε is the ε-insensitive loss function defined as:
$L_\varepsilon = 0$ if $|y(k) - g(x(k), w)| < \varepsilon$,
$L_\varepsilon = |y(k) - g(x(k), w)| - \varepsilon$ otherwise.

A high value of the hyperparameter C means that we insist on finding a model, however complex, that predicts the examples of the training set within an accuracy ε, while a low value of C means that we are ready to trade off the accuracy of the prediction of training data against a low model complexity.

Table 1: List of events of the training/validation database.

	Date	Duration (hours)	Maximum water level (m)
Ev_1	September, 21-24, 1994	35	3.71
Ev_2	October, 4-5, 1995	54	5.34
Ev_3	October, 13-14,1995	92	5
Ev_4	November, 10-12, 1996	82	2.71
Ev_5	November, 5-7, 1997	74	4.2
Ev_6	November. 26-27, 1997	66	2.58
Ev_7	December, 18-19, 1997	104	5.37
Ev_8	October, 20-21, 1999	34	3.64
Ev_9	September, 28-29, 2000	46	4.8
Ev_{10}	November, 12-14, 2000	71	2.77
Ev_{11}	September, 24-25, 2006	23	2.24
Ev_{12}	October, 19-20, 2006	55	6.61
Ev_{13}	November, 17-18, 2006	34	2.75
Ev_{14}	September, 8-9, 2002	29	9.71
Ev_{15}	November, 20-23, 2007	70	2.69
Ev_{16}	October, 21-23, 2008	43	5.57
Ev_{17}	31st October – 3rd November 2008	81	5.53

Table 2: List of events of the test database.

	Date	Duration (hours)	Maximum water level (m)
Ev_{18}	28th April - 2nd May, 2004	95	3.53
Ev_{19}	January, 28-30, 2006	55	3.16
Ev_{20}	30th Mars - 4th April, 2004	143	3.39
Ev_{21}	December, 1-5, 2003	117	5.35
Ev_{22}	November, 15-18, 2003	95	3.80
Ev_{23}	October, 1-2, 2003	80	3.23

Table 3: Thiessen polygon areal weights.

Rain gauges	Area (km^2)	Ratio
Saint Roman	142	27.10 %
Mialet	96	18.32 %
Barre des Cévennes	90	17.18 %
Soudorgues	86	16.41 %
Saumane	62	11.83 %
Anduze	48	9.16 %
Total	524	100%

The dual form of the optimization problem (3) can be solved by quadratic programming. The resulting SVR model can be expressed as a linear combination of kernel functions that depend on the *support vectors*, i.e. on the training examples that lie outside the ε-tube after completion of training:

$$g(x,\boldsymbol{\alpha},b)=\sum_{k=1}^{N_s}\alpha_k\kappa\left(x,x(k)\right)+b \qquad (4)$$

where $\boldsymbol{\alpha}$ and b are the vector of parameters and the bias, $x(k)$ is the k-th support vector, N_S is the number of support vectors, and κ is the kernel function. In this paper, we choose a Gaussian kernel, defined as

$$\kappa(u,v)=\exp\left(-\left\|u-v\right\|^2/\sigma^2\right)$$

where σ is the width of a Gaussian function centered at the training examples.

The model complexity and the generalization ability of the SVR model depend mainly on three hyper-parameters ε, C and σ.

5. CLUSTERING THE EVENTS

The proposed approach consists in clustering the events $\{Ev_i,\ i = 1 \text{ to } 17\}$ of the training database, each cluster containing events whose models have a similar behavior. This approach differs from the hydrological point of view, which considers usually the similarity between events based on the number of peaks, the duration of the event, the water level rise time and drop time, the maximum level, etc.

To cluster the events, the first step consists in defining a similarity measure between each pair of events (Ev_i, Ev_j): two events Ev_i and Ev_j are said to have a similar behavior if model $M_h(i)$ trained on event Ev_i alone can predict satisfactorily event Ev_j, and if model $M_h(j)$ trained on event Ev_j can predict satisfactorily event Ev_i.

The ability of model $M_h(i)$ to predict event Ev_j is assessed by the value of the usual root mean squared error, denoted by s_{ij}: the smaller the value of s_{ij}, the better the prediction of event j by model $M_h(i)$.

Figure 1 shows the 17x17 non-symmetric similarity matrix S whose elements are the s_{ij} obtained from the models of the 17 events of the training database.

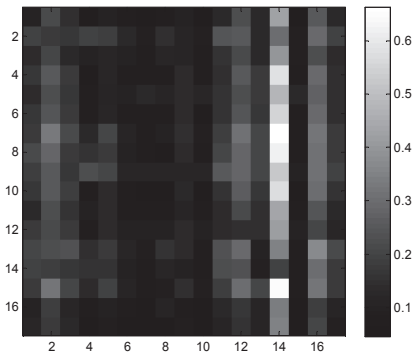

Figure 1: Similarity matrix with h = 1 (30mn).

The models $M_h(i)$ are obtained by applying support vector regression on the data of each event. The tuning parameter ε is set to 10 cm, which is the order of magnitude of the water level measurement uncertainty. Since training is performed on a single event, cross-validation cannot be used for finding the hyperparameters of the model; therefore, the

width σ of the Gaussian kernel was set to 0.5, based on the distribution of data in the regression space.

The regularization parameter C was set to 10^5. For each prediction horizon $h \in \{1, 2, 4, 6, 8\}$ (in units of 30mn), a similarity matrix S was computed.

An agglomerative hierarchical clustering [16] algorithm merged sequentially (i.e. concatenated the time series of) events and event sub-clusters into larger and larger clusters, thereby providing data views at different levels of aggregation. The hierarchical clustering procedure is shown in the diagram of figure 2. Initially, each event was its own singleton group, and a model was trained from each event. The two most similar groups i and j were defined as:

$$i, j = \arg\min_{k,l}\left(\max\left(s_{kl}, s_{lk}\right)\right). \quad (5)$$

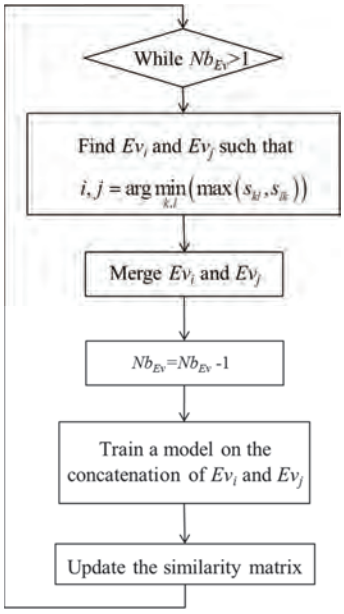

Figure 2: The agglomerative hierarchical clustering procedure. (Nb_{Ev} is the number of events)

Then, the two most similar models were merged, the corresponding time series were concatenated, a new model was trained from the new time series, and the similarity matrix S was updated. This operation was repeated until all the events were merged into a single cluster.

Based on the dendrograms thus obtained (see figure 3), different clusters of events were found for each prediction horizon, as shown on Table 4.

The clusters that contain events 4, 7, 10, 11, 13 and 15 are almost unchanged for the five horizons of prediction. All these events are not intense. The other clusters change with the evolution of the prediction horizon and do not have an obvious hydrological interpretation. Singleton clusters contain mainly the intense invents 2, 12, 14 and 16. In the following, only the non-singleton groups were used for the design of specific models.

The clusters that contain events 4, 7, 10, 11, 13 and 15 are almost unchanged for the five horizons of prediction. All these events are not intense. The other clusters change with the evolution of the prediction horizon and do not have an obvious hydrological interpretation. Singleton clusters contain mainly the intense invents 2, 12, 14 and 16. In the following, only the non-singleton groups were used for the design of specific models.

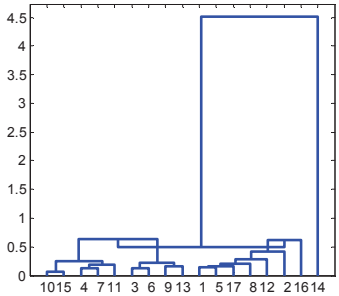

Figure 3: Dendrogram of the agglomerative hierarchical clustering h = 1 (30mn).

Table 4: Clusters obtained for different prediction horizons

30mn	1h	2h	3h	4h
4, 7, 10, 11, 15	4, 7, 10, 13, 15	4, 7 ,10 ,13, 15	4, 7, 10, 11, 13, 15	4, 7, 10, 11, 13, 15
3, 6, 9, 13	3, 6	1, 2, 3, 5, 6, 9	1, 3, 5, 6, 8, 9, 17	3, 5, 6, 8, 9, 17
1, 2, 5, 8, 12, 17	8, 12, 17	8, 12, 17	2	1, 2
14	1, 5, 16	11	12	12
16	2	14	14	14
	9	16	16	16
	11			
	14			

6. RESULTS

In this section, we describe the performances of the specific models and of the global models for the prediction horizons of interest (30mn, one to four hours). For each horizon, a specific SVR model is designed from each non-singleton cluster of flood events, and a global SVR model is designed from all flood events. Cross-validation with a grid search method was applied to find the SVR hyperparameters σ and C, the order n_a and the rainfall window length n_b.

The optimal set of SVR hyperparameters and the optimal n_a and n_b are given in table 5. The value of σ obtained for specific models is much smaller than for the global model. This means that, in order to account for the diversity of events, the global model tends to be closer to linear than the specific models. Two quantities were computed in order to assess the quality of the models: the Nash coefficient C_N and the persistence C_p, defined respectively as:

$$C_N = 1 - \frac{\sum_{k=1}^{N}\left(y(k) - g\left(x(k), \boldsymbol{\alpha}, b\right)\right)^2}{\sum_{k=1}^{N}\left(y(k) - \bar{y}\right)^2}$$

where $\bar{y}$ is the mean water level, and

$$C_p = 1 - \frac{\sum_{k=1}^{N}\left(y\left(k+h_p\right) - g\left(x\left(k+h_p\right), \boldsymbol{\alpha}, b\right)\right)^2}{\sum_{k=1}^{N}\left(y\left(k+h_p\right) - y(k)\right)^2}.$$

Table 6: Optimal CVR hyperparameters and optimal n_a and n_b of the global and specific models

		30mn	1h	2h	3h	4h
Global model	n_a	3	4	4	5	5
	n_b	2	4	1	1	1
	σ	50	50	50	50	50
	C	10^5	10^5	$5\ 10^5$	$5\ 10^5$	$5\ 10^5$
Specific model 1	n_a	1	4	2	2	4
	n_b	5	2	2	2	2
	σ	1	1	1	1	1
	C	10^5	10^5	10^5	10^5	10^5
Specific model 2	n_a	3	2	1	5	1
	n_b	3	2	3	4	5
	σ	5	5	1	1	1
	C	10^5	10^5	10^5	10^5	10^5
Specific model 3	n_a	3	4	4	x	5
	n_b	5	5	4	x	2
	σ	1	5	5	x	50
	C	10^5	10^5	10^5	x	10^5
Specific model 4	n_a	x	4	x	x	x
	n_b	x	5	x	x	x
	σ	x	5	x	x	x
	C	x	10^5	x	x	x

The values of C_N are in [−∞, 1], a high value of C_N indicating a good forecasting performance. The persistence $C_p \in [-\infty, 1]$ provides a comparison of the quality of the prediction of the model to the forecast provided by the naïve model $\hat{y}(k+h) = y(k)$. If the persistence is positive, the model forecast is better than the naive forecast. The closer C_N and C_p to 1, the better the forecasting performances.

The specific models and the global model are then applied to the 6 events of the test database (Ev_{18}, Ev_{19}, Ev_{20}, Ev_{21}, Ev_{22}, Ev_{13}). Table 6 shows the Nash and persistence coefficients of the global model and of the best specific model obtained by cross-validation for different prediction horizons.

In all cases except 30mn forecasts, the best specific model generalizes better on the test events than the global model. The performances obtained on event E_{22} are shown graphically on Figure 4. The improvement obtained by the best specific model is substantial, especially on the persistence criterion ($C_p \geq 0.9$ for 3h and 4h-ahead forecasts). The accuracy of the estimated water level peak is much better for predictions made by the best specific model than by the global model. For comparison, we performed the same procedure by clustering the events randomly, which resulted in substantially worse results.

Table 6: Comparison of C_N and C_p for the global model and the best specific model.

	Models	C_N, C_p	30mn	1h	2h	3h	4h
Ev_{18}	Global	C_N	0.99	0.99	0.96	0.92	0.88
		C_p	0.21	0.42	0.54	0.63	0.67
			0.99	0.99	0.97	0.96	0.94
			-0.18	0.40	0.72	0.83	0.83
			1.00	0.99	0.97	0.95	0.92
			0.46	0.55	0.71	0.81	0.85
			1.00	0.99	0.99	0.98	0.87
			0.27	0.70	0.85	0.91	0.76
			1.00	0.98	0.95	0.91	0.86
			0.52	0.43	0.54	0.62	0.65
			0.99	0.99	0.97	0.94	0.92
			0.03	0.65	0.74	0.75	0.79
			1.00	0.99	0.97	0.95	0.91
			0.43	0.31	0.48	0.54	0.56
			0.99	0.99	0.98	0.95	0.92
			-0.34	0.29	0.66	0.60	0.59
			1.00	0.99	0.95	0.94	0.94
			0.56	0.44	0.52	0.68	0.79
			0.99	0.99	0.99	0.98	0.98
			0.25	0.75	0.87	0.90	0.94
			0.99	0.96	0.88	0.80	0.74
			0.29	0.41	0.48	0.55	0.63
			0.98	0.97	0.89	0.88	0.59
			-0.05	0.55	0.52	0.72	0.43

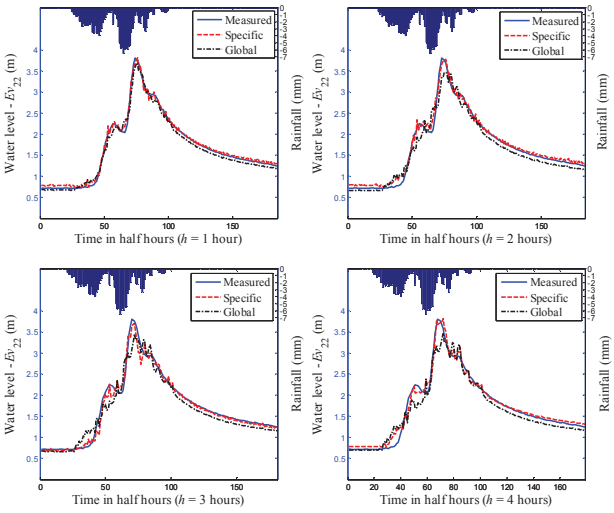

Figure 4: Measured (blue) and estimated (specific model: red dashed; global model: black dashed) water levels for event Ev_{22}; right scale rainfall. 1h, 2h, 3h and 4h horizon forecasts.

7. CONCLUSION

We presented and studied the rainfall-water level modeling in the case of flash floods, in the absence of rainfall predictions, for the Gardon d'Anduze watershed, which is archetypical of flash flood areas. The difficulty of the tasks is twofold: (i) high noise and disturbances, especially during intense events, (ii) unavailability of rainfall predictions, which makes long-term predictions inaccurate. These difficulties were circumvented in part by clustering the flash

floods into different groups such that models of each cluster behave similarly. To that end, an agglomerative hierarchical clustering algorithm was used. For each prediction horizon, a specific SVR predictive model was designed from the events present in each cluster.

The comparisons, in terms of Nash coefficient C_N and of persistence C_p, between the global model derived from all flood events and the specific models showed that in almost all cases one specific model gives better results than the global model: an ensemble of specific models captures more efficiently the underlying mechanisms of the complex flood events than a global model

Our future work will include the selection of the best specific model while forecasting a new event: after inception of a flood, the measured water levels will be compared with the predictions made by all the available specific models, in order to select the specific model (or ensemble of specific models) that is most relevant, thereby providing a decision aid that will be added to the existing tools that are available to the flood forecasting experts.

8. **ACKNOWLEDGMENTS**

The present work is a part of the FLASH project supported by the Agence Nationale de la Recherche in the framework of the "Syscomm" program (ANR-09-SYSC_004). The authors thank the SCHAPI, particularly Caroline Wittwer, Bruno Janet and Arthur Marchandise for providing data and for helpful discussions about flash flood forecasting.

9. REFERENCES

[1] C. Sivapragasam, S. Y. Liong, and M. F. K. Pasha, "Rainfall and runoff forecasting with SSA - SVM approach," *Journal of Hydroinformatics*, vol. 3(3), pp. 141–152, 2001.

[2] C. Sivapragasam and S. Y. Liong, "Flow categorization model for improving forecasting," *Nordic Hydrology*, vol. 36(1), pp. 37–48, 2005.

[3] K. Choy and C. Chan, "Modelling of river discharges and rainfall using radial basis function networks based on support vector regression," *International Journal of Systems Science*, vol. 34(14-15), pp. 763–773, 2003.

[4] X. Yu, S. Y. Liong, and V. Babovic, "EC-SVM approach for realtime hydrologic forecasting," *Journal of Hydroinformatics*, vol. 6(3), pp. 209–223, 2004.

[5] S. Ch, N. Anand, B. Panigrahi, and S. Mathur, "Streamflow forecasting by SVM with quantum behaved particle swarm optimization," *Neurocomputing*, vol. 101, pp. 18–23, 2013.

[6] Y. B. Dibike, S. Velickov, D. Solomatine, and M. Abbott, "Model induction with support vector machines: introduction and applications," *Journal of Computing in Civil Engineering*, vol. 15(3), pp. 208–216, 2001.

[7] S. Y. Liong and C. Sivapragasam, "Flood stage forecasting with support vector machines," *Journal of the American Water Resources Association*, vol. 38(1), pp. 173–196, 2002.

[8] M. Behzada, K. Asgharia, M. Eazia, and M. Palhang, "Generalization performance of support vector machines and neural networks in runoff modeling," *Expert Systems with Applications*, vol. 36(4), p. 7624–7629, 2009.

[9] J. Y. Lin, C. T., Cheng, and K.W. Chau, "Using support vector machines for long-term discharge prediction," *Hydrological Sciences Journal*, vol. 51(4), pp. 599–612, 2006.

[10] P. S. Yu, S. T. Chen, and I. Chang, "Support vector regression for realtime flood stage forecasting," *Journal of Hydrology*, vol. 328(3-5), pp. 704–716, 2006.

[11] L. Ljung, *System identification: theory for the user*. Prentice-Hall, Upper Saddle River, NJ, 1999.

[12] G. Artigue, A. Johannet, V. Borrell, and S. Pistre, "Flash flood forecasting in poorly gauged basins using neural networks: case study of the Gardon de Mialet basin (southern France)," *Nat. Hazards Earth Syst. Sci.*, vol. 12, pp. 3307–3324, 2012.

[13] M. Toukourou, A. Johannet, G. Dreyfus, and P. Ayral, "Rainfall-runoff modeling of flash floods in the absence of rainfall forecasts: the case of Cevenol flash floods," *Applied Intelligence*, vol. 35(2), pp. 178–189, 2011.

[14] A. Thiessen, "Precipitation averages for large areas," *Mon. Weather Rev.*, vol. 39, pp. 1082–1084, 1911

[15] B. Schölkopf and A. Smola, *Learning with Kernels: Support vector machines, regularization, optimization, and beyond*, M. MIT Press, Cambridge, Ed., 2002.

[16] A. Jain and R. Dubes, *Algorithms for Clustering Data*, N. Prentice-Hall, Englewood Cliffs, Ed., 1998.