



## Full Length Research Paper

# Prédiction de la moyenne horaire de la vitesse du vent sur le site de Lomé par réseau de neurones

Adekunlé Akim Salami, Ayité. S. A. Ajavon, Mawugno. K. Kodjo, Koffi-Sa Bedja.

Département de Génie Electrique, Ecole Nationale Supérieure d'Ingénieurs (ENSI) – Université de Lomé - Lomé, Togo

Received December 2016 – Accepted February 2017



\*Corresponding author. E-mail: [akim\\_salami@yahoo.fr](mailto:akim_salami@yahoo.fr)

Author(s) agree that this article remain permanently open access under the terms of the Creative Commons Attribution License 4.0 International License.

## Résumé:

This study presents software with graphical user interfaces, developed in our Laboratory to predict wind speeds. The aim of the study is to predict the mean hourly wind speed of the site of Lomé with a neural network approach. The interesting result obtained in the site of Lomé with a Mean Square Error of prediction  $MSE = 1,293$  and a linear correlation coefficient between the observed and predicted data  $R^2 = 0.71$  shows clearly that it is possible to predict the hourly wind speed with a neural network approach.

**Keyword:** mean hourly wind speed, times series, neural network, prediction.

## Cite this article:

Akim A. Salami, Ayité. S. A. Ajavon, Mawugno. K. Kodjo, Koffi-Sa Bedja (2017). Prédiction de la moyenne horaire de la vitesse du vent sur le site de Lomé par réseau de neurones. Revue Cames – Sci. Appl. & de l'Ing., Vol. 2(1), pp. 01-12. ISSN 2312-8712.

## 1. Introduction

La prédiction de la vitesse du vent est indispensable du fait que la production de l'énergie éolienne liée à la variation de la vitesse du vent doit concurrencer les autres sources d'énergie qui ne sont pas aussi trop variables en terme de puissance active.

Plusieurs modèles de prédictions ont été développés pour la prédiction des données de la vitesse du vent sur un site. Ils incluent des modèles ARIMA de Box-Jenkins, la chaîne de Markov, les filtres de Kalman, les modèles bilinéaires, les modèles à seuil, la logique floue, le neuroflou et les réseaux de neurones artificiels [1], [2], [4], [8]. Cependant parmi les méthodes numériques ne tenant pas compte de connaissances plus approximatives voir subjectives pour la description des phénomènes naturels; seuls les modèles de réseaux de neurones artificiels pour la prévision des données de la vitesse du vent ont démontré de faibles erreurs de prédictions [9].

Pour ces raisons, nous avons développé un outil logiciel permettant de faire la prédiction des vitesses du vent sur un site en utilisant les méthodes d'intelligences artificielles (réseaux de neurones, logique floue et neuroflou). L'utilisation de cet outil pour la prédiction de la moyenne horaire des vitesses du vent sur le site de Lomé par réseau de neurones doit passer d'abord par un prétraitement des données

météorologiques collectées sur ledit site et par une présentation de l'approche neuronale utilisée.

## 2. Problématique

La puissance électrique d'origine éolienne produite par un système de conversion d'énergie cinétique du vent en énergie électrique est liée aux caractéristiques du système de conversion et à la variation temporelle du vent.

Lorsqu'une bonne caractérisation du site d'exploitation de l'énergie électrique d'origine éolienne est faite, le choix optimal des caractéristiques du système de conversion de cette énergie est facile.

Cependant, si les caractéristiques du système de conversion de l'énergie du vent en énergie électrique sont connues, seule une maîtrise des variations du vent est indispensable pour une exploitation optimale de cette énergie.

Le vent est caractérisé par sa direction et sa vitesse qui sont toutes des variables aléatoires, imprévisibles et non linéaires.

Les systèmes de conversion de l'énergie du vent sont dotés des dispositifs pour mesurer la direction du vent et adapter rapidement la direction de l'axe du rotor pour se placer face au vent [5]. La direction du vent n'a donc que fort peu d'influence sur la production

d'électricité, d'autant plus que généralement, les vents ont tendance à avoir une ou deux directions dominantes pour lesquelles la majeure partie de l'énergie est produite.

Ainsi les caractères aléatoire et imprévisible de la puissance électrique d'origine éolienne à la sortie d'un système de conversion d'énergie du vent sont liés à la variation de la vitesse du vent.

Les outils les plus utilisés pour rendre plus prévisible la vitesse du vent sur un site sont les réseaux de neurones, la logique floue, et le neuroflou. Ces trois outils ont été intégrés dans une application que nous avons développée en Matlab pour la prédiction de la vitesse du vent et de la production éolienne sur un site. L'application de ce logiciel pour la prédiction de la moyenne horaire de vitesse du vent sur le site de Lomé par réseau de neurones est-elle possible avec une bonne précision? Si oui, qu'elle serait la structure et les paramètres du modèle de prédiction de la moyenne horaire de vitesse du vent sur ce site. Ainsi pour y parvenir, la présentation des données météorologiques sur le site de Lomé et de l'approche neuronale utilisée s'impose.

### 3. Les données météorologiques

Notre centre d'intérêt à travers cet article est d'élaborer un modèle de prédiction de la moyenne horaire de vitesses du vent sur le site de Lomé à partir des données météorologiques observées sur ce site. Après avoir présenté la source et le prétraitement des données, nous avons fait le choix des variables d'entrée du modèle neuronale de prédiction.

#### 1.- Source des données

Les données météorologiques utilisées dans cette étude sont des données recueillies sur le site de la Société Aéroportuaire Lomé Tokoin au TOGO. Ces données sont enregistrées chaque jour par intervalle d'une heure à une hauteur de 10 m au-dessus du sol et sont obtenues à travers la base de données météorologique du site Web 'http://weather.uwyo.edu/surface/meteogram' [3]. de l'Université de Wyoming (Department of Atmospheric Science). La fiche de données fournie par [3] (Tableau 1), comporte pour chaque enregistrement : le nom de site (STN), la date et l'heure d'enregistrement (TIME), la température ambiante (TMP) en °C, la pression atmosphérique (ALTM) en hPa, la température de la rosée (DEW) en °C, la direction du vent (DIR) en degré, l'humidité relative (RH) en %, la visibilité (VIS) en km, la vitesse du vent (SPD) en m/s etc. Ainsi les données recueillies suivant le format du tableau 1 doit subir un

prétraitement afin d'être le plus commode et le plus fiable à l'exploitation.

#### 2.- Prétraitement des données

Nous avons utilisé le logiciel EXCEL pour effectuer le prétraitement des données météorologiques. La modélisation a été réalisée par l'outil NNTOOL (Neural Network Tool) de MATLAB utilisé par le logiciel MasterEol 1.0 que nous avons développé dans un environnement MATLAB (nous rappelons que le logiciel MasterEol 1.0 est un outil d'aide à la décision pour la caractérisation du potentiel éolien, la prédiction des vitesses du vent et l'optimisation du dimensionnement des aérogénérateurs) [7]. Pour être conforme aux spécificités de MATLAB, les données ont été prétraitées dans EXCEL. Ce prétraitement est d'autant plus nécessaire que, d'une part, certaines données se présentent dans le logiciel sous forme de chaîne de caractères alors qu'il nous faut des données numériques, et d'autre part, il existe des informations se présentant sous des formats non utilisables par MATLAB. Les données prétraitées sont lues par l'application MasterEol qui aidera l'utilisateur à effectuer le choix et la normalisation des paramètres d'entrée-sortie du modèle de prédiction.

#### 3.- Choix et normalisation des paramètres d'entrée-sortie du modèle de prédiction

Notre objectif est de prédire la moyenne horaire de la vitesse du vent en prenant des variables physiques exogènes et endogènes du processus comme les paramètres d'entrée de notre modèle. L'utilisation de toutes les variables d'entrée sera coûteuse en temps de calcul. Toutefois certains paramètres sont dépendants et fortement corrélés entre eux, ainsi il n'est pas nécessaire d'utiliser toutes les variables comme paramètres d'entrée. Dans cet article nous allons utiliser le test de corrélation linéaire entre variables définie par la relation (1).

$$r = \frac{\overline{x.y} - \bar{x}.\bar{y}}{\sqrt{(\overline{x^2} - (\bar{x})^2)(\overline{y^2} - (\bar{y})^2)}} \quad (1)$$

où :  $x$  et  $y$  sont des variables.

Afin d'adapter nos données à la réalisation du modèle neuronal, nous avons procédé à la normalisation de leurs valeurs dans un intervalle [0, 1] par la relation (2).

**Tableau 1.** Présentation d'une fiche de données du site Web [3] constituant la source de données

Observations for LOME TOKOIN MIL, Togo (DXXX)										
1300Z 5 Jul 2004 to 1700Z 5 Jul 2004										
STN	TIME	ALTM	TMP	DEW	RH	DIR	SPD	VIS	CLOUDS	
	DD/HHMM	hPa	°C	°C	%	deg	m/s	km		
DXXX	26/1500	1009.2	30	24	70	200	6	10	FEW020	BKN250
DXXX	26/1400	1010.2	31	24	66	190	7	10	FEW020	BKN250
DXXX	26/1300	1010.8	31	25	70	200	6	10	FEW020	BKN250
DXXX	26/1200	1011.9	31	26	75	200	6	10	SCT020	BKN250
DXXX	26/1100	1012.9	31	25	70	210	5	10	SCT016	BKN250
DXXX	26/1000	1012.9	30	25	75	210	6	10	BKN015	BKN230
DXXX	26/0900	1013.9	29	25	79	210	5	10	BKN013	BKN230
DXXX	26/0830	1012.9	29	26	84	220	4	10	BKN013	BKN230
DXXX	26/0800	1012.9	28	25	84	220	3	8	SCT012	BKN230
DXXX	26/0730	1011.9	27	25	89	0	0	7	FEW010	BKN230
DXXX	26/0630	1011.9	26	23	84	0	0	3	FEW010	BKN050 BKN230 F
DXXX	26/0600	1010.8	26	24	89	0	0	4	FEW010	BKN050 BKN230 F
DXXX	26/0530	1010.8	26	23	84	0	0	6	SCT010	BKN100

$$X_n = \frac{(x_{\max} - X)}{(x_{\max} - x_{\min})} \quad (2)$$

où :  $x_{\max}$  et  $x_{\min}$  représentent respectivement les valeurs maximale et minimale des données représentées sous forme d'un vecteur  $X$ . Les données normalisées sont utilisées pour réaliser un modèle de prédiction suivant l'approche neuronale que nous allons présenter dans la suite de ce travail.

**4. Réseau de neurones**

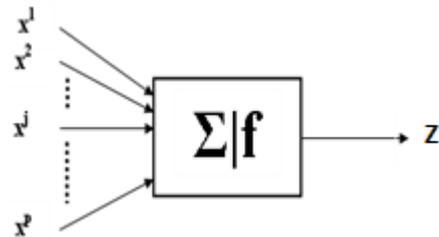
Les réseaux de neurones constituent un courant bien précis de l'intelligence artificielle, dont le modèle de développement est basé sur l'idée de calquer des algorithmes d'apprentissage sur un modèle simplifié du cerveau.

Le neurone formel (figure 1) est un modèle (relation (3)) qui se caractérise par un état interne  $s \in S$ , des signaux d'entrée  $x^1, \dots, x^p$  et une fonction  $f$  de transition (linéaire ou logistique) d'état [6] :

$$z = h(x^1, \dots, x^p) = f\left(\beta_0 + \sum_{j=1}^p \beta_j x^j\right) \quad (3)$$

$\beta_0$  étant appelé le biais du neurone. Cette combinaison affine est déterminée par un vecteur de poids  $[\beta_1, \dots, \beta_p]$  associé à chaque neurone et dont

les valeurs sont estimées dans la phase d'apprentissage. Ils constituent "la mémoire" ou "connaissance répartie" du réseau.



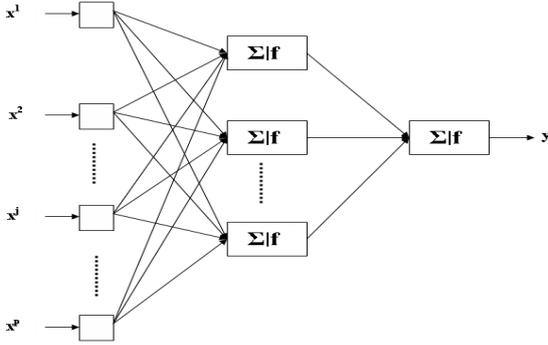
**Figure 1.** Représentation d'un neurone formel

Le réseau de neurones utilisé dans ce travail pour la prédiction des vitesses du vent sur le site de Lomé est un perceptron multicouche avec un algorithme d'apprentissage de rétropropagation du gradient, car ce modèle est adapté à des problèmes de prédiction [6], [9].

Le perceptron multicouche (PMC) est un ensemble de neurones interconnectés [6]. Il est caractérisé par son architecture et le calcul optimal de ses différents paramètres de liaison se fait par apprentissage.

En matière d'architecture on peut dire que : une couche est un ensemble de neurones n'ayant pas de connexion entre eux ; une couche d'entrée lit les signaux entrants, un neurone par entrée  $x^j$  ; une couche en sortie fournit la réponse du système. Selon

les bibliographies, la couche d'entrée qui n'introduit aucune modification n'est pas comptabilisée. Une ou plusieurs couches cachées participent au transfert. Un neurone d'une couche cachée est connecté en entrée à chacun des neurones de la couche précédente et en sortie à chaque neurone de la couche suivante. La figure 2 nous montre un exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie.



**Figure 2.** Exemple de perceptron multicouche élémentaire avec une couche cachée et une couche de sortie

Un perceptron multicouche réalise donc une transformation suivant la relation (4)

$$y = \phi(x^1, \dots, x^p; \beta) \quad (4)$$

où :  $\beta$  est le vecteur contenant chacun des paramètres  $\beta_{jkl}$  de la  $j^{\text{ième}}$  entrée neurone de la  $\ell^{\text{ième}}$  couche ; la couche d'entrée ( $\ell = 0$ ) n'est pas paramétrée, elle ne fait que distribuer les entrées sur tous les neurones de la couche suivante.

Les entrées d'un réseau sont notées  $x^1, \dots, x^p$  comme les variables explicatives d'un modèle tandis que les poids des entrées sont des paramètres  $\beta$  à estimer lors de la procédure d'apprentissage et que la sortie est la variable à expliquer ou cible du modèle.

S'agissant de l'apprentissage, si nous supposons que l'on dispose d'une base d'apprentissage de taille  $n$  d'observations  $(x_i^1, \dots, x_i^p; y_i)$  des variables explicatives  $X^1, \dots, X^p$  et de la variable à prévoir  $Y$ , l'apprentissage est l'estimation  $\hat{\beta}$  des paramètres du modèle solutions du problème des moindres carrés (Equivalent à une maximisation de la vraisemblance dans le cas gaussien) (relation (5)) :

$$\hat{\beta} = \arg \min_b Q(b) \quad (5)$$

avec :

$$Q(b) = \frac{1}{n} \sum_{i=1}^n [y_i - \phi(x_i^1, \dots, x_i^p; (b))]^2 \quad (6)$$

L'algorithme d'optimisation le plus utilisé est celui de rétropropagation du gradient basé sur l'idée suivante : en tout point  $b$ , le vecteur gradient de  $Q$  pointe dans la direction de l'erreur croissante. Pour faire décroître  $Q$  il suffit donc de se déplacer en sens contraire. Il s'agit d'un algorithme itératif modifiant les poids de chaque neurone selon la relation (7) [5], [6], [9].

$$b_{jkl}(i) = b_{jkl}(i-1) + \Delta b_{jkl}(i) \quad (7)$$

où : la correction  $\Delta b_{jkl}(i)$  exprimée par la relation (8) est proportionnelle au gradient et à l'erreur attribuée à l'entrée concernée  $\varepsilon_{jkl}(i)$  et incorpore un terme d'inertie  $\alpha b_{jkl}(i-1)$  permettant d'amortir les oscillations du système.

$$\Delta b_{jkl}(i) = -\tau \varepsilon_{jkl}(i) \frac{\partial Q}{\partial b_{jkl}} + \alpha b_{jkl}(i-1) \quad (8)$$

Le coefficient de proportionnalité  $\tau$  est appelé le taux d'apprentissage. Il peut être fixe à déterminer par l'utilisateur ou encore varier en cours d'exécution selon certaines règles paramétrées par l'utilisateur. Il paraît en effet intuitivement raisonnable que, grand au début pour aller plus vite, ce taux décroisse pour aboutir à un réglage plus fin au fur et à mesure que le système s'approche d'une solution. La formule de rétropropagation de l'erreur fournit, à partir des erreurs observées sur les sorties, l'expression de l'erreur attribuée à chaque entrée de la couche de sortie à la couche d'entrée.

La littérature sur le sujet propose des quantités de recettes destinées à améliorer la vitesse de convergence de l'algorithme ou bien lui éviter de rester collé à une solution locale défavorable. Des propriétés (dynamique markovienne ergodique et convergence vers la mesure stationnaire) de cet algorithme impliquent une convergence presque sûre ; la probabilité d'atteindre une précision fixée a priori tend vers 1 lorsque la taille de l'échantillon d'apprentissage tend vers l'infini.

Une amélioration importante consiste à introduire un terme de pénalisation ou de régularisation dans le critère à optimiser. Celui-ci devient alors :

$$\hat{\beta} = \arg \min_b Q(b) + \delta \|b\|^2 \quad (9)$$

Le paramètre  $\delta$  doit être fixé par l'utilisateur ; plus il est important et moins les paramètres ou poids peuvent prendre des valeurs "chaotiques" contribuant ainsi à limiter les risques de surapprentissage.

### 5. Application des réseaux de neurones pour la prédiction de moyenne horaire de vitesse du vent sur le site de Lomé

Nous avons appliqué la méthode citée en utilisant les séries de données horaires collectées sur le site de Lomé.

Pour des raisons pratiques à la méthode de prédiction, nous avons séparé les données en deux groupes. Les premiers 80% des données servent de données d'apprentissage pour la classification et l'estimation des paramètres du modèle, et les derniers 20% des données servent de données de validation afin de tester le modèle [1], [4], [8], [9].

L'algorithme d'apprentissage du réseau de neurones utilisé dans cet article, la rétropropagation du gradient est mis en œuvre par l'outil NNTOOL de la boîte à outil Neural Network Toolbox de MATLAB. La fenêtre de l'outil NNTOOL est appelée par le logiciel (MasterEol 1.0) que nous avons réalisé dans un environnement MATLAB pour nos travaux antérieurs [7].

Le logiciel MasterEol 1.0 dont la structure et l'organigramme fonctionnel sont représentées respectivement par la Figure 3 et la Figure 4 comprend cinq (05) modules :

- Base de données ;
- potentiel éolien ;
- traitement du signal ;
- génération artificielle de données ;
- prédiction.

Le logiciel MasterEol accède aux fichiers de données, analyse les données, calcule le potentiel éolien d'un site, élabore des modèles de génération artificielle de données et des modèles de prédictions de vitesses du vent et de la production éolienne sur un site.

Le module de prédiction de MasterEol permet de faire la prédiction des vitesses du vent sur un site en utilisant les méthodes d'intelligences artificielles (réseaux de neurones, logique floue et neurofloue) qui sont adaptés aux problèmes de la prédiction éolienne. Ces méthodes présentent des avantages considérables par rapport aux méthodes physiques et statistiques traditionnellement utilisées dans ce but.

Le module prédiction dont l'organigramme fonctionnel est illustré sur la figure 5 nous permet de faire appel au programme NNTOOL (du logiciel MATLAB) qui ouvre la fenêtre Network/Data Manager qui nous permettra d'importer, de créer, d'utiliser et d'exporter de modèle de réseaux de

neurones et données. Les données exportées sont traitées et sont affichées dans une fenêtre Prédiction (de MasterEol) sous forme de courbes et mesures d'apprentissage et de validation.

Un clic sur le bouton « Create » de la fenêtre Network/Data Manager permet d'accéder à la fenêtre Create New Network. Dans cette fenêtre nous avons choisi le type d'architecture du réseau qui est le Feed Forward backpropagation, le nombre de couche qui est de deux (02) et les fonctions de transitions sigmoïdes pour les neurones de la couche cachée et la fonction de transition linéaire pour le neurone de la couche de sortie (Figure 8).

Après l'apprentissage du réseau, l'onglet « Weights » de la fenêtre « Network » permet de visualiser les paramètres  $\hat{\beta}$  (voir §4) constituant les poids et biais du modèle neuronal (Figure 9, Figure 10, Figure 11 et Figure 12).

#### 1.- Choix des variables d'entrées

Dans le but d'avoir un modèle parcimonieux nous avons effectué une analyse des coefficients de corrélation entre les variables d'entrées ( $x^1, \dots, x^p$ ) que sont la température ambiante (TMP) en °C, la pression atmosphérique (ALTM) en hPa, la température de la rosée (DEW) en °C, la direction du vent (DIR) en degré, l'humidité relative (RH) en %, la visibilité (VIS) en km, la vitesse du vent (SPD) en m/s.

Le tableau 2 nous montre les coefficients de corrélation inter-variables calculés par la relation (1).

**Tableau 2.** Présentation des coefficients de corrélation inter-variables

Variables	SPD	ALTM	TMP	DEW	RH
SPD	1	-0.013	0.285	-0.03	-0.38
ALTM		1	-0.359	-0.25	0.210
TMP			1	0.332	-0.51
DEW				1	0.213
RH					1

Pour constituer les entrées du modèle de réseau de neurones, on ne retiendra, d'une part, qu'une seule variable dans un groupe de variables très corrélées. Ainsi l'analyse du tableau 2 ne nous permet d'éliminer que l'humidité relative du vent car étant bien corrélée à la température. D'autre part, les diverses séances d'apprentissage du modèle menées sur différentes combinaisons de variables d'entrée, grâce à la comparaison des diverses MSE (Mean Square Error c'est à dire Erreur Quadratique Moyenne), ont permis d'écarter des variables d'entrée moins influençant que d'autres.

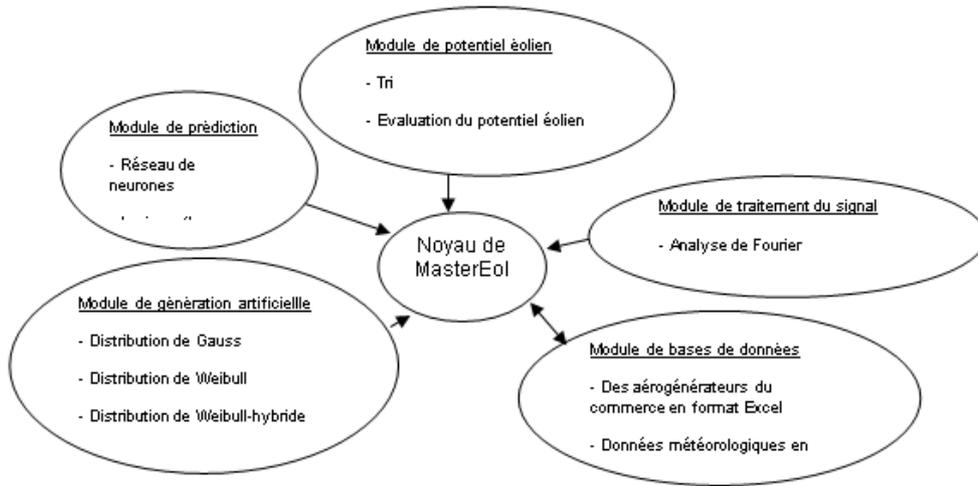


Figure 3. Schéma de la structure du logiciel MasterEol

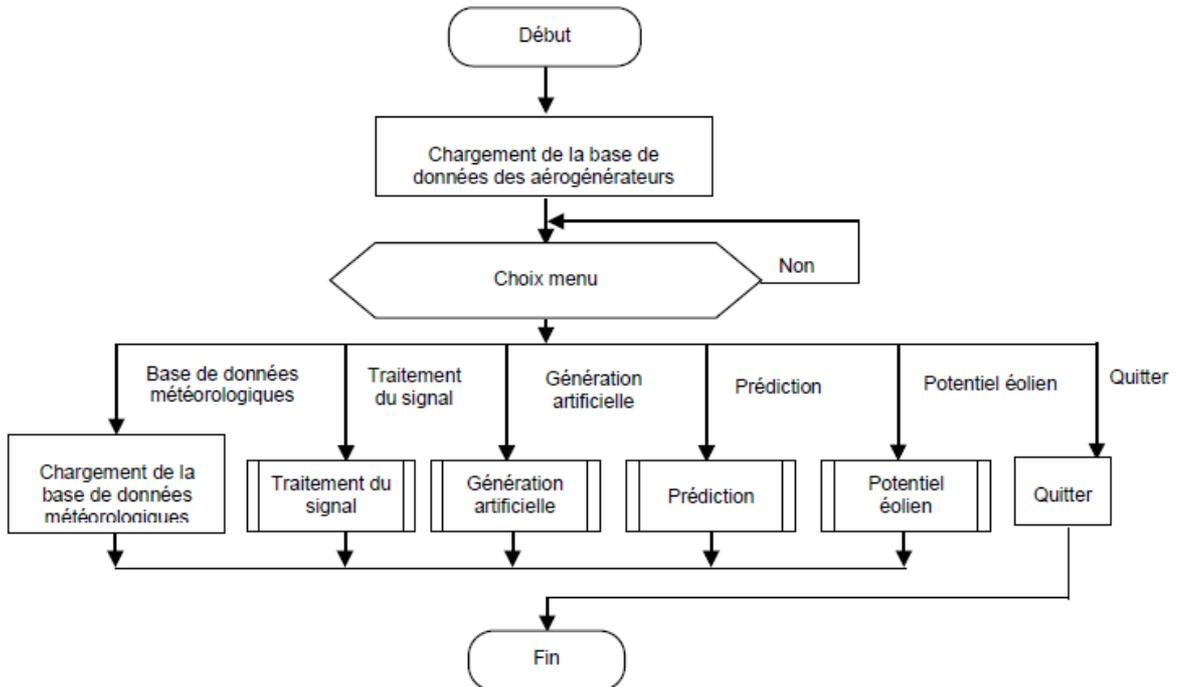


Figure 4. Organigramme principal du logiciel MasterEol

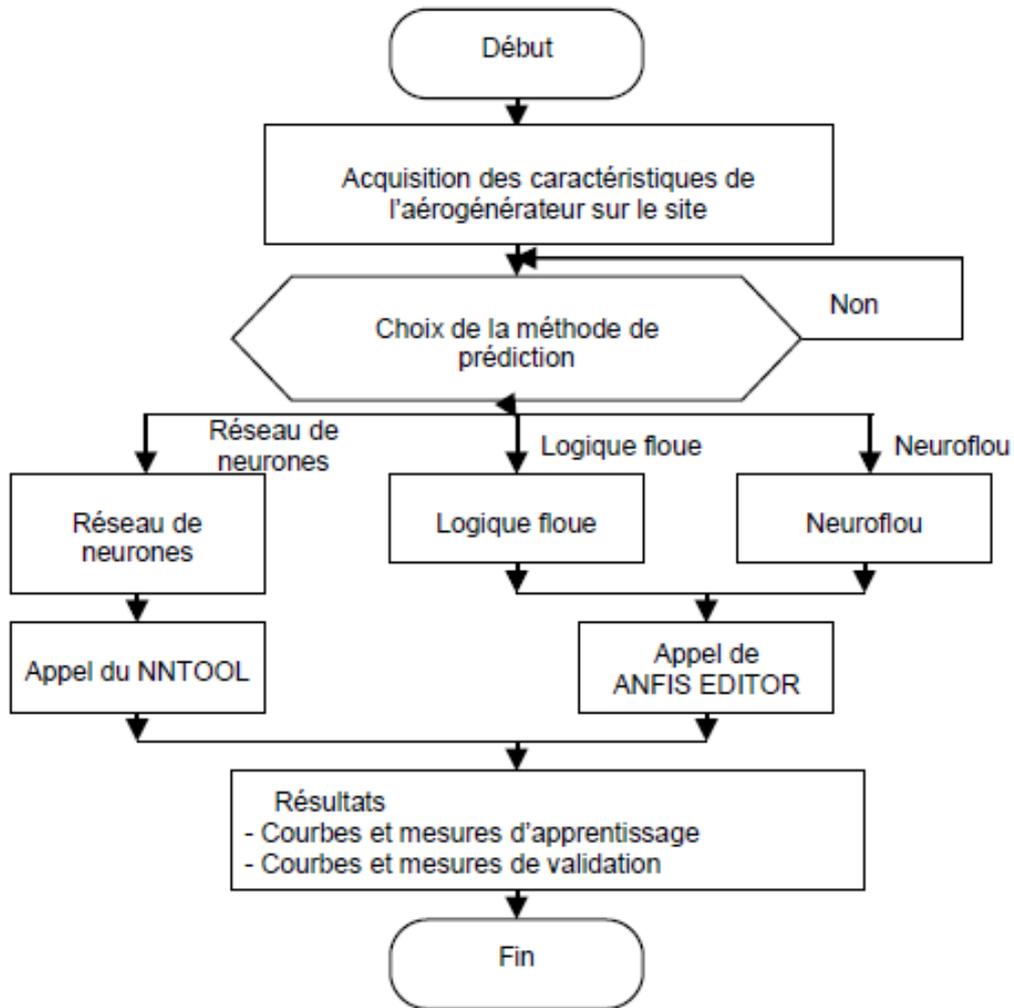


Figure 5. Organigramme du module prédiction du logiciel MasterEol

En définitif, les variables d'entrées du modèle de réseau de neurones sont constituées par la vitesse du vent ( $spd(t-1)$ ), la température ( $tmp(t-1)$ ), et la pression ( $altm(t-1)$ ), le tout à l'instant  $t-1$ , et par la vitesse ( $spd(t-2)$ ) à l'instant  $t-2$  ; soit au total quatre variables d'entrées [8]. La sortie du modèle est bien évidemment la vitesse du vent

( $spd(t)$ ) à l'instant courant  $t$ . La figure 6 nous montre la représentation temporelle de la variable de sortie (la vitesse du vent à l'instant courant  $t$ ) et la figure 7 nous illustre la représentation temporelle de chaque variable d'entrée (la vitesse du vent, la température, et la pression, le tout à l'instant  $t-1$ , et par la vitesse à l'instant  $t-2$ ).

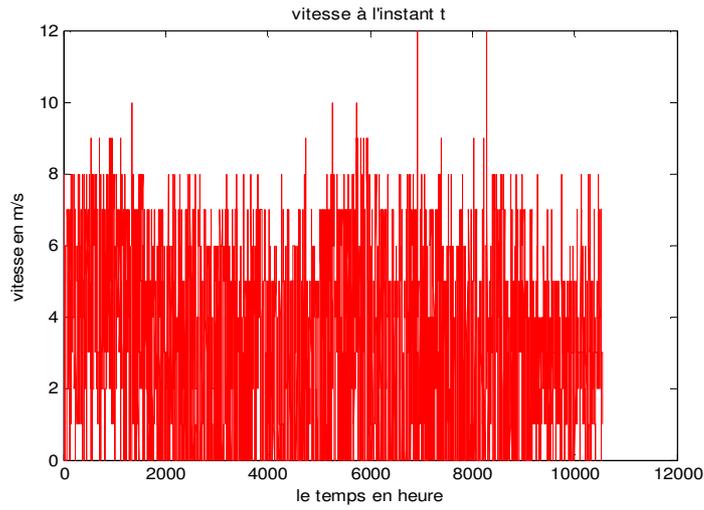


Figure 6. Courbe de la variable de sortie du modèle

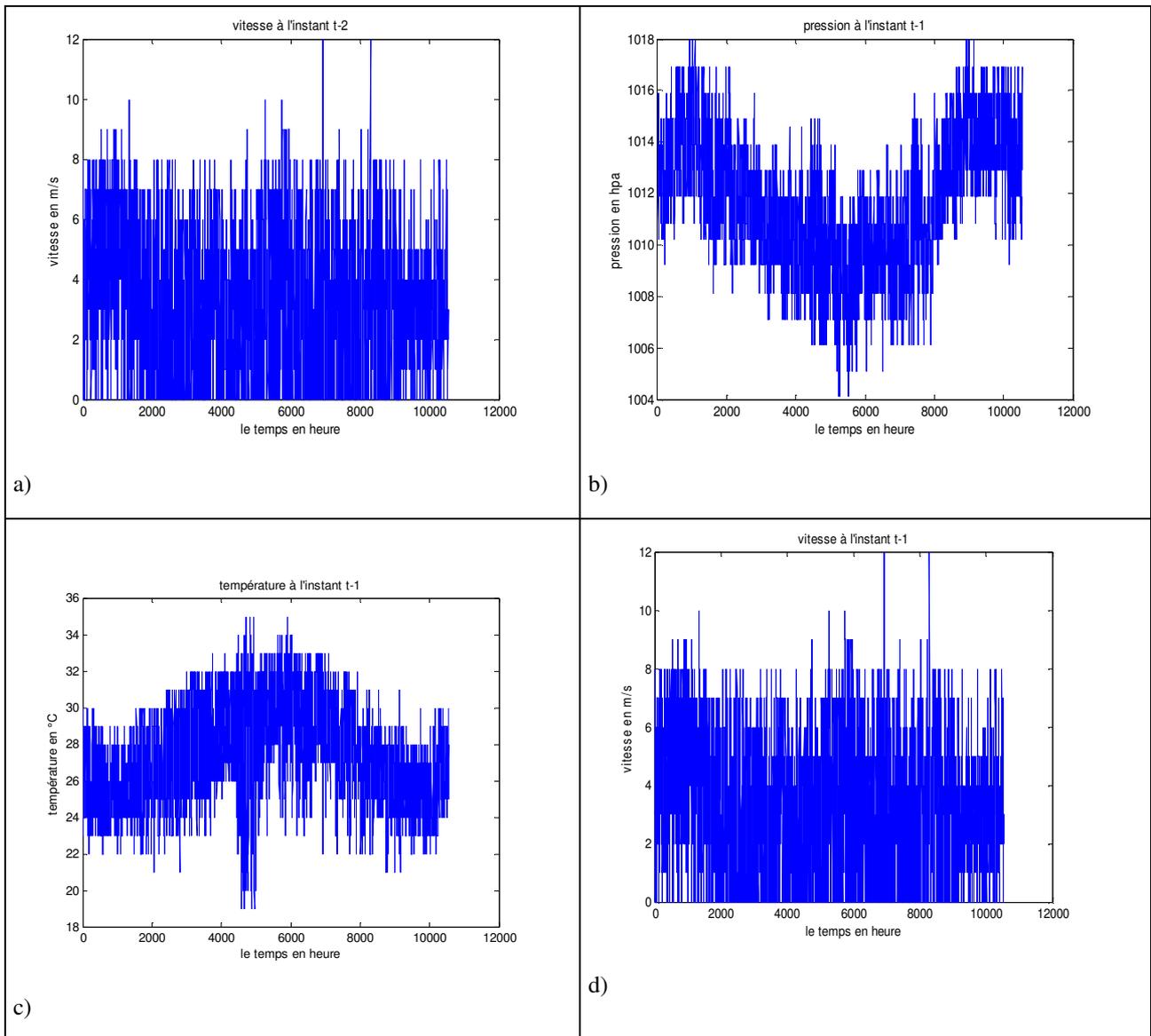


Figure 7. Courbes des variables d'entrée du modèle

## 2.- Modèle de réseau de neurones pour la prédiction de la moyenne horaire de la vitesse du vent

Pour la prédiction des vitesses du vent sur le site de Lomé, nous avons opté pour un modèle de perceptron multicouche avec un algorithme d'apprentissage de rétropropagation du gradient, car ce modèle est adapté à des problèmes de prédiction [5], [6].

Afin d'avoir un modèle parcimonieux, nous avons limité le nombre de couches à deux (une couche cachée et une couche de sortie). La couche cachée utilise une fonction sigmoïde comme fonction d'activation, alors que la couche de sortie utilise une fonction linéaire.

Cette configuration a été utilisée dans [9] et a fourni de bonne performance pour la prédiction des vitesses du vent.

Notre objectif est de prédire la moyenne horaire des vitesses du vent dans un futur proche (1 heure), pour cela la couche de sortie est composée d'un seul neurone linéaire. Dans le but de déterminer le nombre de neurones dans la couche cachée, nous avons essayé plusieurs séries d'apprentissage en variant le nombre de neurones de cette couche de un (01) à vingt (20). A partir des résultats obtenus, nous avons constaté qu'une couche cachée avec dix (10) neurones présente de bonne performance en terme de nombre de paramètres réduits par rapport aux autres.

L'architecture du modèle de perceptron multicouche adoptée pour la prédiction des vitesses du vent sur le site de Lomé est représentée sur la figure 8. Les valeurs des paramètres  $\hat{\beta}$  constituant la matrice des poids et biais de ce modèle sont représentées sur les figures 9, 10, 11 et 12.

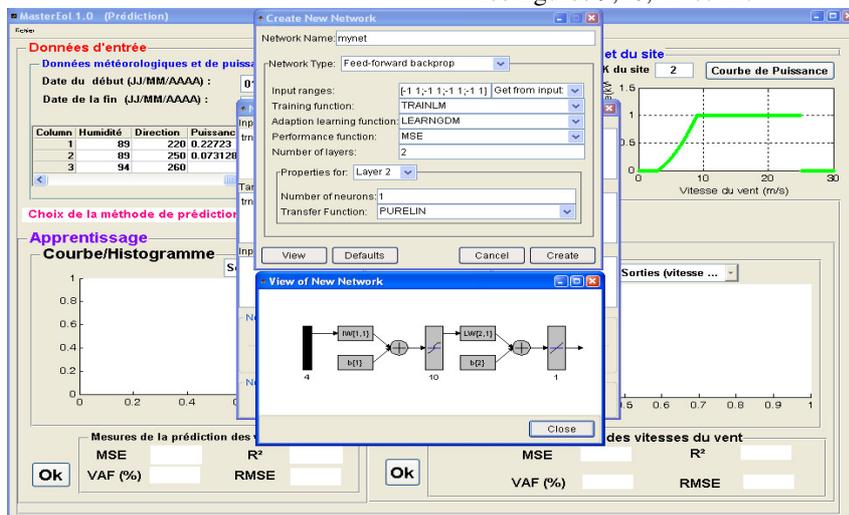


Figure 8. Architecture du modèle dans la fenêtre View of New Network de NNTOOL de MATLAB

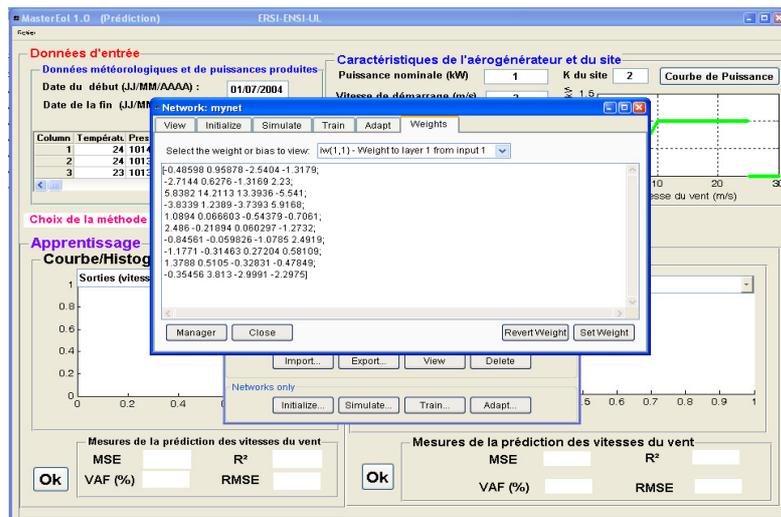


Figure 9. Valeurs des poids de la première couche du modèle dans la fenêtre Network de NNTOOL de MATLAB

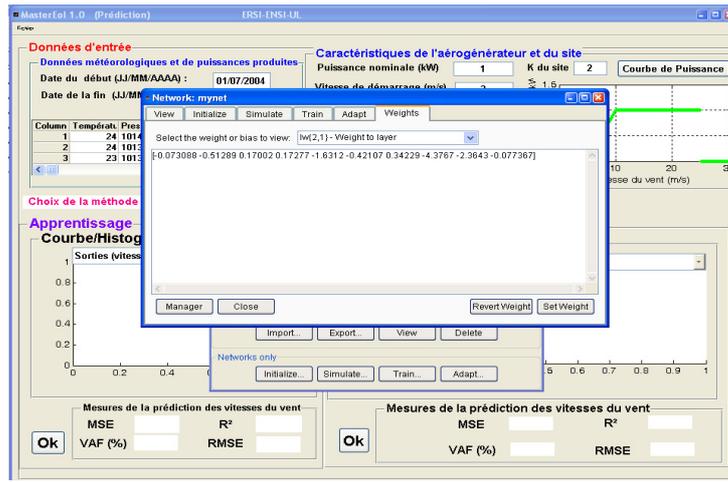


Figure 10. Valeurs des poids de la deuxième couche du modèle dans la fenêtre Network de NNTOOL de MATLAB

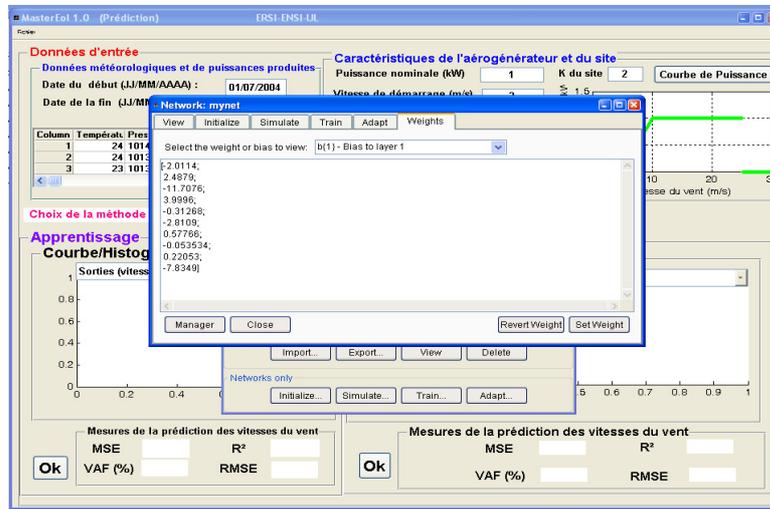


Figure 11. Valeurs des biais de la première couche du modèle dans la fenêtre Network de NNTOOL de MATLAB

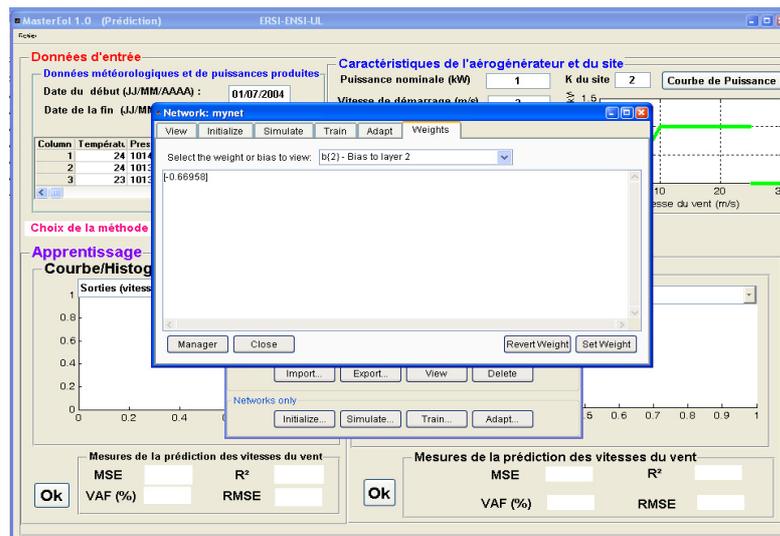


Figure 12. Valeurs des biais de la deuxième couche du modèle dans la fenêtre Network de NNTOOL de MATLAB

### 6. Analyse des résultats

Le modèle de réseau de neurones pour la prédiction de la moyenne horaire des vitesses du vent sur le site de Lomé nous montre les écrans des résultats la fenêtre Prédiction de MasterEol (voir Figures 13 et 14) :

- une Erreur Quadratique Moyenne (MSE) d'une valeur de 2,019 au cours de l'apprentissage du modèle qui est réduite à 1,0293 au cours de la validation du modèle ;
- une Variance Account For (VAF) d'une valeur de 62,6312% au cours de l'apprentissage du modèle

qui est réduite à 61,2352% au cours de la validation du modèle ;

- un coefficient de corrélation linéaire ( $R^2$ ) d'une valeur de 0,7255 au cours de l'apprentissage du modèle qui est réduite à 0,7164 au cours de la validation du modèle ;
- les histogrammes (figure 13) et les autocorrélogrammes (figure 14) des erreurs au cours de l'apprentissage et de la validation du modèle nous montrent que les distributions de ces erreurs sont gaussiennes et que plus de 60% de données sont prédites avec une erreur nulle lors de la validation.

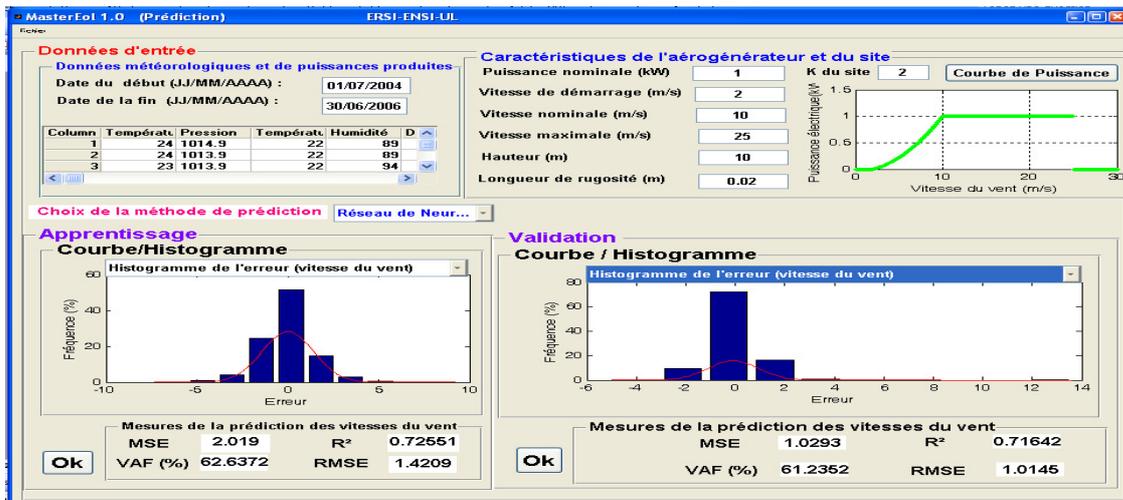


Figure 13. Histogrammes des erreurs de prédiction de la moyenne horaire des vitesses du vent par le réseau de neurones

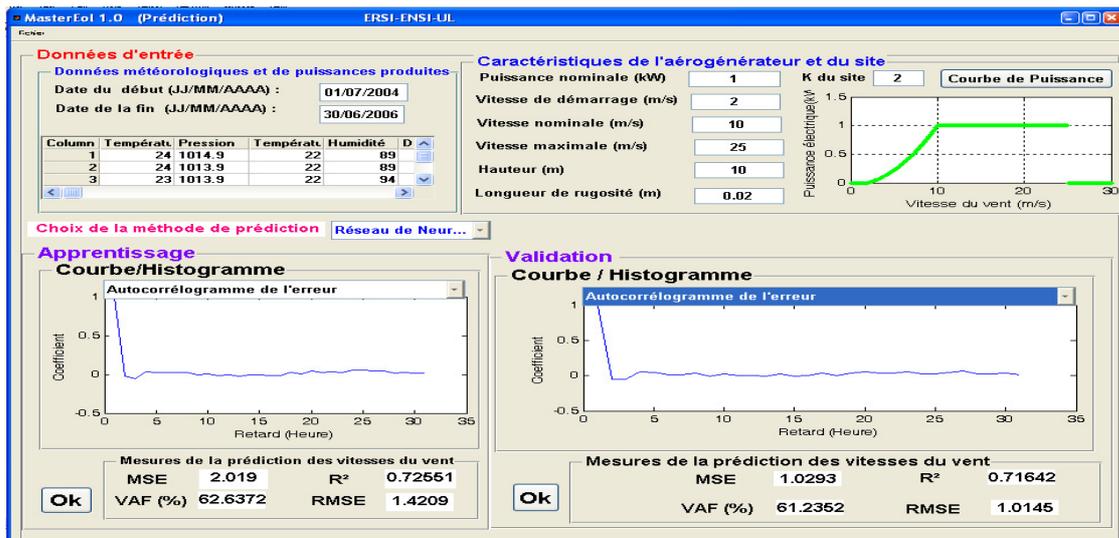


Figure 14. Autocorrélogrammes des erreurs de prédiction de la moyenne horaire des vitesses du vent par le réseau de neurones

A partir des résultats obtenus, nous pouvons dire que le modèle de prédiction de la moyenne horaire des vitesses du vent sur le site de Lomé par le réseau de neurones présente de bonnes performances, cela se remarque surtout par une diminution de l'Erreur Quadratique Moyenne (MSE) lors de la validation du modèle. Ainsi ce modèle de réseau de neurones dont les paramètres (biais et poids) sont déterminés (Figures 9, 10, 11 et 12) peut être utilisé pour prédire de façon fiable la moyenne horaire des vitesses du vent sur le site de Lomé au Togo.

## 7. Conclusion

La prédiction de la moyenne horaire de la vitesse du vent par un réseau de neurones sur le site de Lomé a été étudiée dans cet article avec un outil très convivial développé en Matlab. Le modèle neuronal conçu est un perceptron multicouche avec une couche

cachée dont la fonction d'activation est une fonction sigmoïde, alors que la couche de sortie utilise une fonction linéaire. Ce modèle comporte quatre variables d'entrée (la température à l'instant  $t-1$ , la pression à l'instant  $t-1$ , les vitesses à l'instant  $t-1$  et à l'instant  $t-2$ ) et une variable de sortie (la vitesse à l'instant  $t$ ). L'utilisation des réseaux de neurones pour la prédiction de la moyenne horaire de la vitesse du vent sur le site de Lomé nous a permis d'obtenir un bon résultat de prédiction. Puisque les distributions des erreurs sont des gaussiennes de moyennes nulles et le modèle présente un coefficient de corrélation linéaire ( $R^2$ ) entre données observées et données prédites supérieur à 0,7.

La puissance éolienne étant proportionnelle au cube de la vitesse du vent, ce modèle peut être utilisé pour prédire la puissance éolienne sur le site de Lomé.

floue, des réseaux de neurones et des algorithmes génétiques, thèse de doctorat, Université Henri Poincaré, Nancy-I, **1999**.

## REFERENCES

- [1] Berge E., Experiences with wind forecasting techniques in Norway. Proceedings of the First IEA Joint action symposium on wind forecasting Techniques Norrköping, 59-64, **2002**.
- [2] Bossanyi E., Short-term wind prediction using Kalman filters. Wind Energy, 9: 1-8, **1985**.
- [3] 'http://weather.uwyo.edu/surface/meteogram'.
- [4] Kamal L. Jafri Y., Times series models to simulate and forecast hourly averages wind speed in Quetta, Pakistan, Solar Energy, 61:23-32, **1997**.
- [5] Kodjo M. K., Bédja K., Ajavon A. S. A., Faye R. M., Lishou C., Neural network for predictive control of the mechanism of orientation of wind turbine, Journal des Sciences Pour l'Ingénieur (JSPI), Presses Universitaires de Dakar, N° 9, pp. 75-85, **2008**.
- [6] Lofti B., Contribution à la commande de la machine asynchrone, utilisation de la logique
- [7] Salami. A. A., Modélisation du vent à partir des méthodes d'analyse de données imparfaites : étude et réalisation d'un outil d'aide à la décision pour la caractérisation du potentiel éolien, la prédiction de vitesses du vent et l'optimisation du dimensionnement des aérogénérateurs au TOGO, thèse de doctorat, Université de Lomé, Lomé, **2008**.
- [8] Salami A. A., Tiem S., Lishou C., Bédja K., La logique floue pour la prédiction de la moyenne horaire de la vitesse du vent sur le site de Lomé, Journal des Sciences Pour l'Ingénieur (JSPI), Presses Universitaires de Dakar, N° 8, pp. 88-96, **2007**.
- [9] Sfetsos A., A novel approach for the forecasting of mean hourly wind speed time series, Renew Energy, 27: 163-174, **2002**.