

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ferhat Abbas / Sétif 1
Faculté des Sciences Économiques,
Commerciales et des Sciences de Gestion
Département :



جامعة فرحات عباس / سطيف 1
كلية العلوم الاقتصادية، التجارية وعلوم التسيير
القسم:

Thèse présentée pour l'obtention du diplôme de
doctorat en sciences de gestion
Spécialité : sciences économiques

Titre:

Prévision des Indices des Prix à la Consommation
par les Réseaux de Neurones : le cas algérien

Encadrant :

Dr Khassef Djamel eddine

Par :

Bourioune Tahar

رئيسا	جامعة سطيف 1	أستاذ محاضر (أ)	د. بن عطاءالله عادل
مشرفا و مقررا	جامعة سطيف 1	أستاذ محاضر (أ)	د. خاسف جمال الدين
مناقشا	جامعة سطيف 1	أستاذ محاضر (أ)	د. مرابط ساعد
مناقشا	جامعة البويرة	أستاذ	أ.د. حيدوشي عاشور
مناقشا	جامعة الجزائر 3	أستاذ	أ.د. جنيدي مراد
مناقشا	جامعة الشلف	أستاذ	أ.د. البشير عبد الكريم

DEDICACES

Je dédie ce modeste travail à mes défunts parents et à toute ma famille.

REMERCIEMENTS

Ma reconnaissance va envers Monsieur Khassef Djamel-eddine, Maître de conférence A, à l'université de Sétif, pour avoir d'abord accepté de remplacer mon ex encadrant parti en Arabie Séoudite, Daignez accepter, cher maître, ma reconnaissance.

Ma profonde gratitude va également à :

Monsieur Benattalah Adel, maître de conférences A, à l'université de Sétif, d'avoir consenti de présider le jury. Pour cela, je vous exprime ma considération.

Monsieur Merabet Saad, maître de conférences A, à l'université de Sétif, pour l'honneur que vous me faites d'avoir accepté d'évaluer mon travail.

Monsieur Hidouchi Achour, Professeur à l'université de Bouira, pour la considération que vous m'apportez d'avoir accepté d'être examinateur de ma thèse.

Monsieur Djenidi Mourad, Professeur à l'université d'Alger³, pour l'égard que vous me faites d'avoir consenti d'évaluer mon travail.

Monsieur El bachir Abelkrim, Professeur à l'université de Chlef, pour l'estime que vous me témoignez d'avoir accepté d'évaluer mon travail.

SOMMAIRE

SOMMAIRE.....	I
LISTE DES FIGURES	IV
LISTE DES TABLEAUX	VI
ACRONYMES	VIII
INTRODUCTION GENERALE	1
REVUE DE LITTERATURE.....	7
CH1 PREVISION DES IPC PAR LES METHODES TRADITIONNELLES.....	9
1.1 LES MODELES STRUCTURELS	9
1.2 LES MODELES STATISTIQUES	9
1.2.1 <i>Les modèles de Box-Jenkins et de Sims</i>	9
1.2.1.1 Le modèle SARIMA (p, d, q) (s, P, D, Q).....	9
1.2.1.2 LE MODELE VAR (p).....	11
1.2.2 <i>Prévision des IPC par le modèle SARIMA (p,d ,q)(s,P,D,Q)</i>	13
1.2.2.1 DONNEES	13
1.2.2.2 Stratégie de prévision	13
1.2.2.3 Spécification du modèle et prévision.....	13
1.2.2.3.1 Identification du modèle SARIMA (p,d,q) (s,P,D,Q).....	13
1.2.2.3.2 Estimation	20
1.2.2.3.3 Validation.....	23
1.2.2.3.4 prevision.....	25
CH2 PREVISION DES IPC PAR LE MODELE FFNN	28
2.1 LES MODELES ANN.....	28
2.2 LE MODELE FFNN	30
2.2.1 <i>Structure</i>	30
2.2.2 <i>L'algorithme BP</i>	31
2.2.2.1 Procédure d'apprentissage des poids w_{kj}	31
2.2.2.2 Procédure d'apprentissage des poids w_{ji}	32
2.2.3 <i>Prévision</i>	34
2.2.3.1 DONNEES	34
2.2.3.2 architecture du réseau.....	34
2.2.3.3 Méthodologie de prévision	34
2.2.3.4 Spécification et résultats.....	34
2.2.3.4.1 PREVISION PAR LE MODELE FFNN1.....	36
2.2.3.4.2 Le modèle FFNN2	37
2.2.3.4.3 Le modèle FFNN3	40
2.2.4 <i>Performance modèle FFNN vs SARIMA</i>	41
CH3 PREVISION DES IPC PAR LE MODELE GRNN	43
3.1 LES MODELES NEURONAUX STATISTIQUES.....	43
3.2 LE MODELE GRNN	45
3.3 STRUCTURE	47
3.4 PREVISION	48
3.4.1 <i>DONNEES</i>	48
3.4.2 <i>Structure du réseau</i>	48
3.4.3 <i>Méthodologie de prévision</i>	48
3.4.4 <i>Spécification et résultats</i>	48
3.4.4.1 choix de la variante optimale.....	48
3.4.4.2 prédiction par le modèle GRNN	49

3.4.4.3 PREVISION	49
3.5 PERFORMANCE COMPAREE DU MODELE GRNN	50
CH4 PREVISION DES IPC PAR LE MODELE RBFN	52
4.1 LE MODELE RBFN	52
4.2 REGLES D'APPRENTISSAGE	53
4.2.1 PSEUDO-INVERSE TECHNIQUE	54
4.2.2 GRADIENT DESCENT LEARNING	55
4.2.3 HYBRID LEARNING	56
4.2.3.1 CHOIX DES ' C_j ' PAR L'ALGORITHME « K-MEANS »	56
4.2.3.2 CHOIX DES ' Σ_j ' PAR L'ALGORITHME « K-NEAREST NEIGHBOURS »	57
4.2.3.3 CHOIX DES W^*_{kj} PAR L'ALGORITHME « STEEPEST DESCENT »	57
4.3. PREVISION	57
4.3.1 Données	57
4.3.2 architecture	57
4.3.3 Méthodologie de prévision	58
4.3.4 SPECIFICATION ET RESULTATS	58
4.3.4.1 CHOIX DE LA VARIANTE OPTIMALE	58
4.3.4.2 PREDICTION PAR LE MODELE RBFN	59
4.3.4.3 PREVISION	59
4.4 PERFORMANCE DU MODELE RBFN	60
CH5 PREVISION DES IPC PAR LE MODELE TDNN	62
5.1 LES MODELES DYNAMIQUES	62
5.2 LE MODELE TDNN	62
5.2.1 STRUCTURE TDNN	62
5.2.2 PREVISION	
5.2.2.1 DONNEES	64
5.2.2.2 ARCHITECTURE DU RESEAU	64
5.2.2.3 Méthodologie de prévision	64
5.2.2.4 Spécification et résultats	64
5.2.2.5 Performance relative du modèle TDNN	67
5.3 LE MODELE NNAR	68
5.3.1 Structure	68
5.3.2 ARCHITECTURE DU RESEAU	69
5.3.3 Prévision	69
5.3.3.1 Prédiction	69
5.3.3.2 Prévision	70
5.3.4 Performance relative du modèle NNAR	70
CH6 PREVISION DES IPC PAR LE MODELE NNARMAX	72
6.1 LES MODELES RECURRENTS RNN	72
6.2 LE MODELE NNARMAX (BECCALI, 2008)	73
6.2.1 Structure	73
6.2.2 L'algorithme d'entraînement BPTT	75
6.3 PREVISION	76
6.3.1 Données	76
6.3.2 Architecture du modèle	76
6.3.3 Méthodologie de prévision	77
6.3.4 Spécification et résultats	78
6.3.4.1 Prédiction	78
6.3.4.2 Prévision	79
6.4 PERFORMANCE COMPAREE DU MODELE NNARMA	80

CH7 PREVISION DES IPC PAR LE MODELE FL	81
7.1 THEORIE DES ENSEMBLES FLOUS	81
7.1.1 <i>sous ensemble flou</i>	81
7.1.2 <i>OPERATEURS LOGIQUES</i>	81
7.1.3 <i>INFERENCE FLOUE</i>	82
7.2 STRUCTURE	83
7.3 PREVISION	84
7.3.1 <i>Données</i>	84
7.3.2 <i>Architecture du système</i>	84
7.3.3 <i>Méthodologie de prévision</i>	85
7.3.4 <i>Spécification et Résultats</i>	85
7.3.4.1 <i>LE MODELE FL</i>	85
7.3.4.2 <i>Prédiction et Prévision</i>	87
7.3.5 <i>Performance comparée</i>	87
CONCLUSION.....	89
BIBLIOGRAPHIE.....	92
ANNEXES	100

TABLE DES ILLUSTRATIONS

Liste des figures

FIGURE 1. 1 REPRESENTATION GRAPHIQUE DE YT	14
FIGURE 1. 2 REPRESENTATION GRAPHIQUE D' YTSA.....	14
FIGURE 1. 3 REPRESENTATION D'YTSA PAR LE MODELE SARIMA	25
FIGURE 1. 4 PREVISION D'YTSA PAR LE MODELE SARIMA.....	26
FIGURE 1. 5 PREVISION DE YT PAR LE MODELE SARIMA	26
FIGURE 2. 1 UN NEURONE FORMEL.....	28
FIGURE 2. 2 LE PERCEPTRON	29
FIGURE 2. 3 MLP [2:2:1].....	29
FIGURE 2. 4 STRUCTURE FFNN [N:M:1].....	30
FIGURE 2. 5 L'ALGORITHME BP.....	33
FIGURE 2. 6 AJUSTEMENT DE YTSA PAR FFNN1.....	36
FIGURE 2. 7 PREVISION DE YT PAR FFNN1	37
FIGURE 2. 8 AJUSTEMENT DE YTSA PAR FFNN2	38
FIGURE 2. 9 PREVISION DE YT PAR FFNN2	39
FIGURE 2. 10 AJUSTEMENT DE YTSA PAR FFNN3	40
FIGURE 2. 11 PREVISION DE YT PAR FFNN3	41
FIGURE 2. 12PREVISIONS COMPAREES SARIMA, FFNN1	42
FIGURE 3. 1 STRUCTURE GRNN [K:M:2:1].....	47
FIGURE 3. 2 AJUSTEMENT DE YTSA PAR GRNN	49
FIGURE 3. 3 PREVISION DE YT PAR GRNN	50
FIGURE 3. 4 PREVISIONS COMPAREES SARIMA, FFNN1,GRNN.....	51
FIGURE 4. 1 STRUCTURE RBFN [K :M:1].....	52
FIGURE 4. 2 AJUSTEMENT DE YTSA PAR RBFN	59
FIGURE 4. 3 PREVISION DE YT PAR RBFN	60
FIGURE 4. 4 PREVISIONS COMPAREES SARIMA, FFNN1, GRNN, RBFN.....	61
FIGURE 5. 1 STRUCTURE TDNN.....	63
FIGURE 5. 3 AJUSTEMENT D'YT PAR TDNN	66
FIGURE 5. 4 PREVISION DE YT PAR TDNN	66
FIGURE 5. 5 PREVISIONS COMPAREES SARIMA, FFNN1, GRNN, RBFN,TDNN	67
FIGURE 5. 6 STRUCTURE DU NNARX	68
FIGURE 5. 7 ARCHITECTURES DU MODELE NNARX...	68
FIGURE 5. 8 'SERIES PARALLEL ARCHITECTURE' DU NNAR.....	69
FIGURE 5. 9 AJUSTEMENT D'YTSA PAR NNAR	69
FIGURE 5. 10 PREVISION DE YT PAR NNAR.....	70
FIGURE 5. 11 PREVISIONS COMPAREES SARIMA, FFNN1,GRNN,RBFN,TDNN&NNAR.....	71
FIGURE 6. 1 RESEAU D'ELMAN (ENN)	72
FIGURE 6. 2 RESEAU DE JORDAN (JNN).....	72
FIGURE 6. 3 LE MODELE NNARMAX(2,2,2,1)	74
FIGURE 6. 4 EXEMPLE DU RESEAU RECURRENT	75
FIGURE 6. 5 RESEAU RECURRENT DEPLIE	76
FIGURE 6. 6 1-STEP AHEAD PREDICTION SOURCE: MATLAB 2011	78
FIGURE 6. 7 3-STEP AHEAD PREDICTION SOURCE: MATLAB 2011.....	78

FIGURE 6.8 12-STEP AHEAD PREDICTION SOURCE: MATLAB 2011.....	79
FIGURE 6.9 PREVISION D' YT PAR NNARMA	79
FIGURE 6.10 PREVISIONS COMPAREES SARIMA,FFNN1,GRNN,RBFN,TDNN,NNARMA	80
FIGURE 7.1 FUZZY INFERENCE SYSTEME (MAMDANI).....	83
FIGURE 7.2 STRUCTURE DU MODELE FL	84
FIGURE 7.3 ARCHITECTURE DU SYSTEME.....	85
FIGURE 7.4 REPRESENTATION DES VARIABLES FLOUES.....	86
FIGURE 7.5 PREDICITION D'YTSA PAR FL	87
FIGURE 7.6 PREVISION DE YT PAR FL.....	87
FIGURE 7.7 PREVISIONS COMPARES DES MODELES.....	88

Liste des tableaux

TABLEAU 1. 1 TEST ANOVA DE Y _T	14
TABLEAU 1. 2 TEST ANOVA D'YTSA.....	14
TABLEAU 1. 3 TEST DF DU MODELE 3 SOURCE : EVIEWS.....	15
TABLEAU 1. 4 TEST DF DU MODELE 2 SOURCE : EVIEWS.....	16
TABLEAU 1. 5 REGRESSION DU MODELE 2 SOURCE : EVIEWS	17
TABLEAU 1. 6 CORRELOGRAMME DU MODELE 2 SOURCE : EVIEWS	17
TABLEAU 1. 7 CHOIX DU RETARD OPTIMAL SOURCE : EVIEWS	18
TABLEAU 1. 8 TEST ADF DU MODELE 3 SOURCE : EVIEWS	18
TABLEAU 1. 9 TEST ADF DU MODELE 2 SOURCE : EVIEWS	19
TABLEAU 1. 10 REGRESSION DU MODELE 2 SOURCE : EVIEWS	20
TABLEAU 1. 11 REGRESSION DU MODELE 1 SARIMA SOURCE : EVIEWS	21
TABLEAU 1. 12 REGRESSION DU MODELE 2 SARIMA SOURCE : EVIEWS	21
TABLEAU 1. 13 REGRESSION DU MODELE 3 SARIMA SOURCE : EVIEWS	22
TABLEAU 1. 14 CRITERES DU POUVOIR PREDICTIF D'YTSA	22
TABLEAU 1. 15 TESTS DE STATIONNARITE ET D'INVERSIBILITE DES RACINES SOURCE EVIEWS	23
TABLEAU 1. 16 TEST SUR LES ESTIMATIONS (COEFFICIENTS) SOURCE : EVIEWS	23
TABLEAU 1. 17 TEST DE NON CORRELATION SERIELLE DES RESIDUS E _t (L-B) SOURCE : EVIEWS	23
TABLEAU 1. 18 CHOW BREAKPOINT TEST 2009:04 SOURCE : EVIEWS	24
TABLEAU 1. 19 TEST DE NORMALITE DES E _t SOURCE : EVIEWS	24
TABLEAU 1. 20 TEST D'HOMOSCEDASTICITE ARCH TEST SOURCE : EVIEWS	24
TABLEAU 1. 21 TEST DE NULLITE DE LA MOYENNE DES E _t SOURCE : EVIEWS	24
TABLEAU 1. 22 PREVISION D'YTSA PAR LE MODELE SARIMA.....	25
TABLEAU 1. 23 PREVISION DE Y _T PAR SARIMA SOURCE : EVIEWS	26
TABLEAU 1. 24 CORRELOGRAMME DE Y _T SOURCE : EVIEWS.....	100
TABLEAU 1. 25 CORRELOGRAMME D' Y _T SA SOURCE : EVIEWS.....	101
TABLEAU 1. 26 CORRELOGRAMME DU MODELE 1 SARIMA SOURCE : EVIEWS	102
TABLEAU 1. 27 CORRELOGRAMME DU MODELE 2 SARIMA SOURCE : EVIEWS	103
TABLEAU 1. 28 CORRELOGRAMME DU MODELE 3 SARIMA SOURCE : EVIEWS.....	104
TABLEAU 1. 29 PERFORMANCE DU MODELE SARIMA SOURCE : EVIEWS	104
TABLEAU 2. 1 PERFORMANCE DES VARIANTES DU MODELE FFNN1 SOURCE MATLAB.....	35
TABLEAU 2. 2 PARAMETRES D'ENTRAINEMENT DU MODELE FFNN1 SOURCE MATLAB.....	35
TABLEAU 2. 3 IPC PREVISIONNELS SELON FFNN1	36
TABLEAU 2. 4 PERFORMANCE DES VARIANTES DU MODELE FFNN2 SOURCE MATLAB.....	38
TABLEAU 2. 5 IPC PREVISIONNELS SELON FFNN2	39
TABLEAU 2. 6 PERFORMANCE DES VARIANTES DU MODELE FFNN3 SOURCE MATLAB.....	40
TABLEAU 2. 7 IPC PREVISIONNELS SELON FFNN3	41
TABLEAU 2. 8 PERFORMANCE COMPAREE DU MODELE FFNN	42
TABLEAU 3. 1 PERFORMANCE DES VARIANTES DU MODELE GRNN SOURCE MATLAB.....	49
TABLEAU 3. 2 IPC PREVISIONNELS SELON GRNN	49
TABLEAU 3. 3 PERFORMANCE COMPAREE DE GRNN	50
TABLEAU 4. 1 PERFORMANCE DES VARIANTES DU MODELE RBFN SOURCE MATLAB	58
TABLEAU 4. 2 IPC PREVISIONNELS PAR RBFN	59
TABLEAU 4. 3 PERFORMANCE COMPAREE DU RBFN	60
TABLEAU 5. 1 PERFORMANCE DES VARIANTES DU MODELE TDNN SOURCE MATLAB.....	65
TABLEAU 5. 2 PARAMETRES D'ENTRAINEMENT DU MODELE TDNN SOURCE MATLAB	65

TABLEAU 5.3 IPC PREVISIONNELS PAR TDNN	66
TABLEAU 5.4 PERFORMANCE COMPAREE DU MODELE TDNN	67
TABLEAU 5.5 IPC PREVISIONNELS PAR NNAR	70
TABLEAU 5.6 PERFORMANCE COMPAREE DU MODELE NNAR	70
TABLEAU 6.1 PARAMETRES D'ENTRAINEMENT DU MODELE NNARMA SOURCE MATLAB	77
TABLEAU 6.2 IPC PREVISIONNELS PAR NNARMA.....	79
TABLEAU 6.3 PERFORMANCE COMPAREE DU MODELE NNARMA	80
TABLEAU 7.1 PERFORMANCE COMPAREE DU MODELE FLOU SOURCE: MATLAB 2011A	88

Acronymes

AC	Autocorrelation Function
ADF	Augmented Dickey Fuller test
AIC	Akaike Information criteria
ANN	Artificial Neural Network
AR(I)MA	Autoregressive (Integrated) Moving Average
BL	BiLinear Model
BP	Back Propagation algorithm
BPTT	Back Propagation Through Time
BVAR	Bayesienne vector Autoregressive model
CENSUS X-11	CENSUS Method II seasonal adjustment method
DF	Dickey Fuller test
EGARCH	Exponential GARCH
ES	Exponential Smoothing model
FARIMA	Fractional ARIMA
FDP	Fonction de Densité de Probabilité
FFNN	Feed Forward Neural Network
FNN	Fuzzy Neural Network
(G)ARCH	(Generalized) Autoregressive conditional heteroskedastic
GDBP	Gradient Descent Back Propagation
GRNN	General Regression Neural Network
IPC	Indice des Prix à la Consommation
K-MEANS	K-MEANS algorithm clustering
K_NN	K-Nearest Neighbors classification algorithm
KRLSR	Kernel Recursive Least Square Regression
LM	Levenberg Marquardt algorithm
LMS	Least Mean of Squares
LSTAR	Logistic Smooth Transition Autoregression
MA	Moving Average
MAE	Mean Average Error
MAPE	Mean Average Percentage Error
MAR	Mixture Autoregressive
MBL	Mixture Bilinear
MLR	Multiple Linear Regression
MLP	Multiple Linear Perceptron
MS-AR	Markov Switching AutoRegressive

MSE	Mean Square Error
NLR	Non Linear Regression
NNAR	Neural Network Autoregressive
NNARMAX	Neural Network Autoregressive Mean Average with exogenous variable X
NNFIR	Neural Network Finite Impulse Response
NNOE	Neural Network Output Error
PACF	Partial Autocorrelation function
PNN	Probability Neural Network
PS	Processus Stochastique
Q-stat	statistique de Ljung-Box
R	coefficient de corrélation de Pearson
RBFN	Radial Basis Function Network
RMSE	Root Mean SquaresErrors
RMSNN	Recursive Multi-Step Neural Network
RNN	Recurrent Neural Network
RNA	Réseau de Neurones Artificiels
RTRL	Real TimeRecurrent Learning
SARIMA	Seasonal Autoregressive Integrated Moving Average
(S)ARIMAX	(S)ARIMAwith exogenous variable X
SC	Schwarz Criteria
SE	Simultaneous Equations
SETAR	Self Exciting Treshold Autoregressive
SLR	Simple Linear Regression
SSE	Sum of Squared Errors
Tanh(tansig)	Hyperbolic Tangent sigmoid function
TAR	Threshold Autoregressive
TARCH	Threshold Autoregressive conditional heteroskedastic
TDL	Time Delay Layer
TDNN	Time Delay Neural Network
TRAINGDM	Training function based on Gradient Descent with Momentum
TRAINGDX	Training function based on GDM and adaptive learning rate
VAR	Vector Autoregressive
VARMA	VectorAutoRegressive Moving Average
VMA	Vector Mean Average
VECM	Vector Error Correction Model

INTRODUCTION GENERALE

L'indice des prix à la consommation IPC sert à mesurer l'inflation, source de déstructuration socio-économique du pays. Une prévision efficace de l'IPC relatif au taux d'inflation (cité dans ce travail, pour la commodité, par IPC) permettrait alors de se prémunir des effets néfastes de cette dernière en élaborant des politiques monétaires convenables.

Les conséquences de mauvaises prévisions peuvent être néfastes. Au niveau micro économique celles-ci peuvent mener à la dissolution de l'entreprise. Au niveau macro économique, elles peuvent conduire à l'échec des politiques économiques.

Par conséquent, en vue de réaliser des prévisions congrues, les techniques de prévision se sont ingénérées au fil de l'histoire à 'mimer' les chroniques économiques et financières réelles.

Dans la quête de techniques de prévision de plus en plus performantes, les séries chronologiques ont connu trois approches : traditionnelle, dynamique et non paramétrique.

L'approche traditionnelle (Yafee, 2000), à court terme, suppose que la série statistique (Y_t) est le produit de causes déterministes superposées mais indépendantes les unes des autres (trend T_t , cycle C_t et saison S_t) et un résidu aléatoire ε_t considéré comme une erreur d'observation selon le modèle général $Y_t = f(t, \varepsilon_t)$ et les modèles classiques d'ajustement par décomposition: $Y_t = F_t + S_t + \varepsilon_t$, $Y_t = F_t \cdot S_t \cdot \varepsilon_t$, $Y_t = F_t \cdot S_t \cdot (1 + \varepsilon_t), \dots$ / $F_t = T_t + C_t$

A long terme, cette approche, utilisée surtout en modélisation structurelle, suppose que la série Y_t est déterminée par diverses variables explicatives X_{it} selon le modèle général $Y = f(X, \Theta)$ / f : fonction linéaire ou non; Θ : paramètres

Les méthodes naïve, des moyennes mobiles (MA,...), du lissage exponentiel (ES,...), de décomposition (Census X-11,...), de régression (modèle linéaire simple SLR, linéaire multiple MLR, non linéaire NLR),... constituent l'essentiel des techniques de lissage et de prévision de ces modèles.

Cette évolution des modèles déterministes s'explique par la prise en charge progressive des diverses caractéristiques statistiques des séries chronologiques réelles (cycle, trend, saisonnalité)

L'approche dynamique des séries chronologiques (Prankratz, 1991), préconisée par BOX-JENKINS (1976) puis SIMS (1980), est née de la déception de ces techniques traditionnelles de prévision construites à partir de l'extrapolation des composantes déterministes et des modèles structurels d'inspiration keynésienne.

Selon cette école, une série chronologique y_1, \dots, y_t ne serait que la réalisation d'un processus stochastique Y_1, \dots, Y_t . Nul besoin d'un modèle explicatif ; le passé de la variable étudiée est l'unique variable explicative.

Le modèle autorégressif moyenne mobile ARMA (p,q) est leur modèle de base. C'est un modèle linéaire stationnaire de représentation générale $\Phi(L) Y_t = \theta(L) \varepsilon_t$. Le modèle SARIMA est son extension aux modèles non stationnaires saisonniers.

Les modèles statistiques (Madsen, 2008) de base SARIMA (p ;d ;q)(P ;D ;Q) , (S)ARIMAX, VARMA(p,q) (vecteur autorégressif moyenne mobile), ... ont été proposés pour remplacer leurs homologues structurels: SLR, MLR, SE (système d'équations simultanées), ... (Wooldridge, 2009)

Ces modèles, à leur tour, se sont efforcés de 'singer' les séries réelles économiques, financières, ... par la prise en considération de caractéristiques statistiques complémentaires (volatilité, asymétrie, changement de régime, multimodalité,...). Ceci a donné naissance à la classe des modèles à erreurs conditionnellement hétéroscédastiques (ARCH, GARCH, EGARCH, TARCH,...), bilinéaires BL et modèles à régime TAR, SETAR,... (Engle, 1982).

Néanmoins, l'hypothèse fondamentale de linéarité des modèles statistiques de base aurait été une contrainte rigide qui s'est traduite en pratique par leur faible performance dans le traitement des séries économiques de nature souvent non linéaire (Refenes, 1994). Le recours aux modèles paramétriques non linéaires NLR n'a pas été d'un grand secours en raison du manque d'une théorie de spécification de ces modèles ; le prévisionniste se trouvant souvent embarrassé devant l'impossibilité de tester une infinité de spécifications (Eyden, 1996). Les modèles TARCH, BL, TAR, ... n'ont pas été, à leur tour, d'un grand concours car, de part leur nature, ils ne peuvent modéliser les variétés de non linéarité autres que celles pour lesquelles ils ont été développés (Khashei, 2009).

L'approche non paramétrique du traitement des séries chronologiques (Graupe, 2007), basée sur les réseaux de neurones artificiels ANN et préluée par Rumelhart (1986) à travers le

Perceptron Multicouches MLP, a baptisé une nouvelle génération de modèles. Relevant plutôt des techniques de traitement des données, ces derniers semblent combler les lacunes des modèles statistiques. Ce sont des modèles adaptatifs, exploratoires, non paramétriques, non linéaires, capables d'approximer n'importe quelle fonction continue et de prévoir un futur output (Yegnanarayana, 2004).

Les aboutissements des travaux de recherche entrepris dans le cadre de la prévision financière (Moshiri, 1997) et récemment macro-économique semblent conforter progressivement l'hypothèse d'une supériorité des modèles de prévision neuronaux sur les modèles traditionnels du point de vue performance. *Est-ce bien le cas ? Les modèles a-théoriques neuronaux sont-ils plus performants en prévision que les modèles traditionnels statistiques ?*

Cependant, ces modèles dénommés à raison « modèles à boîtes noires » sont entâchés de beaucoup d'insuffisances. Le manque d'un fondement théorique, l'incapacité à traiter les variables qualitatives et les *données non fiables* constituent parmi leurs principaux désavantages. Les modèles de logique floue FL basés sur la théorie des ensembles flous et la théorie des possibilités (Zadeh, 1965) *semblent* surmonter les limites des modèles ANN. Ils prennent en considération les données linguistiques (assez fort, fort, très fort) et sont ainsi proches du langage humain. Ils permettent d'exprimer les connaissances qu'on a a priori sur un système sous forme de règles explicites. Actuellement, ils sont expérimentés dans beaucoup de domaines et *semblent* être relativement compétitifs. *Est-ce bien le cas ? Les récents modèles FL sont-ils plus performants que les modèles ANN ?*

En résumé, lequel des modèles (SARIMAX, ANN ou FL) serait le plus performant pour la modélisation / prévision des IPC¹ de l'Algérie?

Dans une tentative de modélisation de l'inflation en Algérie, notre travail consiste à étudier, du point de vue fiabilité, les modèles neuronaux ANN relativement aux modèles traditionnels statistiques SARIMA puis aux récents modèles de logique floue FL. La précellence des modèles neuronaux sur les modèles statistiques et flous de point de vue efficacité constitue notre hypothèse fondamentale à vérifier empiriquement. La prééminence de la variante neuronale « statistique » forme notre hypothèse secondaire. Les critères retenus pour la

¹ L'IPC relatif (mensuel) équivaut au taux d'inflation (mensuel)

performance des prévisions sont la racine de la moyenne des carrés des erreurs RMSE, la moyenne des valeurs absolues des erreurs MAE et le coefficient de détermination R^2 .

Méthodologiquement², nous procéderons par une étude comparative entre les modèles ANN, SARIMA et FL sur la base de la performance de leurs prévisions. Pour l'étude du modèle SARIMA, nous utiliserons, dans le cadre de l'économétrie des séries temporelles, les techniques d'analyse (stationnarité, stratégie Box-Jenkins, statistique inférentielle ...) et de prévision (prédiction, performance, ...) des séries chronologiques.

Pour l'étude des cinq modèles neuronaux retenus (réseau à propagation-avant FFNN, à fonctions de base radiales RBFN, à régression généralisée GRNN, à délais temporels TDNN et modèle neuronal moyenne mobile autorégressif à variables exogènes NNARMAX), nous appliquerons, aux données, les procédures d'apprentissage 'rétro propagation BP' ou 'rétro propagation dans le temps BPTT' selon le cas de chaque modèle et aux poids, les techniques d'optimisation (l'algorithme d'entraînement Levenberg-Marquardt LM,...). Les prévisions seront effectuées selon la méthode recursive multi-step neural network RMSNN. Eviews9 et Matlab 2011a seront nos logiciels de base.

Pour l'étude du modèle Flou, nous appliquerons aux données, le type d'inférence Mamdani. L'output prévu $\hat{\pi}_t$ est calculé à l'issue des opérations d'agrégation par la méthode de la « coupure » et de défuzzification par la méthode « centroïde ». Les fonctions triangulaire, trapézoïdale, gaussienne,.... constitueront les principales fonctions d'appartenance.

Ce travail est structuré en sept chapitres:

Le premier chapitre se consacrera à la prévision des IPC par les méthodes traditionnelles statistiques SARIMA (p,d,q) (s,P,D,Q). Après un bref rappel du background théorique des modèles statistiques, nous procéderons à une analyse de notre série des IPC, s'étalant sur la période 2001-2020, en suivant la méthodologie de Box-Jenkins (identification, estimation, validation puis prévision). A l'issue de cette dernière, nous présenterons le meilleur modèle SARIMA-GARCH, ses prévisions et les valeurs des critères de performance retenus (RMSE, MAE et R^2).

²Vu le caractère statistique du sujet, les rappels théoriques des modèles se feront au début de chaque chapitre

Les chapitres 2 à 4 traiteront les modèles neuronaux statiques : FFNN, RBFN et GRNN. Dans le chapitre 2, nous présenterons, après un rappel théorique des modèles neuronaux, trois variantes du modèle neuronal FFNN : FFNN1, FFNN2 et FFNN3. Nous exposerons d'abord la structure générale du modèle FFNN puis nous décrirons la procédure d'apprentissage des poids 'BP'; ensuite, pour chacune des trois variantes, nous déterminerons l'architecture représentative que nous entraînerons par l'algorithme LM sous le logiciel MATLAB ; après quoi, nous entamerons la procédure d'ajustement et de prévision du modèle. Enfin nous finirons par la comparaison des prévisions des différentes variantes FFNN entre elles avant de confronter les prévisions de la meilleure variante FFNN à celles du modèle SARIMA.

Le chapitre 3 sera consacré au modèle GRNN. Comme ce modèle ne requiert pas d'entraînement itératif, nous nous contenterons de formuler la procédure mathématique sous-jacente de régression puis exposerons sa structure générale. Ensuite, nous présenterons l'architecture adéquate avant d'entamer la procédure de prédiction et de prévision du modèle. Enfin nous finirons par la comparaison des prévisions de la meilleure variante GRNN avec celles des modèles SARIMA et FFNN.

Le chapitre 4 sera réservé à l'étude du modèle RBFN. Nous exposerons d'abord la structure générale de ce modèle et son mode de fonctionnement ; ensuite, nous formulerons 3 règles d'apprentissage : la pseudo-inverse technique, le « gradient descent learning » et l'« hybridlearning ». Dans le cadre de cette dernière, nous présenterons quelques méthodes de clustering : les algorithmes « k-means », « K-nearestneighbours »,... Enfin, nous entamerons la procédure de choix et de prévision par ce modèle. Nous terminerons par la comparaison des performances de la meilleure variante RBFN avec celles des modèles SARIMA, FFNN et GRNN.

Le chapitre 5 initiera les modèles neuronaux dynamiques et se subdivisera en 2 sections. La première section sera dédiée au modèle TDNN. Après une définition des modèles dynamiques non récurrents, nous exposerons la structure et le mode de fonctionnement du modèle TDNN. Ensuite, nous choisirons l'architecture appropriée puis entamerons la procédure de prédiction et de prévision par ce modèle. Nous terminerons cette section par la performance relative du modèle TDNN. La seconde section traitera du modèle neuronal autorégressif NNAR. Nous appliquerons ici la même méthodologie poursuivie dans l'étude du modèle TDNN.

Le chapitre 6 traitera des modèles neuronaux récurrents. Après une définition des modèles récurrents RNN et de leur mode de fonctionnement, nous présenterons la structure et la formalisation mathématique du modèle NNARMAX. Nous décrirons ensuite l'algorithme d'apprentissage BPTT et l'illustrons par un exemple. Après l'exposition de l'architecture appropriée et de la procédure de prévision par ce modèle, nous terminerons par la performance comparée entre le modèle NNARMAX et les modèles SARIMA, FFNN, GRNN, RBFN, TDNN et NNAR.

Enfin, le chapitre 7 sera dédié au modèle FL. Après un bref rappel de la logique floue et de la théorie des ensembles flous, nous procéderons de la même méthodologie appliquée aux chapitres précédents.

REVUE DE LITTERATURE

En 1999, Stock et Watson (USA) s'interrogèrent sur la relative performance des modèles non linéaires; ils mirent en confrontation des modèles univariés linéaires (AR, ES) avec des modèles non linéaires (LSTAR, RNA)(Stock,1999). Leurs résultats furent malheureusement non concluants.

En 2000, Moshiri et Cameron (CANADA), en comparant divers modèles RNA (BPN, RBFN, RNN) avec une panoplie de modèles statistiques (ARIMA, VAR, BVAR), conclurent à une supériorité limitée des modèles RNA (Moshiri,2000).

En 2004, McNelis (USA) construisit une structure composite de modèles non linéaires appelée "trimmed ANN-basedthick model" et conclut à sa supériorité relativement au modèle AR(p) (McNelis, 2004).

En 2005, Nakamura (USA), quant à elle, aboutit à une supériorité conditionnée des modèles ANN (FFNN) relativement à AR(1) : l'horizon prévisionnel devrait, dit-elle, être de 2 trimestres au plus, la spécification simple et les procédures d'estimation spécialisées (Nakamura, 2005).

En 2006, Binner et al (USA) présentèrent un nouveau modèle non linéaire hybride, le "MS-AR" (Markov SwitchingAutoRegressive); ils le mirent en compétition avec des modèles RNA (RNN) et AR(p) et confirmèrent sa relative supériorité (Binner, 2006).

En 2007, Hu (USA, JAPON, HONG KONG, TAIWAN) poursuivit la même idée. En appliquant le modèle 'courbe de Phillips', il mit en compétition le modèle naïf, linéaire, ... (RNA-thin) et des modèleshybrides 'ANN-thick' (ANN-Thick- W_e , ANN-Thick- W_o). Il aboutit à la supériorité des modèles ANN-thick sur le modèle naïf et dans certains cas seulement sur le modèle linéaire (Hu, 2007).

En 2007, Neves (Brazil), dans la classe des modèles NNARMAX, mit en rivalité le modèle neuronal NNAR(p) contre le modèle AR(p) et aboutit à la supériorité de NNAR(p) (Neves, 2007).

En 2009, Binner (USA) eut l'idée de mettre au point un modèle dynamique mixte de type NNARX basé sur la méthode Kernel "the kernel recursive least square regression (KRLSR)"

qu'il mit en compétition avec le modèle RNN ; les résultats montrèrent que le modèle KRLSR est plus performant en prévisibilité que le modèle RNN (Binner, 2009).

En 2010, Düzgün (Turquie) testa le modèle neuronal GRNN qui selon lui n'a fait l'objet d'aucune publication et le mit en challenge avec le modèle ARIMA et le modèle neuronal FFNN. Les calculs plaidèrent en faveur du modèle GRNN relativement aux modèles FFNN et ARIMA (Düzgün, 2010)

En 2011, Ghodsi et al (Iran) construisirent des modèles de prévision des prix du marché d'électricité de la ville canadienne d'Ontario et conclurent à la supériorité du modèle neuronal FL relativement aux modèles ANN et ARIMA (Ghodsi, 2011).

En 2013, Mehri (Iran) compara, dans la prédiction de l'éclosion chez les poules pondeuses, les modèles ANN, FL et MLR. Il conclut à la suprématie du modèle ANN (Mehri, 2013).

En 2014, Rahmati et al (Iran), construisirent des modèles de prévision des besoins de consommation en eau de la ville de Téhéran et aboutirent à la supériorité du modèle ANFIS relativement aux modèles ARMA, ANN et MLR (Rahmati, 2014).

En 2015, Djurovic et al (Serbie), dans la prévision du niveau des eaux souterraines, aboutirent à l'équivalence des modèles ANN et ANFIS du point de vue performance (Djurovic,2015). A la même date, Ramesh (Inde) compara les modèles ARIMA et ANFIS relativement à la prévision de la météo. Il aboutit à la conclusion que le modèle ARIMA est le plus performant bien qu'il prenne relativement plus de temps (Babu, 2015).

En 2016, Ranjithet al (congrès mondial d'engineering de Londres) comparèrent les modèles ANFIS et ANN dans la prévision des énergies photovoltaïques (PV) et éoliennes. Ils conclurent à ce que le modèle ANN est plus performant que le modèle ANFIS dans la prévision de PV (Kuamr, 2016).

.....

A l'issue de cette revue de la littérature, nous remarquons qu'il n'y a pas d'unanimité concernant la précellence d'un modèle, ce qui justifierait notre recherche. Les modèles de prévisions regroupent les modèles statistiques, neuronaux et flous.

CH1 PREVISION DES IPC PAR LES METHODES TRADITIONNELLES

Les modèles de prévision traditionnels peuvent être sectionnés en modèles structurels et statistiques. Ces derniers feront l'objet de notre travail.

1.1 LES MODELES STRUCTURELS

Ce sont des modèles de régression basés sur des variables explicatives exogènes. Le modèle non linéaire :

$$Y = f(X; \theta) + \varepsilon$$

$$Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}; \quad \theta = \begin{pmatrix} \theta_1 \\ \vdots \\ \theta_p \end{pmatrix}; \quad X = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{k1} \\ \vdots & & & \vdots \\ x_{1n} & x_{2n} & \cdots & x_{kn} \end{pmatrix}; \quad \varepsilon = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}; \quad f: \text{non linéaire} \quad (1.1)$$

est estimé par les méthodes 'trials and errors', 'optimisation' ou 'linéarisation' (Gajarati, 2004)

1.2 LES MODELES STATISTIQUES

Une chronique $(y_t, t \in \mathbb{Z})$ serait, pour ces modèles, la réalisation d'un processus stochastique $(Y_t, t \in \mathbb{Z})$; le passé de la variable étudiée est l'unique variable explicative. Ces modèles conviennent à la prévision à court terme. Le modèle ARMA (p,q) est leur modèle de base. C'est un modèle linéaire stationnaire de représentation générale $\Phi(L) Y_t = \theta(L) \varepsilon_t$ où $\Phi(L) =$ polynôme de retard AR ; $\theta(L) =$ polynôme de retard MA. En cas d'hétéroscédasticité des résidus, on ajoutera au modèle ARMA un modèle GARCH (p,q) des erreurs.

1.2.1 LES MODELES DE BOX-JENKINS ET DE SIMS³

1.2.1.1 LE MODELE SARIMA (P, D, Q) (s, P, D, Q)

Les processus stochastiques P.S. peuvent être stationnaires ou non. Le modèle ARMA (p,q)⁴ est un processus stationnaire $(X_t, t \in \mathbb{Z})$ qui vérifie :

³Sims (Christopher.A.), *Macroeconomics and reality*, pp.1-48, 1980

⁴Hurlin (C.), *Cours d'économétrie appliquée: séries temporelles*, chap. 1, 2004

$$\begin{aligned} \Phi(L) X_t &= \delta + \theta(L) \varepsilon_t \\ \text{où } \varepsilon_t &\sim iid(0, \sigma^2) \\ \Phi(L) &= \sum_{i=0}^p \phi_i L^i \\ \theta(L) &= \sum_{i=0}^q \theta_i L^i \\ L^p(X_t) &= X_{t-p} \end{aligned} \tag{1.2}$$

p : ordre du polynome AR
q : ordre du polynome MA
 $(\phi_p, \theta_q) \in R^{2*}$
 les racines de $\Phi(L)$ et $\theta(L)$ ont des modules > 1
 $\delta \in R$

Un P.S. $(X_{t \in Z})$ non stationnaire est I(d) « intégré d'ordre d » si $(1-L)^d X_t$ est stationnaire.

Le modèle ARIMA (p, d, q)⁵ est un processus non stationnaire I(d) qui vérifie :

$$\begin{aligned} \Phi(L) (1-L)^d X_t &= \delta + \theta(L) \varepsilon_t \\ \text{où } \varepsilon_t &\sim iid(0, \sigma^2) \\ \Phi(L) &= \sum_{i=0}^p \phi_i L^i \\ \theta(L) &= \sum_{i=0}^q \theta_i L^i \\ L^p(X_t) &= X_{t-p} \end{aligned} \tag{1.3}$$

p : ordre du AR
d : ordre de différenciation
q : ordre du MA
 $\Phi_0 = \theta_0 = 1$

Le modèle SARIMA multiplicatif (p, d, q) (s, P, D, Q) est un prolongement du modèle ARIMA (p, d, q) contenant des parties saisonnières $\Phi_s(L), \theta_s(L), \dots$

La représentation SARIMA (p, d, q) (s, P, D, Q) (Poo, 2003) est un processus non stationnaire qui vérifie :

⁵Hurlin (C.), *op. cit.*, chap. 2

$$\Phi(L) \Phi_s(L)(1-L)^d (1-L^s)^D X_t = \delta + \theta(L)\theta_s(L) \varepsilon_t$$

où $\varepsilon_t \sim iid(0, \sigma^2)$

$$\Phi(L) = \sum_{i=0}^p \phi_i L^i \quad (1.4)$$

$$\theta(L) = \sum_{i=0}^q \theta_i L^i$$

$$\Phi_s(L) = \sum_{i=0}^p \phi_{s,i} L^{Pi} \quad \text{polynôme de retard AR saisonnier}$$

$$\theta_s(L) = \sum_{j=0}^q \phi_{s,j} L^{Qj} \quad \text{polynôme de retard MA saisonnier}$$

$$L^p (X_t) = X_{t-p}$$

p : ordre du polynome AR

d, D : ordres de différenciation

q : ordre du polynome MA

$$(1-L)X_t = X_t - X_{t-1}$$

$$(1-L^s)X_t = X_t - X_{t-s}$$

La prévision, dans ce cadre, ne s'effectue qu'après identification, estimation et validation du meilleur modèle. Récursivement, le prédicteur du processus ARMA (p,q) à l'horizon h, est :

$$\hat{Y}_{t+h} = \hat{Y}_t(h) = E(Y_{t+h} / Y_t, Y_{t-1}, \dots) = \sum_{k=1}^{h-1} \phi_k \hat{Y}_{t+h-k} + \sum_h^{t+h} \phi_k Y_{t+h-k} \quad (1.5)$$

En cas d'hétéroscédasticité des résidus, on complètera le modèle ARMA par un modèle GARCH (p,q) des erreurs tel que :

$$\Phi(L) X_t = \delta + \theta(L) \varepsilon_t$$

$$\varepsilon_t / I_{t-1} \sim (0, \sigma_t^2)$$

$$\varepsilon_t = \theta \varepsilon_{t-1} + \eta_t$$

$$\varepsilon_t = \eta_t \sigma_t$$

$$\sigma_t^2 = c + \sum_{i=1}^p \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^q \beta_j \sigma_{t-j}^2 = c + \alpha(L) \varepsilon_t^2 + \beta(L) \sigma_t^2$$

$$\eta_t \sim iid N(0,1)$$

$$\alpha_0 > 0 \quad \alpha_i > 0 \quad \beta_j > 0 \quad 0 \leq \sum_{i=1}^p \alpha_i + \sum_{j=1}^q \beta_j < 1$$

1.2.1.2 LE MODELE VAR(P)

Ce modèle (Kirchgassner, 2012), contrairement à celui du SE en économétrie structurelle, est un modèle AR(p) multivarié où le passé de la variable vectorielle X_t est l'unique explicative.

Un P.S. vectoriel ($X_t, t \in \mathbb{Z}$) de dimension (n, 1) admet une représentation générale VAR(p) si :

$$\Phi(L)X_t = c + \varepsilon_t$$

où c est un vecteur de constantes
 $(n,1)$

$$\Phi(L) = \sum_{i=0}^{\infty} \Phi_i L^i \quad (1.6)$$

$$\Phi_i \text{ satisfait } \Phi_0 = I_0 \text{ et } \Phi_i \neq 0_n \quad \forall i \in \overline{0,p}$$

(n,n)

$$\varepsilon_t \sim \text{iid} (0_n, \Omega),$$

(n,n)

$$\Omega : \text{symétrique d.p.}$$

(n,n)

$$X_t = \begin{pmatrix} x_{1,t} \\ x_{2,t} \\ \dots \\ x_{n,t} \end{pmatrix} \quad \varepsilon_t = \begin{pmatrix} \varepsilon_{1,t} \\ \varepsilon_{2,t} \\ \dots \\ \varepsilon_{n,t} \end{pmatrix} \quad \Phi_i = \begin{pmatrix} \Phi_{1,1}^i & \Phi_{1,2}^i & \dots & \Phi_{1,n}^i \\ \Phi_{2,1}^i & \Phi_{2,2}^i & \dots & \Phi_{2,n}^i \\ \dots & \dots & \Phi_{j,k}^i & \dots \\ \Phi_{n,1}^i & \Phi_{n,2}^i & \dots & \Phi_{n,n}^i \end{pmatrix} \quad \forall i \in \overline{0,p}$$

La forme réduite du VAR(p) est: $X_t = c - \phi_1 X_{t-1} - \phi_2 X_{t-2} \dots - \phi_p X_{t-p} + \varepsilon_t$ (1.7)

VAR(p) est stationnaire si les racines $\lambda_i \quad i = \overline{1,n}$ du déterminant du polynôme matriciel $\Phi(L)$ sont supérieurs à 1.

$$|\Phi(\lambda_i)| = |I_n \lambda_i^p - \Phi_1 \lambda_i^{p-1} - \Phi_2 \lambda_i^{p-2} - \dots - \Phi_{p-1} \lambda_i - \Phi_p| = 0 \quad \text{pour } |\lambda_i| > 1, \quad \forall i = \overline{1,n}$$

Du fait que tout VAR(p) stationnaire peut se réduire à un VAR(1) centré $X_t = \Phi_0 + \Phi_1 X_{t-1} + \varepsilon_t$ et s'écrire sous la forme d'un VMA(∞), le prédicteur d'un VAR(p) aux horizons 1 et h s'écrit :

$$\hat{X}_{T+1} = E(X_{T+1} / X_T, X_{T-1}, \dots, X_1) = \hat{\Phi}_0 + \hat{\Phi}_1 X_T \quad (1.8)$$

$$\hat{X}_{T+h} = \hat{X}_T(h) = E(X_{T+h} / X_T, X_{T-1}, \dots, X_1) = (1 + \hat{\Phi}_1 + \hat{\Phi}_1^2 + \dots + \hat{\Phi}_1^{h-1}) \hat{\Phi}_0 + \hat{\Phi}_1^h X_T$$

X_T, X_{T-1}, \dots, X_1 : échantillon de réalisation disponible

$\hat{\Phi}_0, \hat{\Phi}_1$: estimateurs convergents

Le modèle Varma multivarié constitue l'extension du modèle VAR à l'instar du modèle Arma univarié:

$$\Phi(L)X_t = c + \Theta(L) \varepsilon_t \quad / \quad \Theta(L) = \sum_{i=0}^q \Theta_i L^i$$

1.2.2 PREVISION DES IPC PAR LE MODELE SARIMA (p,d,q)(s,P,D,Q)

1.2.2.1 DONNEES

Considérons le modèle SARIMA (p, d, q) (s, P, D, Q) pour prévoir les IPC en Algérie (π_t). Les données sont tirées des séries «collections statistiques : IPC » publiées par l'ONS^{6,7}. La période totale s'étale sur 228 mois, de 2002 :01 à 2020:12; la période de prévision couvre 18 mois. Le prétraitement des données se restreint à leur dessaisonnalisation. Parce que nous considérons un horizon de court terme, nous nous contenterons des valeurs historiques du taux d'inflation $\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}$; $k \in \mathbb{N}$ comme variables explicatives.

1.2.2.2 STRATEGIE DE PREVISION

Les prévisions s'effectueront selon la méthode récursive RMSSN, les procédures statique et dynamique. Les critères d'appréciation des prévisions seront le RMSE, MAE et R². Les prévisions se feront à l'horizon 18 (2021 :01 - 2022 :06).Les prédictions se feront par régression linéaire.

1.2.2.3 SPECIFICATION DU MODELE ET PREVISION

Nous procéderons d'abord à l'identification du modèle SARIMA (p,d,q)(s,P,D,Q) adéquat (1.2.2.3.1), nous estimerons ensuite les paramètres de ce modèle (1.2.2.3.2) enfin nous tenterons de valider ce dernier (1.2.2.3.3) avant d'effectuer des prévisions (1.2.2.3.4) (Ngo, 2013) . Notre logiciel de base sera Eviews9.

1.2.2.3.1 IDENTIFICATION DU MODELE SARIMA (p,d,q)(s,P,D,Q)

L'étude de la stationnarité de la série chronologique sera entreprise au moyen des tests informels (analyses des plots et des moments) puis des tests formels (tests DF, ADF).Une fois la série stationnarisée, nous procéderons à sa spécification.

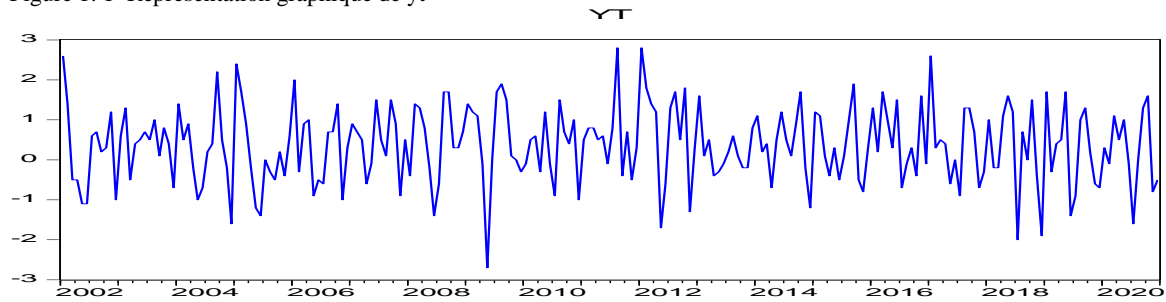
A. ANALYSE DES PLOTS ET DES MOMENTS

L'examen du graphe d'Yt (figure 1.1) augure une saisonnalité et une stationnarité en moyenne.

⁶ ONS, *Indices des prix à la consommation*, rétrospective statistique 1962-2011 p.16-23

⁷ ONS, *Indices des prix à la consommation*, 301, dec.2020 ; 216 , évolution 2010-2019

Figure 1. 1 Représentation graphique de yt



source : Eviews

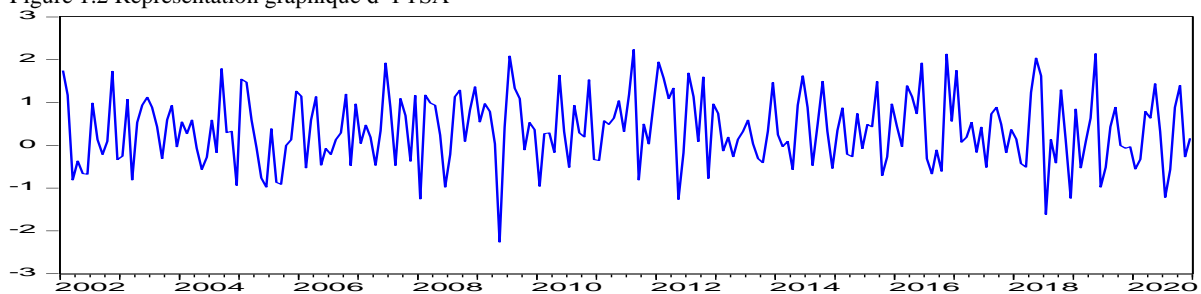
L'analyse de la variance d'Yt (tableau 1.1) affirme l'existence d'une saisonnalité

Tableau1. 1 Test ANOVA de Yt source : Eviews

Source variations	Somme carrés	D.L.	Moyennes carrés	F	P	Valeur critique F
Lignes	5.48	18	0.30	0.42	0.941	1.66
Colonnes	56.69	11	5.15	7.19	2.92E-10	1.84
Erreur	141.95	198	0.72			
Total	204.12	227				

Le correlogramme d'Yt (ACF) confirme l'existence d'une saisonnalité (tableau 1.24, annexe A). Les résultats du Test de BUYS BALLOT ($\sigma_i = 0.13Y_i + 0.86$; $\alpha_{prob} = 0.67$) certifient que la série Yt est additive. Nous procéderons alors à sa dessaisonalisation (fig 1.2) par la méthode MA.

Figure 1.2 Représentation graphique d'YTSA



source : Eviews

Le correlogramme d'YTSA (tableau 1.25, annexe B) et le test ANOVA (tableau 1.2) authentifient la dessaisonalisation d'Yt; ce qui nous permet d'entamer les tests formels.

Tableau1. 2 Test ANOVA d'YTSA source : Eviews

Source variations	Somme carrés	D.L.	Moyennes carrés	F	P	Valeur critique F
Lignes	8.73	17	0.51	0.36	0.991	1.68
Colonnes	1.61	11	0.15	0.10	0.9990	1.84
Erreur	264.85	187	1.42			
Total	275.19	215				

B. TESTS FORMELS

En suivant la stratégie Dukkey-Fuller (DF), nos tests partiront du modèle le plus complet (modèle 3) au modèle le plus simple (modèle 1).

i. STRATEGIE DE TESTS DF

$$\Delta y = \varphi y_{t-1} + c + \beta t + \varepsilon_t$$

D'après le tableau 1.3, il n'y a pas de racine unitaire et le coefficient « β » est non significatif, passons alors au modèle 2.

Tableau1.3 Test DF du modèle 3 source : Eviews

Null Hypothesis: Y TSA has a unit root
Exogenous: Constant, Linear Trend
Lag Length: 0 (Fixed)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-14.47035	0.0000
Test critical values:		
1% level	-3.998997	
5% level	-3.429745	
10% level	-3.138397	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation

Dependent Variable: D(Y TSA)

Method: Least Squares

Date: 09/30/21 Time: 12:20

Sample (adjusted): 2002M02 2020M12

Included observations: 227 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Y TSA(-1)	-0.959774	0.066327	-14.47035	0.0000
C	0.320053	0.110030	2.908765	0.0040
@TREND("2002M01")	0.000181	0.000817	0.222005	0.8245
R-squared	0.483180	Mean dependent var		-0.006963
Adjusted R-squared	0.478566	S.D. dependent var		1.117250
S.E. of regression	0.806770	Akaike info criterion		2.421573
Sum squared resid	145.7968	Schwarz criterion		2.466836
Log likelihood	-271.8485	Hannan-Quinn criter.		2.439837
F-statistic	104.7100	Durbin-Watson stat		1.995120
Prob(F-statistic)	0.000000			

$$\text{.MODELE 2 : } \Delta y = \rho y_{t-1} + c + \varepsilon_t$$

Selon le tableau 1.4, il n'y a pas de racine unitaire et le coefficient « c » est significatif.

Tableau1. 4 Test DF du modèle 2 source : Eviews

Null Hypothesis: Y TSA has a unit root

Exogenous: Constant

Lag Length: 0 (Fixed)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-14.50031	0.0000
Test critical values:		
1% level	-3.459101	
5% level	-2.874086	
10% level	-2.573533	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation

Dependent Variable: D(Y TSA)

Method: Least Squares

Date: 09/30/21 Time: 12:26

Sample (adjusted): 2002M02 2020M12

Included observations: 227 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
Y TSA(-1)	-0.959721	0.066186	-14.50031	0.0000
C	0.340715	0.058567	5.817517	0.0000
R-squared	0.483067	Mean dependent var		-0.006963
Adjusted R-squared	0.480769	S.D. dependent var		1.117250
S.E. of regression	0.805064	Akaike info criterion		2.412982
Sum squared resid	145.8289	Schwarz criterion		2.443158
Log likelihood	-271.8735	Hannan-Quinn criter.		2.425158
F-statistic	210.2591	Durbin-Watson stat		1.994765
Prob(F-statistic)	0.000000			

N'ayant pas de racine unitaire selon les résultats du tableau 1.4, Y TSA est alors I(0) avec dérive c.à.d. $Y TSA_t = \rho Y TSA_{t-1} + c + \varepsilon_t$.

ρ et c sont significatifs d'après le tableau 1.5.

Tableau1.5 Régression du modèle 2 source : Eviews

Dependent Variable: D(YTSA)

Method: Least Squares

Date: 10/02/21 Time: 12:20

Sample (adjusted): 2002M02 2020M12

Included observations: 227 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
YTSA(-1)	-0.959721	0.066186	-14.50031	0.0000
C	0.340715	0.058567	5.817517	0.0000
R-squared	0.483067	Mean dependent var		-0.006963
Adjusted R-squared	0.480769	S.D. dependent var		1.117250
S.E. of regression	0.805064	Akaike info criterion		2.412982
Sum squared resid	145.8289	Schwarz criterion		2.443158
Log likelihood	-271.8735	Hannan-Quinn criter.		2.425158
F-statistic	210.2591	Durbin-Watson stat		1.994765
Prob(F-statistic)	0.000000			

Vérifions maintenant que les ϵ_i sont des bruits blancs. A partir du lag 13 du correlogramme ci-dessous (tab 1.6), $Q\text{-stat} > \chi_{0,95}^2$ ($29.32 > 22.36$), on rejette alors l'hypothèse d'absence d'autocorrelation $H_0: \rho_1 = \rho_2 = \dots = 0$, ceci nous permet d'entamer la stratégie des tests ADF.

Tableau1.6 Correlogramme du modèle 2 source : Eviews

Date: 10/01/21 Time: 10:31

Sample: 2002M01 2022M12

Included observations: 227

Q-statistic probabilities adjusted for 1 dynamic regressor

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.001	0.001	7.E-05	0.993
		2 -0.17...	-0.17...	7.3290	0.026
		3 -0.08...	-0.09...	9.1264	0.028
		4 -0.08...	-0.12...	10.918	0.028
		5 0.032	-0.00...	11.161	0.048
		6 0.083	0.038	12.780	0.047
		7 -0.01...	-0.03...	12.852	0.076
		8 -0.00...	0.013	12.853	0.117
		9 0.078	0.089	14.314	0.112
		1... -0.01...	-0.00...	14.395	0.156
		1... 0.119	0.156	17.818	0.086
		1... 0.020	0.040	17.919	0.118
		1... -0.21...	-0.15...	29.320	0.006
		1... 0.069	0.099	30.484	0.007
		1... -0.03...	-0.09...	30.702	0.010
		1... 0.036	0.037	31.019	0.013
		1... 0.132	0.084	35.355	0.006
		1... -0.05...	-0.05...	36.136	0.007
		1... -0.03...	0.036	36.355	0.010
		2... -0.02...	-0.05...	36.488	0.013

ii. STRATEGIE DES TESTS ADF

Calculons, d’abord, l’ordre de retard optimal « p* » avant d’entamer ces tests. D’après les critères AIC et SC, p*=3 (tab 1.7).

Tableau1.7 Choix du retard optimal source : Eviews

P	Model 3			Model 2			Model 1		
	AIC	SC	HQ	AIC	SC	HQ	AIC	SC	HQ
0	2.421	2.466	2.439	2.412	2.433	2.425	2.544	2.559	2.550
1*	2.392	2.453	2.417	2.384	2.429*	2.402*	2.552	2.582	2.564
2	2.391	2.467	2.422	2.382	2.443	2.407	2.552	2.598	2.570
3*	2.385	2.477	2.422	2.377*	2.453	2.407	2.562	2.623	2.587
4	2.392	2.499	2.435	2.383	2.474	2.420	2.542	2.618	2.573

Nous procéderons aux tests ADF suivant la méthodologie appliquée aux tests DF.

• **MODELE 3** $\Delta y_t = \phi y_{t-1} + \sum_{j=1}^p \xi_j \Delta y_{t-j} + c + \beta t + \varepsilon_t$

D’après le tableau 1.8, il n’ya pas de racine unitaire et le coefficient « β » est non significatif, passons alors au modèle 2.

Tableau1.8 Test ADF du modèle 3 source : Eviews

Null Hypothesis: YTSA has a unit root
Exogénous: Constant, Linear Trend
Lag Length: 3 (Fixed)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-9.708956	0.0000
Test critical values:		
1% level	-3.999552	
5% level	-3.430013	
10% level	-3.138555	

*MacKinnon (1996) one-sided p-values.
Augmented Dickey-Fuller Test Equation
Dependent Variable: D(YTSA)
Method: Least Squares
Date: 08/16/21 Time: 10:48
Sample (adjusted): 2002M05 2020M12
Included observations: 224 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
YTSA(-1)	-1.406136	0.144829	-9.708956	0.0000
D(YTSA(-1))	0.418503	0.119525	3.501381	0.0006
D(YTSA(-2))	0.217870	0.092820	2.347240	0.0198
D(YTSA(-3))	0.132880	0.066664	1.993278	0.0475
C	0.488077	0.118315	4.125242	0.0001
@TREND("2002M01")	0.000143	0.000814	0.176261	0.8603
R-squared	0.513407	Mean dependent var		0.002370
Adjusted R-squared	0.502247	S.D. dependent var		1.115840
S.E. of regression	0.787244	Akaike info criterion		2.385863
Sum squared resid	135.1061	Schwarz criterion		2.477246
Log likelihood	-261.2166	Hannan-Quinn criter.		2.422749
F-statistic	46.00261	Durbin-Watson stat		2.001188
Prob(F-statistic)	0.000000			

• **MODELE 2** $\Delta y_t = \phi y_{t-1} + \sum_{j=1}^p \xi_j \Delta y_{t-j} + c + \varepsilon_t$

Selon le tableau 1.9, il n'y a pas de racine unitaire et le coefficient « c » est significatif,

Tableau 1.9 Test ADF du modèle 2 source : Eviews

Null Hypothesis: YTSA has a unit root

Exogenous: Constant

Lag Length: 3 (Fixed)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-9.729390	0.0000
Test critical values:		
1% level	-3.459494	
5% level	-2.874258	
10% level	-2.573625	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation

Dependent Variable: D(YTSA)

Method: Least Squares

Date: 08/16/21 Time: 10:49

Sample (adjusted): 2002M05 2020M12

Included observations: 224 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
YTSA(-1)	-1.405417	0.144451	-9.729390	0.0000
D(YTSA(-1))	0.417964	0.119221	3.505782	0.0006
D(YTSA(-2))	0.217680	0.092608	2.350557	0.0196
D(YTSA(-3))	0.132862	0.066516	1.997438	0.0470
C	0.504393	0.073525	6.860150	0.0000
R-squared	0.513338	Mean dependent var		0.002370
Adjusted R-squared	0.504449	S.D. dependent var		1.115840
S.E. of regression	0.785500	Akaike info criterion		2.377077
Sum squared resid	135.1253	Schwarz criterion		2.453229
Log likelihood	-261.2326	Hannan-Quinn criter.		2.407816
F-statistic	57.75101	Durbin-Watson stat		2.001275
Prob(F-statistic)	0.000000			

En rejetant l'hypothèse de racine unitaire selon les résultats du tableau 1.9, on considère que la série YTSA est générée par un processus stationnaire de la forme :

$$I(0) + \rho y_{t-1} + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-2} + \beta_3 \Delta y_{t-3} + c + \varepsilon_t \text{ ou } AR(4).$$

Les coefficients ρ , β_1 et c sont significatifs (tab 1.10)

Tableau1.10 Régression du modèle 2 source : Eviews

Dependent Variable: D(YTSA)

Method: Least Squares

Date: 08/16/21 Time: 12:58

Sample (adjusted): 2002M06 2020M12

Included observations: 223 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.512422	0.081210	6.309838	0.0000
YTSA(-1)	-1.415005	0.173015	-8.178514	0.0000
D(YTSA(-1))	0.423244	0.147145	2.876367	0.0044
D(YTSA(-2))	0.215315	0.122640	1.755664	0.0806
D(YTSA(-3))	0.133803	0.093872	1.425380	0.1555
D(YTSA(-4))	0.009026	0.067331	0.134058	0.8935
R-squared	0.516879	Mean dependent var		0.003708
Adjusted R-squared	0.505748	S.D. dependent var		1.118171
S.E. of regression	0.786108	Akaike info criterion		2.383093
Sum squared resid	134.0987	Schwarz criterion		2.474766
Log likelihood	-259.7149	Hannan-Quinn criter.		2.420101
F-statistic	46.43264	Durbin-Watson stat		2.006910
Prob(F-statistic)	0.000000			

C. SPECIFICATION

Cherchons p, q, P, Q à partir du correlogramme d'YTSA (tab 1.25 annexe B). Ce dernier nous suggère les modèles suivants : SARIMA(22,0,4) (12,2,0, 2); SARIMA(11,0,27) (12,0,0,0) et SARIMA(47,0,4) (12,2,0,2).

1.2.2.3.2 ESTIMATION

A. MODELE 1 PROPOSE : SARIMA (22,0,4) (12,2,0,2)

Selon son corrélogramme cité en annexe C (tab 1.26), les résidus sont des bruits blancs; aussi ce modèle 1 est candidat.

Tableau1. 11 Régression du modèle 1 SARIMA source : Eviews

Dependent Variable: YTSA
 Method: ARMA Maximum Likelihood (OPG - BHHH)
 Date: 08/14/21 Time: 17:48
 Sample: 2002M01 2020M12
 Included observations: 228
 Failure to improve objective (non-zero gradients) after 41 iterations
 Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.359487	0.043057	8.349152	0.0000
AR(2)	-0.345211	0.073845	-4.674803	0.0000
AR(4)	-0.243160	0.082227	-2.957185	0.0035
AR(11)	0.233215	0.053351	4.371343	0.0000
AR(22)	-0.165059	0.052156	-3.164709	0.0018
AR(12)	0.697963	0.128022	5.451898	0.0000
AR(24)	-0.323767	0.103998	-3.113195	0.0021
MA(2)	0.336697	0.069204	4.865295	0.0000
MA(3)	-0.190200	0.049964	-3.806765	0.0002
MA(4)	0.279642	0.086453	3.234608	0.0014
MA(12)	-0.934599	0.180204	-5.186328	0.0000
MA(24)	0.531941	0.191098	2.783609	0.0059
SIGMASQ	0.474705	0.051037	9.301154	0.0000
R-squared	0.268661	Mean dependent var		0.361404
Adjusted R-squared	0.227842	S.D. dependent var		0.807434
S.E. of regression	0.709513	Akaike info criterion		2.266154
Sum squared resid	108.2328	Schwarz criterion		2.461687
Log likelihood	-245.3416	Hannan-Quinn criter.		2.345046
F-statistic	6.581776	Durbin-Watson stat		1.822823
Prob(F-statistic)	0.000000			

B. MODELE 2 PROPOSE : SARIMA (11,0,27) (12, 0,0,0)

Selon son correlogramme cité en annexe D (tab 1.27), les résidus sont des bruits blancs ; par conséquent ce modèle 2 est retenu comme candidat

Tableau1.12 Régression du modèle 2 SARIMA source : Eviews

Dependent Variable: YTSA
 Method: ARMA Maximum Likelihood (OPG - BHHH)
 Date: 10/03/21 Time: 00:37
 Sample: 2002M01 2020M12
 Included observations: 228
 Convergence achieved after 48 iterations
 Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.373648	0.023217	16.09390	0.0000
AR(2)	-0.193042	0.058022	-3.327052	0.0010
AR(4)	-0.439096	0.115801	-3.791824	0.0002
AR(11)	0.159764	0.067687	2.360335	0.0191
MA(3)	-0.166780	0.062984	-2.647988	0.0087
MA(4)	0.356435	0.128561	2.772509	0.0060
MA(13)	-0.142843	0.068089	-2.097893	0.0371
MA(22)	-0.197457	0.065521	-3.013640	0.0029
MA(27)	-0.226298	0.060560	-3.736774	0.0002
SIGMASQ	0.536042	0.055649	9.632611	0.0000
R-squared	0.174165	Mean dependent var		0.361404
Adjusted R-squared	0.140071	S.D. dependent var		0.807434
S.E. of regression	0.748753	Akaike info criterion		2.322218
Sum squared resid	122.2176	Schwarz criterion		2.472628
Log likelihood	-254.7328	Hannan-Quinn criter.		2.382904
F-statistic	5.108352	Durbin-Watson stat		1.793179
Prob(F-statistic)	0.000003			

C. MODELE 3 PROPOSE :SARIMA (47,0,4) (12,2,0,2)

Enfin, de la même manière que précédemment, ce modèle 3 est aussi candidat (tab 1.28, annexe E)

Tableau1. 13 Régression du modèle 3 SARIMA source : Eviews

Dependent Variable: YTSA

Method: ARMA Maximum Likelihood (OPG - BHHH)

Date: 10/03/21 Time: 23:48

Sample: 2002M01 2020M12

Included observations: 228

Failure to improve objective (non-zero gradients) after 34 iterations

Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.356928	0.040683	8.773453	0.0000
AR(2)	-0.495340	0.093187	-5.315570	0.0000
AR(4)	-0.265788	0.093413	-2.845300	0.0049
AR(11)	0.203068	0.058589	3.465997	0.0006
AR(22)	-0.177790	0.053757	-3.307312	0.0011
AR(12)	0.412437	0.106634	3.867793	0.0001
AR(24)	-0.347256	0.082028	-4.233389	0.0000
AR(27)	-0.167578	0.055278	-3.031574	0.0027
AR(47)	0.152472	0.068295	2.232543	0.0266
MA(2)	0.460446	0.108724	4.235013	0.0000
MA(3)	-0.162493	0.068030	-2.388533	0.0178
MA(4)	0.304989	0.116652	2.614520	0.0096
MA(12)	-0.609346	0.176087	-3.460486	0.0007
MA(24)	0.466821	0.232175	2.010646	0.0456
SIGMASQ	0.459639	0.053555	8.582617	0.0000
R-squared	0.291873	Mean dependent var		0.361404
Adjusted R-squared	0.245329	S.D. dependent var		0.807434
S.E. of regression	0.701433	Akaike info criterion		2.252911
Sum squared resid	104.7976	Schwarz criterion		2.478526
Log likelihood	-241.8319	Hannan-Quinn criter.		2.343940
F-statistic	6.270961	Durbin-Watson stat		1.846563
Prob(F-statistic)	0.000000			

D. CHOIX DU MEILLEUR MODELE

Selon les critères RMSE, R^2 , AIC, SC le meilleur modèle retenu est le modèle 3 (tab 1.16).

Tableau1. 14 Critères du pouvoir prédictif d'YTSA source : Eviews

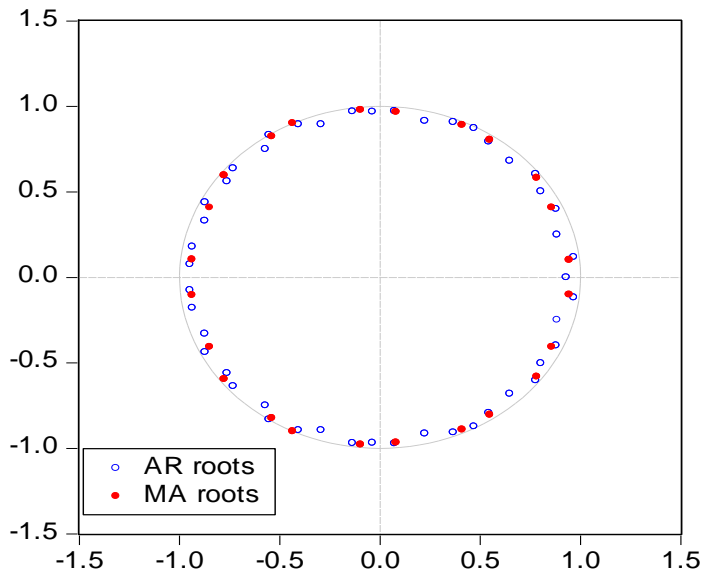
	Model1 SARIMA(22,0,4)(12,2,0,2)	Model2 SARIMA(11,0,27)(12,0,0,0)	Model3 SARIMA(47,0,4)(12,2,0,2)
# coeff signif	11	8	13
sigma ²	0.474	0.536	0.459
R ² adj	0.268	0.174	0.292
aic	2.266	2.322	2.252
sc	2.461	2.472	2.478

Aussi voyons maintenant si on peut valider statistiquement ce modèle.

1.2.2.3.3 VALIDATION

A. Ce modèle est inversible: les modules des racines inversées sont à l'intérieur du cercle 1.

Tableau1. 15 tests de stationnarité et d'inversibilité des racines source Eviews
Inverse Roots of AR/MA Polynomial(s)



B. Les coefficients AR (2,4,11,12,22,24,27,47) et MA(2,3,4,12,24) sont significatifs.

Tableau1. 16 test sur les estimations (coefficients) source : Eviews

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.356928	0.040683	8.773453	0.0000
AR(2)	-0.495340	0.093187	-5.315570	0.0000
AR(4)	-0.265788	0.093413	-2.845300	0.0049
AR(11)	0.203068	0.058589	3.465997	0.0006
AR(22)	-0.177790	0.053757	-3.307312	0.0011
AR(12)	0.412437	0.106634	3.867793	0.0001
AR(24)	-0.347256	0.082028	-4.233389	0.0000
AR(27)	-0.167578	0.055278	-3.031574	0.0027
AR(47)	0.152472	0.068295	2.232543	0.0266
MA(2)	0.460446	0.108724	4.235013	0.0000
MA(3)	-0.162493	0.068030	-2.388533	0.0178
MA(4)	0.304989	0.116652	2.614520	0.0096
MA(12)	-0.609346	0.176087	-3.460486	0.0007
MA(24)	0.466821	0.232175	2.010646	0.0456
SIGMASQ	0.459639	0.053555	8.582617	0.0000

C. Les résidus e_t sont non corrélés dans l'ordre supérieur à 1.

Tableau1. 17 test de non corrélation sérielle des résidus e_t (L-B) source : Eviews

	AC	PAC	Q-Stat	Prob	$\chi^2_{0.95}$
23	-0.031	-0.031	18.235	0.051	18.31
24	-0.001	-0.005	18.235	0.076	19.68
36	0.013	-0.002	24.848	0.358	35.17

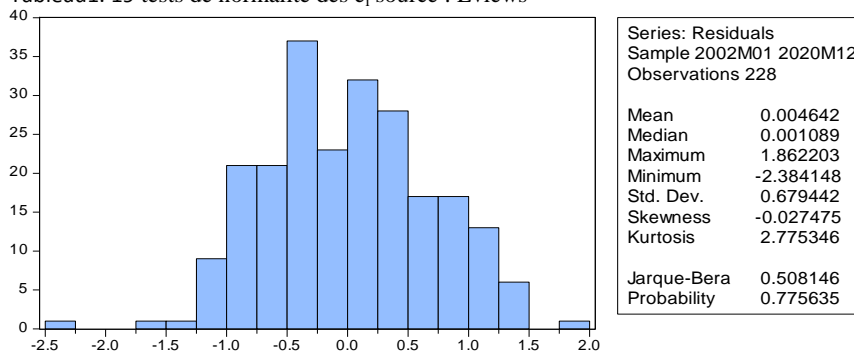
D.La série ne comporte pas de point de rupture

Tableau1. 18 Chow breakpoint test 2017:01source : Eviews

F-statistic	1.095303	Prob. F(15,198)	0.3628
Log likelihood ratio	32.03031	Prob. Chi-Square(15)	0.0064
Wald Statistic	20.51658	Prob. Chi-Square(15)	0.1530

E. Les résidus e_i sont normalement distribués.

Tableau1. 19 tests de normalité des e_i source : Eviews



F. Les résidus sont homoscédastiques, nul besoin d'adjonction du modèle GARCH

Tableau1. 20 test d'homoscédasticité ARCH test source : Eviews
Heteroskedasticity Test: ARCH

F-statistic	0.541358	Prob. F(1,225)	0.4626
Obs*R-squared	0.544859	Prob. Chi-Square(1)	0.4604

G. Les résidus ont une moyenne théorique égale à 0

Tableau1. 21 test de nullité de la moyenne des e_i source : Eviews

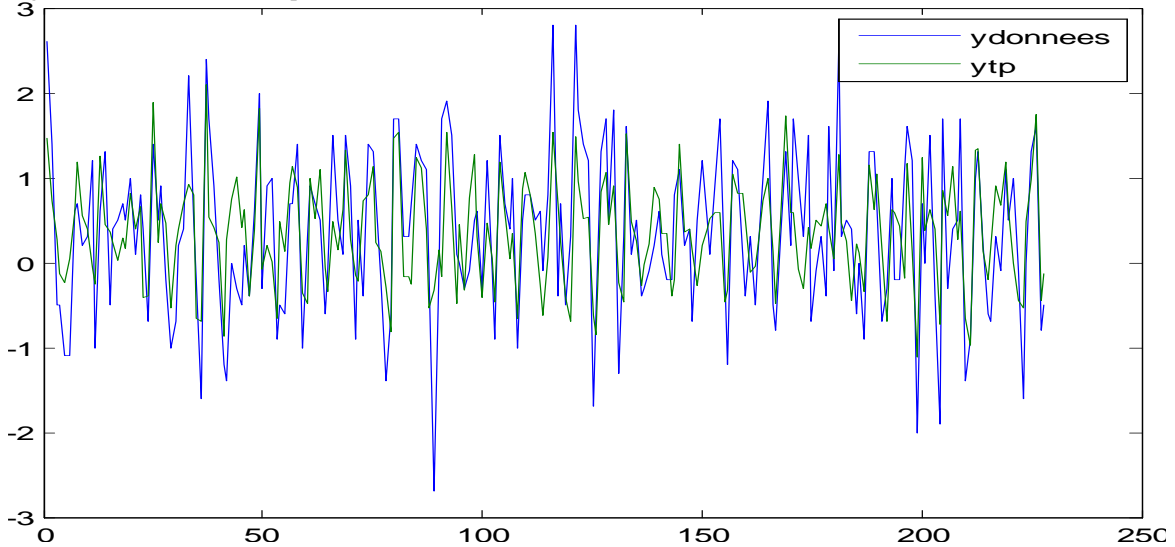
SampleMean = 0.011981
SampleStd. Dev. = 0.437121

Method	Value	Probability
t-statistic	0.400030	0.6895

En résumé, les tests statistiques confirment que ce modèle est stationnaire et que ses résidus sont des bruits blancs gaussiens⁸. Nous pouvons entamer les prévisions. Le meilleur modèle ajustant la série YTSA est alors SARIMA (47,0,4) (12,2,0,2).

Le graphe suivant illustre la qualité de prédiction de la série YTSA par ce modèle.

Figure 1.3 Prédiction d'YTSA par le modèle SARIMA



source : Eviews

1.2.2.3.4 PREVISION

A. PREVISION D'YTSA A partir de ce modèle, nous obtenons les prévisions d'YTSA

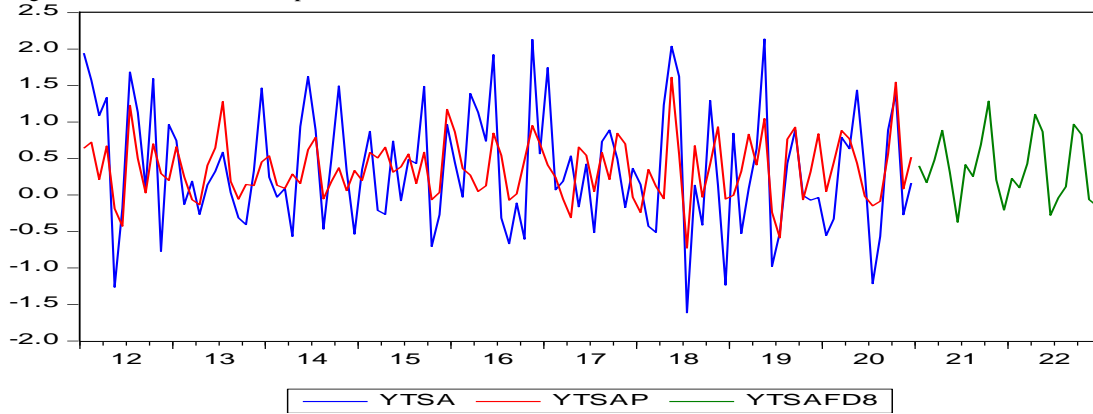
\widehat{YTSA}_{t+h} ; $h = \overline{1,18}$ (tab 1.25) illustrées par le graphe ci-dessous (fig 1.4)

Tableau1. 22 Prédiction d'YTSA par le modèle SARIMA source : Eviews

DATE	2021M01	2021M02	2021M03	2021M04	2021M05	2021M06	2021M07	2021M08	2021M09
YTSAFD8	0,400	0,171	0,479	0,885	0,313	-0,371	0,415	0,256	0,689
DATE	2021M10	2021M11	2021M12	2022M01	2022M02	2022M03	2022M04	2022M05	2022M06
YTSAFD8	1,287	0,209	-0,205	0,226	0,099	0,429	1,104	0,868	-0,276

⁸ $e_i \sim N(0, \sigma)$ Bruit Blanc c.a.de $i \sim iid N(0, \sigma)$

Figure 1.4 Pr evision d'YTSA par le mod le SARIMA



source : Eviews

B. RECOLORIAGE

Les pr evisions d'Yt sont obtenues par saisonnalisation de YTSAF(t): $YTF(t)=YTSAF(t) + \text{factors}$.

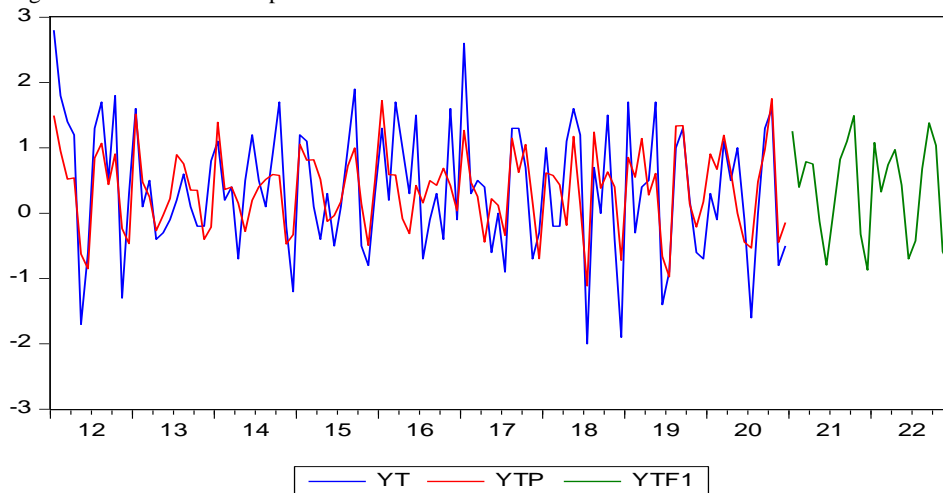
Tableau1. 23 Pr evision de Yt par SARIMA source : Eviews

DATE	2021M01	2021M02	2021M03	2021M04	2021M05	2021M06	2021M07	2021M08	2021M09
YTSAFD8	0,400	0,171	0,479	0,885	0,313	-0,371	0,415	0,256	0,689
FACTORS	0,855	0,228	0,310	-0,134	-0,438	-0,423	-0,386	0,567	0,411
YTF1	1,255	0,399	0,789	0,751	-0,125	-0,795	0,028	0,823	1,100
DATE	2021M10	2021M11	2021M12	2022M01	2022M02	2022M03	2022M04	2022M05	2022M06
YTSAFD8	1,287	0,209	-0,205	0,226	0,099	0,429	1,104	0,868	-0,276
FACTORS	0,205	-0,530	-0,665	0,855	0,228	0,310	-0,134	-0,438	-0,423
YTF1	1,492	-0,321	-0,870	1,081	0,327	0,740	0,970	0,430	-0,700

La s erie des IPC (Y_t) peut  tre mod lis e par le processus SARIMA (47,0,4) (12,2,0,2).

Le graphe ci-dessous illustre la courbe pr evisionnelle d'Yt.

Figure 1.5 Pr evision de Yt par le mod le SARIMA



source : Eviews

C. PERFORMANCE

La précision des prévisions d' Y_{test} appréciée par $RMSE=0.81$, $MAE=0.65$ et $R^2= 0.29$ (tab 1.29, annexe F).

L'ajustement de notre série YTSA par le modèle statistique SARIMA (47,0,4)(12,2,0,2) garch(0,0) est grossier, souvent désynchronisé et décollant (fig 1.3). Les performances de prévision obtenues ($RMSE = 0.81$, $MAE = 0.65$ et $R^2= 0.29$) sont de l'ordre de l'unité, faibles, voire inacceptables. En supposant que ces faibles résultats seraient liés à l'hypothèse fondamentale de linéarité de ce modèle, envisageons maintenant la prévision par les modèles neuronaux non linéaires.

CH2 PREVISION DES IPC PAR LE MODELE FFNN

Le modèle à propagation avant FFNN est le modèle de base des réseaux neuronaux artificiels ANN (Mehrotra,1997).

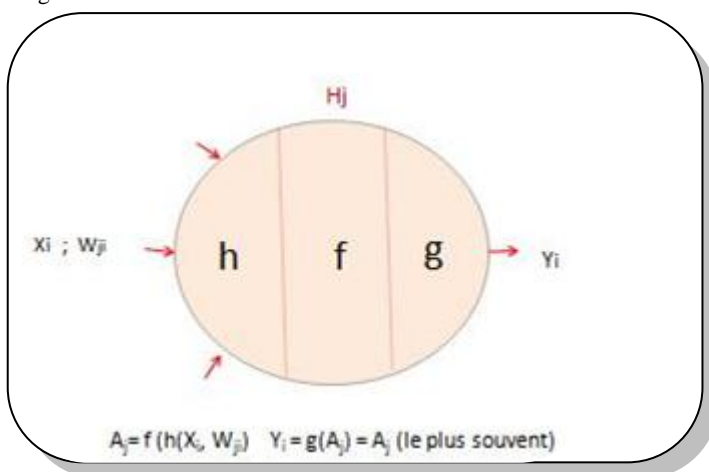
2.1 LES MODELES ANN

Un réseau de neurones désigne des circuits composés de petites unités de calcul (les neurones formels) interconnectées dans une structure particulière. Il cherche à reproduire le fonctionnement élémentaire du cerveau humain par une modélisation de son élément de base, le neurone, et de sa structure, l'interconnexion entre ces neurones. Un réseau de neurones acquiert la connaissance par l'apprentissage.

Le neurone formel (McCulloch&Pitts ,1943) est un processeur très simple qui applique, à chaque vecteur input 'X_i', une fonction d'entrée 'h' (somme, distance,...) puis au résultat obtenu une fonction de transfert 'f' non linéaire (seuil, sigmoïde, gaussienne,..); la fonction d'output 'g' est généralement la fonction linéaire 'purelin'. Les valeurs en sortie 'Y_j' sont obtenues alors selon la formule : $Y_j = g [f (h (X_i, W_{ji}))]$

La combinaison de fonctions (h,f,g) définit le type de neurone: sommateur,distance,..[fig 21].

Figure 2. 1Un neurone formel

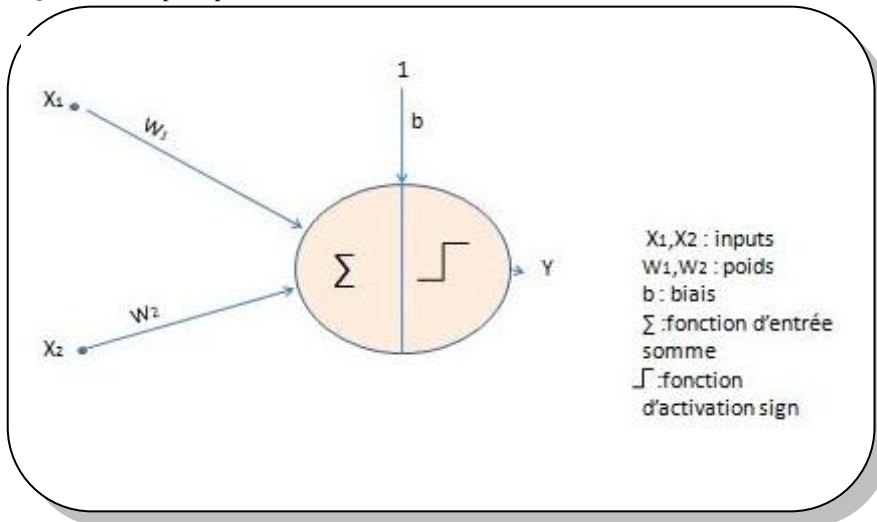


Source : l'auteur

La structure la plus simple d'un réseau de neurones est le perceptron [fig 2.2]. Il est dû à ROSENBLATT (1958).

Il a été inspiré du système visuel ; les valeurs de sortie ne peuvent être que (0,1) ou (-1,1). La règle de modification des poids est la règle de WIDROW-HOF (1960). Elle permet de résoudre les fonctions logiques OR, AND,... excepté XOR.

Figure 2. 2 Le perceptron

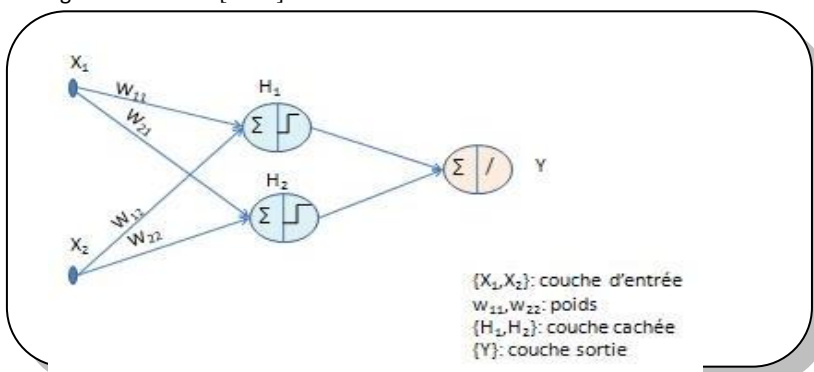


Source : l'auteur

Le perceptron ne survécut pas longtemps. Ces limites furent très tôt établies par PAPERT et MINSKY (1969): En effet, le perceptron ne peut résoudre que les modèles linéairement séparables. Il est incapable, par exemple, de résoudre la fonction logique 'XOR'.

En 1986, Rumelhart introduisit le perceptron multicouches MLP et son algorithme d'apprentissage 'la rétropection du gradient de l'erreur BP (Rumelhart, 1986). Le MLP, caractérisé par l'adjonction de couche (s) intermédiaire(s) à sa structure, est capable de résoudre les modèles non linéairement séparables. La figure suivante [fig 2.3] illustre un MLP à 1 couche cachée comprenant 2 neurones (ou unités). Cette structure MLP a été expérimentée avec succès pour résoudre la fonction logique XOR

Figure 2. 3 MLP [2:2:1]



Source : l'auteur

Les réseaux de neurones peuvent être classés en 2 catégories : les modèles statiques (FFNN, GRNN, RBFN, ...) et dynamiques (TDNN, NOE, NNARMAX). Nous traiterons dans les chapitres 2 à 4 des modèles statiques, 5 et 6 des modèles dynamiques. La comparaison entre ces modèles ANN et le modèle SARIMA (p, d, q) (s,P, D, Q) de point de vue efficacité en prévisibilité sera entreprise.

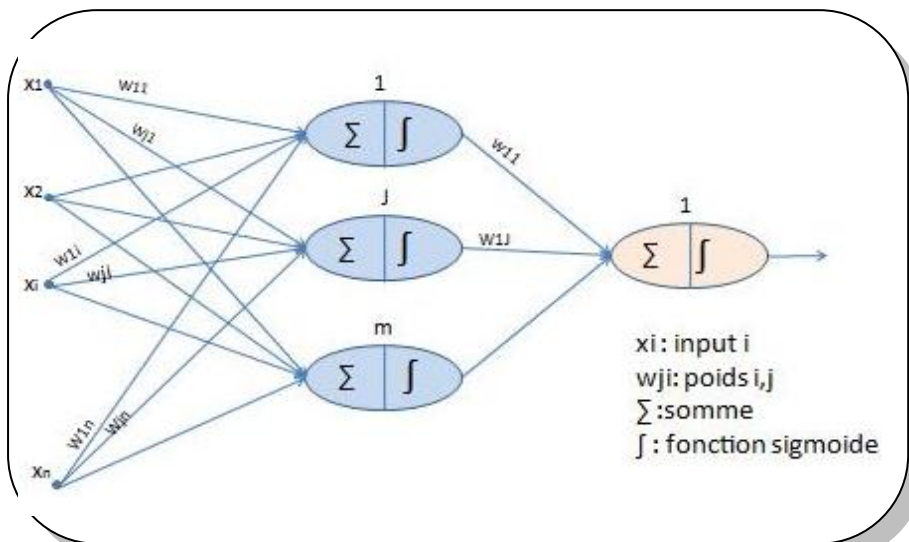
2.2 LE MODELE FFNN

2.2.1 STRUCTURE

Le modèle FFNN est le modèle neuronale type. Statique, supervisé, il est originellement entraîné par l'algorithme BP; parce que généralement il donne de bonnes performances, il est le plus utilisé (Priddy, 2005). Il a été introduit par Rumelhart en 1986 et devint la clé de voûte dans l'histoire des ANN [Fig 2.4].

Le modèle FFNN est un approximateur universel. Il est souvent constitué de 3 couches : couche input, couche cachée (une ou plusieurs) et couche output. Le nombre d'unités de la couche input dépend de la dimension du vecteur input X ; le nombre d'unités de la couche cachée est déterminé par l'erreur de prévision minimale.

Figure 2. 4 Structure FFNN [n:m:1]



Source : l'auteur

2.2.2 L'ALGORITHME BP

2.2.2.1 PROCEDURE D'APPRENTISSAGE DES POIDS W_{kj}

Conformément à la figure [2.5], considérons :

$$y_k(n) = \phi(v_k(n)) \quad (2.1)$$

$$v_k(n) = \sum_j w_{kj}(n) y_j(n) \quad (2.2)$$

$$e_k(n) = d_k(n) - y_k(n) \quad (2.3)$$

$$E(n) = 1/2 \sum_k e_k^2(n) \quad (2.4)$$

$$MSE = \overline{E(n)} = 1/N \sum E(n) \quad (2.5)$$

$d_k(n)$: output final k *actuel*

$y_k(n)$: output final k *prédit*

$E(n)$: demie somme des carrés des erreurs SSE

(n) : n^{ème} itération

y_j : output de l'unité cachée j

Appliquons la méthode d'apprentissage du "delta généralisé" :

$$\Delta w_{kj}(n) = -\eta \nabla_w E(n) = -\eta \nabla_w e_k \quad k=1, \overline{K}$$

Calculons d'abord $\frac{\partial E(n)}{\partial w_{kj}(n)}$

$$\frac{\partial E(n)}{\partial w_{kj}(n)} = \frac{\partial E(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial w_{kj}(n)} =$$

$$\left\{ \begin{array}{l} \frac{\partial E(n)}{\partial e_k(n)} = e_k(n) \quad \frac{\partial e_k(n)}{\partial y_k(n)} = -1 \quad \frac{\partial y_k(n)}{\partial v_k(n)} = \phi'(v_k(n)) \quad \frac{\partial v_k(n)}{\partial w_{kj}(n)} = y_j(n) \end{array} \right\}$$

$$= -e_k(n) \phi'(v_k(n)) y_j(n)$$

$$\text{comme } \delta_k(n) = -\frac{\partial E(n)}{\partial v_k(n)} = -\frac{\partial E(n)}{\partial e_k(n)} \frac{\partial e_k(n)}{\partial y_k(n)} \frac{\partial y_k(n)}{\partial v_k(n)} = e_k(n) \phi'(v_k(n)) \text{ alors}$$

$$\frac{\partial E(n)}{\partial w_{kj}(n)} = -\delta_k(n) y_j(n)$$

Δw_{kj} appliqué aux poids w_{kj} équivaut alors à :

$$\Delta w_{kj}(n) = \eta \delta_k(n) y_j(n)$$

La règle d'actualisation des poids w_{kj} est alors :

$$w_{kj}(n) \leftarrow w_{kj}(n) + \eta \delta_k(n) y_j(n) \quad (\text{Sengupta, 2003})$$

2.2.2.2 PROCEDURE D'APPRENTISSAGE DES POIDS w_{ji}

Considérons, conformément à la figure [2.5] :

$$e_k(n) = d_k(n) - y_k(n) \quad (2.6)$$

$$y_k(n) = \phi(v_k(n)) \quad (2.7)$$

$$v_k(n) = \sum_j w_{kj}(n) y_j(n) \quad (2.8)$$

$$y_j(n) = \phi(v_j(n)) \quad (2.9)$$

$$v_j(n) = \sum_i w_{ji}(n) y_i(n) \quad (2.10)$$

$$\delta_k(n) = \delta v_k(n) = \sum_j w_{kj} \phi'(v_j(n)) \quad (2.11)$$

$$E(n) = 1/2 \sum_k e_k^2(n) \quad (2.12)$$

$$MSE = \overline{E(n)} = 1/N \sum E(n) \quad (2.13)$$

Appliquons la règle d'apprentissage du "delta généralisé" $\Delta w(n) = -\eta \nabla_w E(n)$

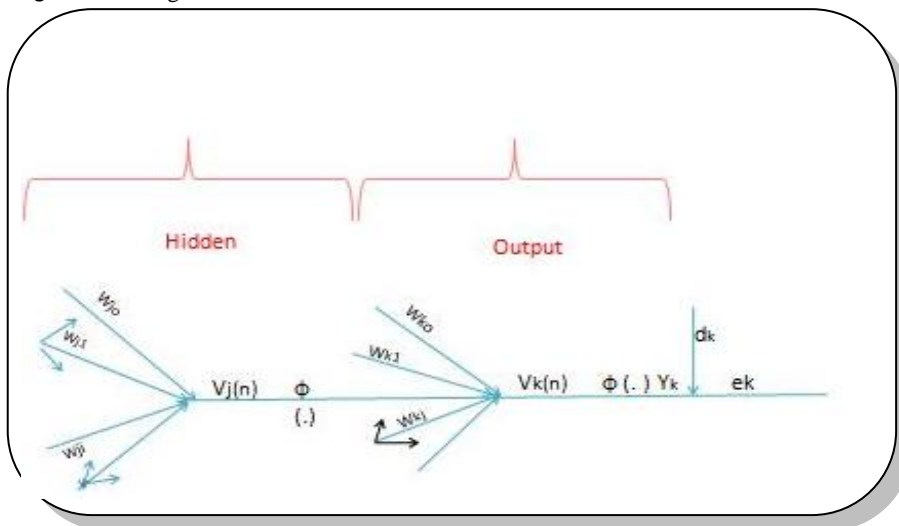
à l'adaptation des poids w_{ji} . Calculons d'abord $\frac{\partial E}{\partial w_{ji}}$:

$$\begin{aligned}
 \frac{\partial E}{\partial w_{ji}} &= -\sum_k [d_k(n) - \phi(v_k(n))] \frac{\partial \phi(v_k(n))}{\partial w_{ji}} \\
 &= -\sum_k [d_k(n) - \phi(v_k(n))] \frac{\partial \phi(in\ k)}{\partial w_{ji}} \\
 &= -\sum_k [d_k(n) - \phi(v_k(n))] \phi'(in\ k) \frac{\partial (in\ k)}{\partial w_{ji}} \\
 &= -\sum_k [d_k(n) - \phi(v_k(n))] \phi'(in\ k) \frac{\partial}{\partial w_{ji}} \sum_j w_{kj} \phi(v_j(n)) \\
 &= -\sum_k [d_k(n) - \phi(v_k(n))] \phi'(in\ k) w_{kj} \frac{\partial \phi(v_j(n))}{\partial w_{ji}} \\
 &= -\sum_k [d_k(n) - \phi(v_k(n))] \phi'(in\ k) w_{kj} \phi'(v_j(n)) \frac{\partial v_j(n)}{\partial w_{ji}} \\
 &= -\sum \Delta_k w_{kj} \phi'(v_j(n)) y_i \left\{ \text{étant donné que } [d(n) - \phi(v_k(n))] \phi'(in\ k) = \Delta_k \quad () \right\} \\
 &= -\Delta_j y_i \left\{ \text{si l'on suppose } \sum \Delta_k w_{kj} \phi'(v_j(n)) = \Delta_j \right\}
 \end{aligned}$$

La règle d'actualisation des poids w_{ji} est alors :

$$w_{ji} \leftarrow w_{ji} + \eta \Delta_j y_i \quad (\text{Dasgupta, 2004})$$

Figure 2. 5 L'algorithme BP



Source : l'auteur

2.2.3 PREVISION

2.2.3.1 DONNEES

Considérons le modèle FFNN pour prévoir les IPC en Algérie (π_t). Nous reprenons les mêmes données prétraitées et hypothèses énoncées dans le modèle SARIMA (p,d,q) (s,P, D, Q) [chapitre 1].

2.2.3.2 ARCHITECTURE DU RESEAU

L'ensemble des variables explicatives $\{\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}\}$ constitue le vecteur input X^k , Le nombre de décalage 'k' est déterminé par l'erreur minimale de prévision. L'output $\hat{\pi}_t$ constitue la valeur prédite de la valeur actuelle de l'IPC relatif π_t . Nous optons pour une seule couche cachée, son nombre d'unités est choisi par la méthode 'trial and error'. sa fonction de transfert est 'tanh'. La couche de sortie comprend un seul neurone et sa fonction est, selon les modèles, «purelin» ou «tanh».

2.2.3.3 METHODOLOGIE DE PREVISION

Les prévisions s'effectuent selon la méthode RMSNN, les procédures statique et dynamique. Les critères d'appréciation des prévisions sont le RMSE, MAE et R². Les données sont subdivisées en 3 groupes : training set (70%), validation set (15%) et testing set (15%) soient respectueusement 160, 34 puis 34 observations. La période de prévision s'étale de 2021:01 à 2022:06. Les prédictions se feront au plus à 8-step-ahead.

FFNN1, FFNN2 et FFNN3 désignent respectivement le modèle à fonction d'output linéaire, le modèle à fonction d'output non linéaire et le modèle non linéaire.

2.2.3.4 SPECIFICATION ET RESULTATS

A. LE MODELE FFNN1⁹

⁹Matlab 2011a, NNTOOL: nftool, 2011

Il comporte une seule couche cachée (Fine, 1999). Les fonctions d'activation et d'output sont respectivement 'tanh' et 'purelin'. Le taux d'inflation π_t^p au temps t est une fonction au plus des 8 IPC retardés, soit $\pi_t^p = f(\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-p})$; $p=1,8$

Nous choisirons d'abord le modèle représentatif de chacune des huit variantes au moyen du nombre optimal de neurones de sa couche cachée qui sera fixé en fonction des critères de performance, par la méthode 'trial and error'. Nous sélectionnerons, ensuite, la meilleure variante du modèle FFNN1 sur la base de l'erreur de prévisibilité minimale. Elle représentera notre modèle FFNN1 qui sera entraîné par l'algorithme LM sous le logiciel MATLAB 2011.

Sur la base des résultats obtenus par simulation dans matlab et consignés dans le tableau 2.1 ci-dessous, la variante π_t^{8*} [8 :20:1] représentera le modèle FFNN1.

Tableau 2. 1 Performance des variantes du modèle FFNN1 source matlab

VARIANTE	Nombre unités dans couche cachée	MSE	R
π_t^1	10	0.33	0.32
π_t^2	10	0.38	0.32
π_t^3	5	0.41	0.39
π_t^4	5	0.50	0.42
π_t^5	5	0.54	0.40
π_t^6	10	0.46	0.47
π_t^7	5	0.50	0.51
π_t^8	20	0.40	0.58

Les principaux paramètres d'entraînement retenus sont résumés dans le tableau suivant :

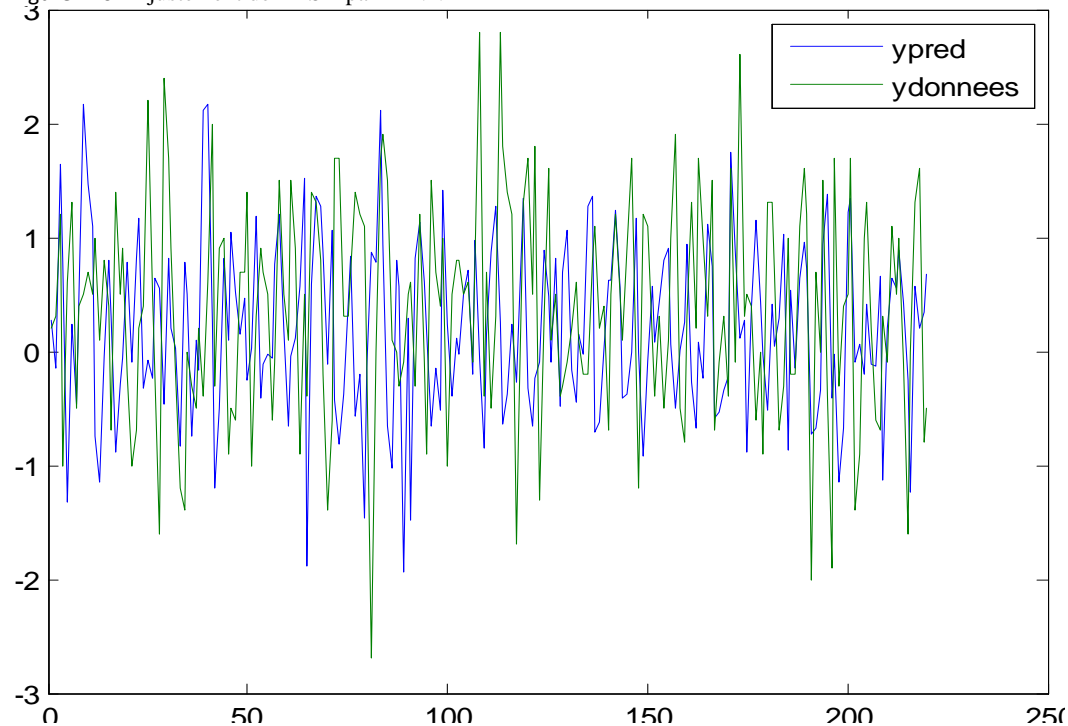
Tableau 2. 2 Paramètres d'entraînement du modèle FFNN1 source matlab

Nombre limite d'époques	1000
Data division	'dividetand'
fonction d'apprentissage	'traingdm'
momentum	0.001
Fonction d'entraînement	'trainlm'
Mse_goal	0
Min_grad	e-005
Mu_inc	10
Max_fail	6
Mu_dec	0.1

i. PREDICTION PAR LE MODELE FFNN1

Le graphe suivant (fig 2.6) illustre la qualité d'ajustement du modèle à la série YTSA.

Figure 2. 6 Ajustement de YTSA par FFNN1



Source : l'auteur

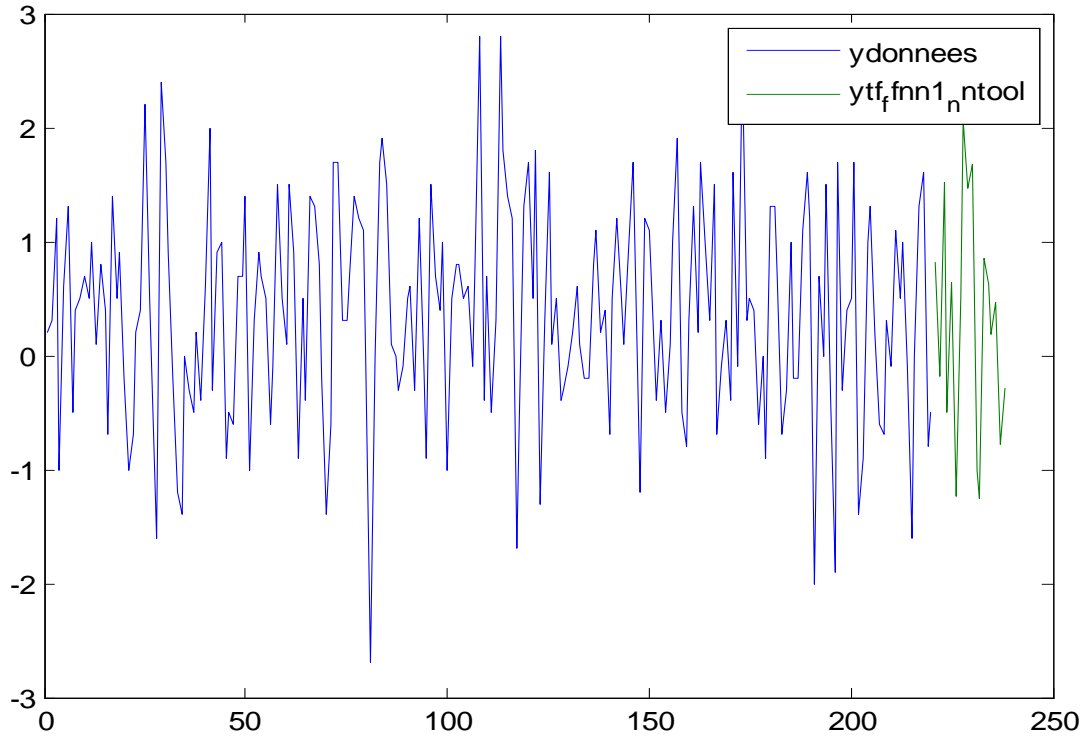
2.2.3.4.1 PREVISION PAR LE MODELE FFNN1

Les IPC prévisionnels janvier 2021 – juin 2022 par ce modèle sont consignés dans le tableau ci-dessous [tab 2.3] et représentés par le graphe [fig 2.7].

Tableau 2. 3 IPC prévisionnels selon FFNN1 Source : l'auteur

Période	janv-21	févr-11	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
\hat{y}_{tsa}	-0.03	-0.42	1.21	-0.36	1.08	-0.82	0.91	1.49	1.06
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
\hat{y}_t	0.82	-0.19	1.52	-0.50	0.65	-1.24	0.52	2.06	1.47
Période	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
\hat{y}_{tsa}	1.46	-0.49	-0.58	-0.01	0.40	-0.13	0.61	-0.34	0.13
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
\hat{y}_t	1.67	-1.02	-1.25	0.85	0.62	0.18	0.47	-0.78	-0.29

Figure 2. 7 Pr evision d'Yt par FFNN1



Source : l'auteur

2.2.3.4.2 LE MODELE FFNN2 ¹⁰

Il comporte une seule couche cach ee (Singh, 2007). Les fonctions d'activation et d'output sont des 'tanh'. Le taux d'inflation π_t^p au temps t est une fonction au plus des 8 IPC retard es, soit

$$\pi_t^p = f(\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-p}); p = \overline{1,8}$$

Le choix de la variante repr esentative de ce mod ele se fera de la m eme mani ere que celle de FFNN1. Les param etres d'entrainement retenus sont identiques   ceux de FFNN1.

Les valeurs MSE, MAE et R obtenues pour les diff erentes variantes du mod ele FFNN2 sont exprim ees dans le tableau ci-dessous.

¹⁰Matlab 7.0, NNTOOL: feed-forward backprop, 2007

Tableau 2. 4 Performance des variantes du modèle FFNN2 source matlab

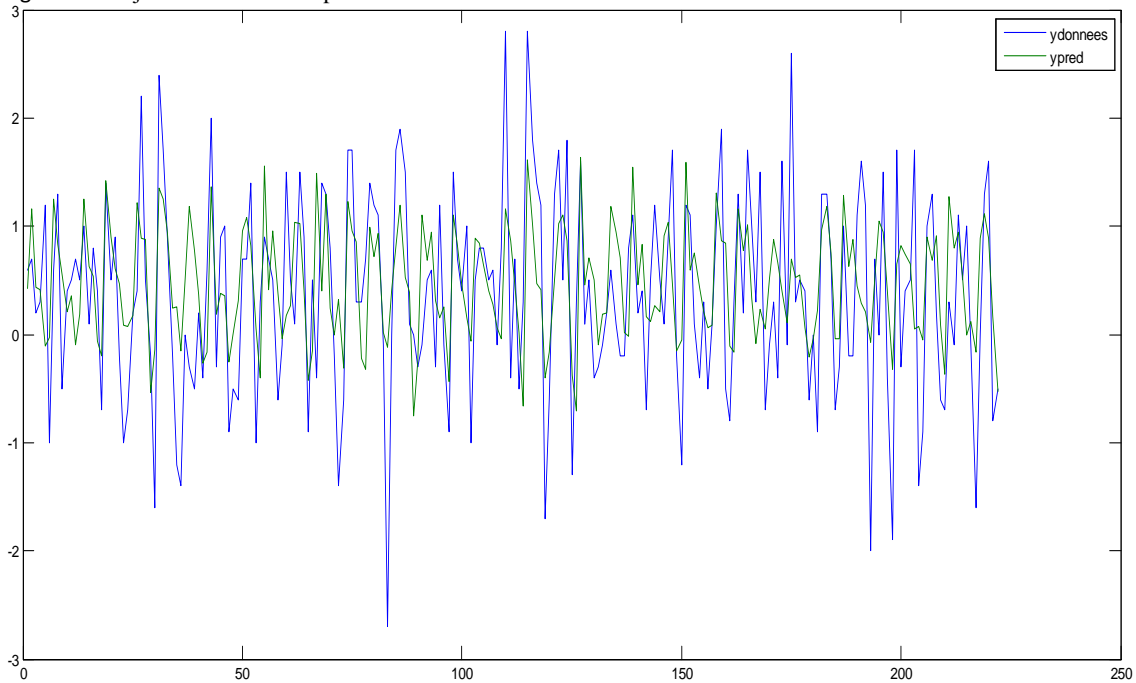
VARIANTE	Nombre unités dans couche cachée	MSE	R
π_t^1	10	0.41	0.28
π_t^2	5	0.31	0.34
π_t^3	30	0.48	0.36
π_t^4	25	0.44	0.36
π_t^5	10	0.39	0.39
π_t^6	15	0.59	0.55
π_t^7	25	0.39	0.38
π_t^8	5	0.50	0.44

Le modèle π_t^6 * [6 :15:1] est retenu pour représenter le modèle FFNN2.

i. PREDICTION PAR LE MODELE FFNN2

Le graphe suivant (fig 2.8) illustre la qualité d'ajustement du modèle à la série YTSA.

Figure 2. 8 Ajustement d'YTSA par FFNN2



Source : l'auteur

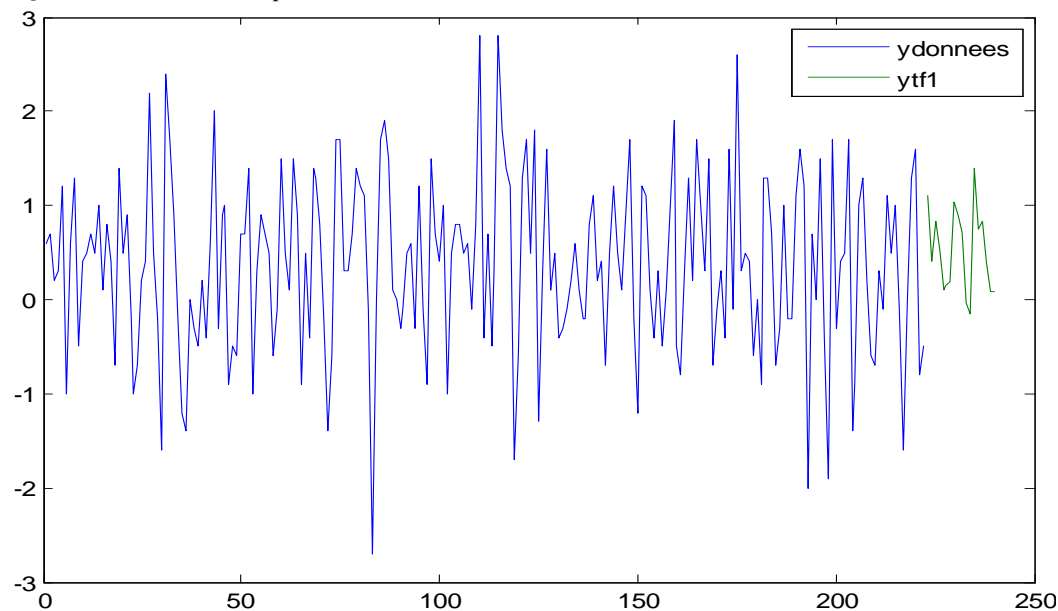
ii. PREVISION PAR LE MODELE FFNN2

Les IPC prévisionnels janvier 2021 – juin 2022 sont rapportés dans le tableau ci-dessous [tab 2.5] et reproduits dans le graphe [fig 2.9].

Tableau 2.5 IPC prévisionnels selon FFNN2 Source : l'auteur

Période	janv-21	févr-21	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
Δy_{tsa}	-0.03	-0.42	1.21	-0.36	1.08	-0.82	0.91	1.49	1.06
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
Δy_t	0.82	-0.19	1.52	-0.49	0.64	-1.24	0.52	2.06	1.47
Période	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
Δy_{tsa}	1.46	-0.49	-0.58	-0.01	0.40	-0.13	0.61	-0.34	0.13
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
Δy_t	1.67	-1.02	-1.24	0.84	0.63	0.18	0.48	-0.78	-0.29

Figure 2.9 Prédiction d' Δy_t par FFNN2



Source : l'auteur

2.2.3.4.3 LE MODELE FFNN3¹¹

Ce modèle ne comporte pas de couche cachée (Garibaldi, 2004). La fonction d'output est la 'tanh'. Il représentera le modèle de forme fonctionnelle non linéaire. Nous sélectionnerons la meilleure variante de ce modèle sur la base de l'erreur de prévisibilité minimale. Elle représentera notre modèle FFNN3 qui sera entraîné par l'algorithme LM sous le logiciel MATLAB.

La variante π_t^6 *[6 :0:1] représentera le modèle FFNN3 compte tenu des résultats générés par MATLAB et consignés dans le tableau suivant :

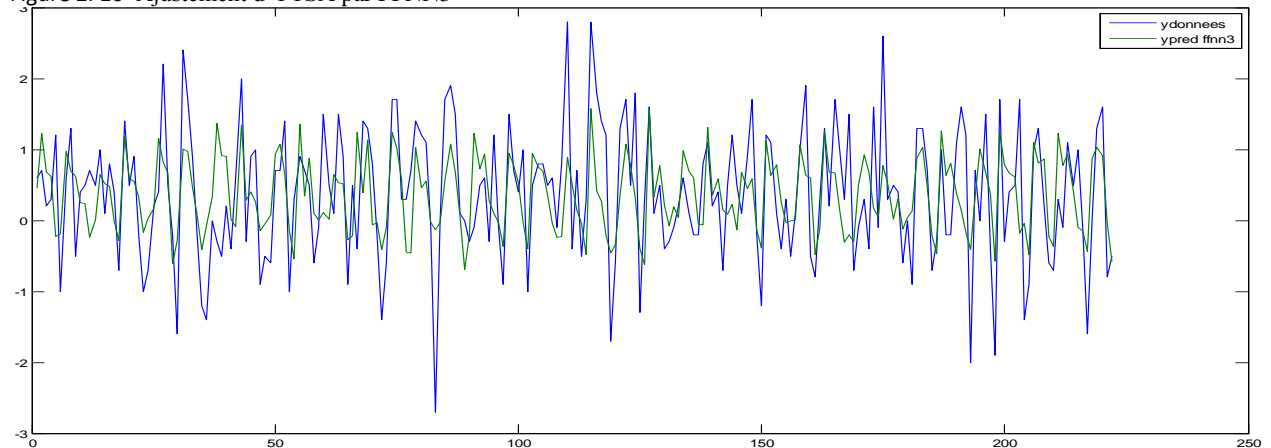
Tableau 2.6 Performance des variantes du modèle FFNN3 source matlab

Variante	MSE_t	MAE	R_t
π_t^1	0.42	0.48	0.32
π_t^2	0.35	0.44	0.43
π_t^3	0.38	0.49	0.43
π_t^4	0.40	0.48	0.52
π_t^5	0.42	0.53	0.45
π_t^6	0.47	0.56	0.65
π_t^7	0.39	0.48	0.34
π_t^8	0.37	0.46	0.39

i. PREDICTION PAR LE MODELE FFNN3

Le graphe ci-dessous [fig 2.10] traduit la qualité d'ajustement du modèle à la série YTSA.

Figure 2.10 Ajustement d'YTSA par FFNN3



Source : l'auteur

¹¹Matlab 7.0, NNTOOL:custom neural network, 2007

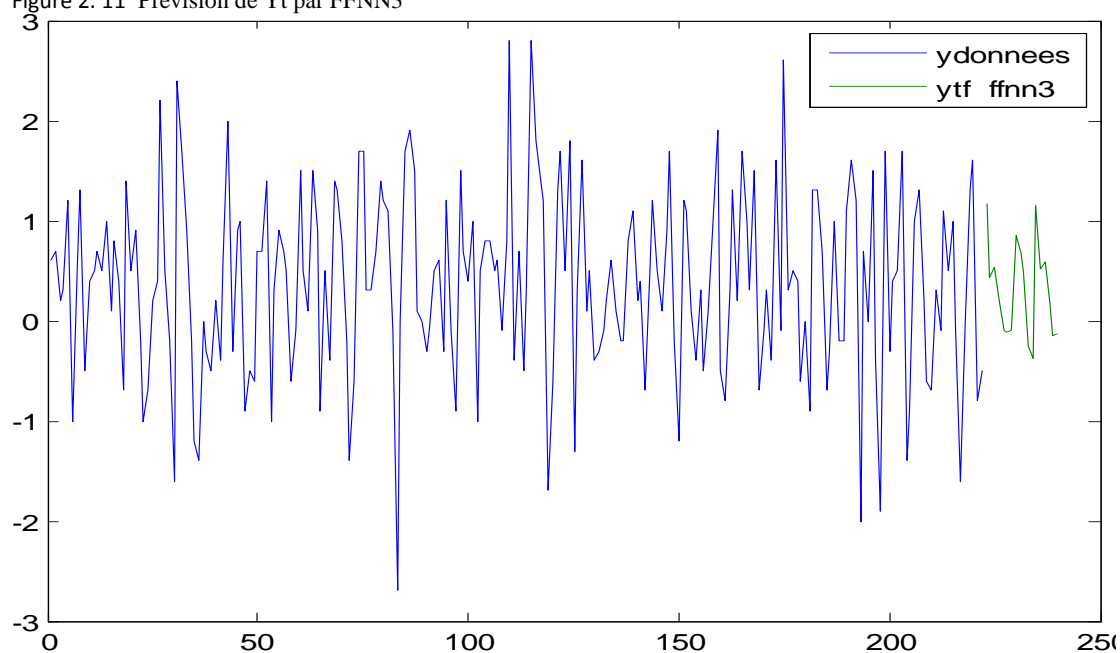
ii. PREVISION PAR LE MODELE FFNN3

Les IPC prévisionnels janvier 2011 – juin 2012 par ce modèle sont consignés dans le tableau ci-dessous [tab 2.7] et représentés par le graphe [fig 2.11].

Tableau 2.7 IPC prévisionnels selon FFNN3 Source : l’auteur

Période	janv-21	févr-21	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
\wedge ytsa	0.31	0.20	0.22	0.33	0.35	0.30	0.28	0.28	0.27
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
\wedge yt	1.16	0.43	0.53	0.20	-0.09	-0.12	-0.11	0.85	0.68
Période	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
\wedge ytsa	0.27	0.28	0.29	0.29	0.29	0.28	0.28	0.28	0.28
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
\wedge yt	0.48	-0.25	-0.37	1.14	0.52	0.59	0.15	-0.16	-0.14

Figure 2. 11 Prévsiion de Yt par FFNN3



Source : l’auteur

2.2.4 PERFORMANCE MODELE FFNN VS SARIMA

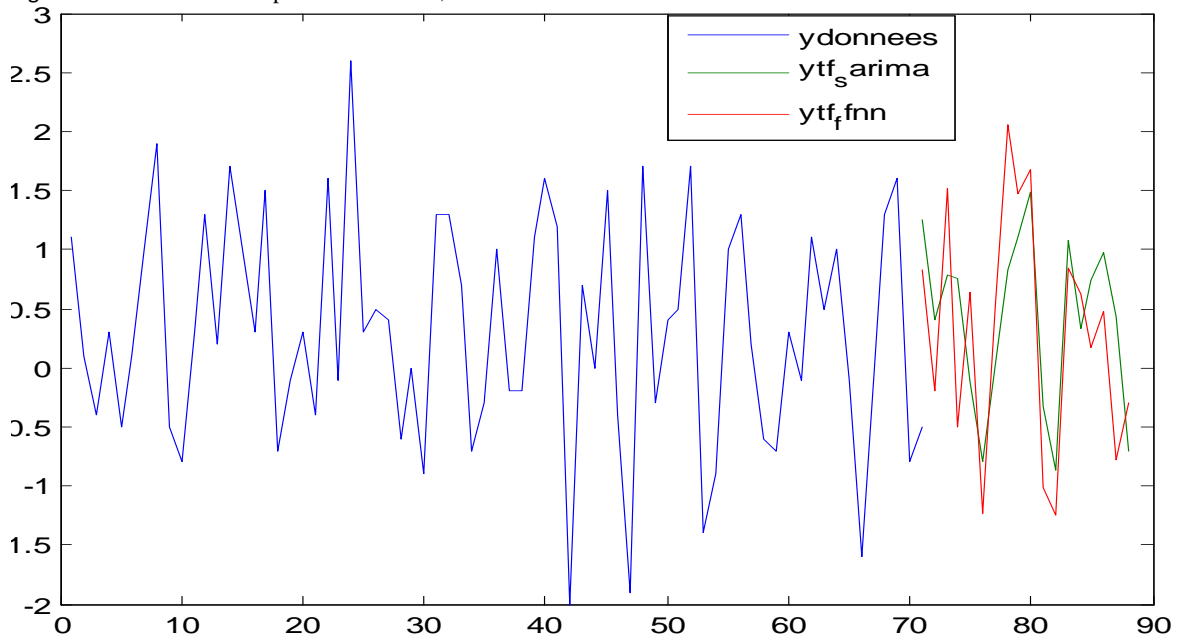
En conclusion, les modèles RNA sont plus performants que le modèle SARIMA. En outre, parmi les modèles neuronaux FFNN, Le modèle FFNN1 est le plus compétitif [tab 2.8].

Tableau 2. 8 Performance comparée du modèle FFNN Source : l'auteur

Variante	RMSE	MAE	R ²
Sarima() ()	0.81	0.65	0.29
FFNN1	0.63	0.48	0.34
FFNN2	0.77	0.61	0.30
FFNN3	0.69	0.57	0.42

Le graphe ci-dessous [fig 2.12] illustre les prévisions du modèle FFNN1 relativement à celles du modèle SARIMA(47,0,4) (12,2,0,2) .

Figure 2. 12 Prévisions comparées SARIMA, FFNN1



Source : l'auteur

L'ajustement des IPC par le modèle FFNN (représenté par la variante FFNN1) se révèle très désynchronisé en amplitude (fig 2.6). Les performances de prévision réalisées (RMSE=0.63 MAE=0.48), bien que meilleures que celles de SARIMA (RMSE=0.81 MAE=0.65), sont encore faibles. Ceci pourrait être dû à la conception purement numérique de ce modèle, Envisageons maintenant les modèles GRNN puis RBFN basés sur une approche statistique.

CH3 PREVISION DES IPC PAR LE MODELE GRNN

Le modèle GRNN constitue le premier modèle neuronal basé sur une approche statistique.

3.1 LES MODELES NEURONAUX STATISTIQUES

Le passage des modèles heuristiques aux modèles statistiques serait dû aux problèmes liés à l'algorithme d'apprentissage BP (problème de minima locaux, temps élevé de computation, ..)

La nouveauté de ces modèles statistiques réside dans le remplacement de la fonction d'activation « sigmoïde » par la fonction de base radiale RBF (ou de Kernel.).

La fonction gaussienne est la fonction de kernel la plus utilisée :

$$K_{\lambda}(x, X) = \exp \frac{-(x - X)^2}{2\lambda^2} \quad (3.1)$$

x: v.a.

X: valeur particulière de x

λ : spread ou écart type

Les principaux modèles neuronaux statistiques sont le modèle neuronal probabiliste PNN, le modèle GRNN et le modèle RBFN.

PNN (Specht, 1990) traite principalement les problèmes de classification et donc les variables catégorielles. Il est plus proche de la méthode « k-nearestneighbour ». Il serait 200.000 fois plus rapide que l'algorithme BP¹².

Le modèle PNN est basé sur la stratégie de décision de Bayes et sur l'estimateur non paramétrique des fonctions de densité à une dimension de Parzen (1962).

L'objet de l'estimation non paramétrique de la fdp consiste, sur la base des données (x_1, x_2, \dots, x_n) d'une variable aléatoire x dont on ne connaît pas la fdp $f(x, \theta_i)$, d'estimer celle-ci par les méthodes classiques : simple, de l'histogramme ou du noyau.

C'est seulement après avoir estimé la fdp que le statisticien s'attellerait à trouver une estimation de la fonction de regression quelle soit linéaire ou non, univariée ou multivariée,...

¹²Specht (D.F.), Probabilistic Neural Networks, *Neural Networks*, 3 :in abstract, 1990

Méthode simple: $f_h^n(x) = \frac{1}{nh} \sum_{i=1}^n w \frac{(x-x_i)}{h}$

$$w(x) = \begin{cases} 1 & \text{si } x \in \left[-\frac{1}{2}, \frac{1}{2}\right] \\ 0 & \text{sinon} \end{cases} \quad [3.2]$$

Méthode de l'histogramme: $f_h^n(x) = \frac{1}{h} \sum_{k=1}^n P(A_k) 1_{A_k}(x)$

A_k : classe k $P(A_k)$: probabilité de classe k

Méthode du noyau: $f_n(x) = \int_{-\infty}^{\infty} \frac{1}{h} \frac{k(x-y)}{h} dF_n(y) = \frac{1}{nh} \sum_{j=1}^n \frac{k(x-X_j)}{h}$ [3.3]

$$/ K(y) = \begin{cases} 1/2 & |y| \leq 1 \\ 0 & |y| \geq 1 \end{cases}$$

$h = h(n)$

$\lim_{n \rightarrow \infty} h(n) = 0$

$\lim_{n \rightarrow \infty} E[f_n(x)] = f(x)$

$n =$ nombre d'observations

$F_n(\cdot)$: fonction de distribution empirique

$k(x)$: kernel function of x

Étendu au cas multidimensionnel par Cacoullos, ce dernier estimateur devient [3.4]:

$$f_n(x) = \int_{-\infty}^{\infty} \frac{1}{h^p(n)} \frac{k(x-X_j)}{h(n)} dF_n(y) = \frac{1}{nh^p(n)} \sum_{j=1}^n \frac{k(x-X_j)}{h(n)} \quad [3.4]$$

$h^p(n) = h_1 \cdot h_2 \dots h_p$ avec

$h_i = h_i(n) = cste > 0 \quad \forall i = \overline{1, p} \quad / \quad \lim_{n \rightarrow \infty} h(n) = 0$ quand $n \rightarrow \infty$

X_j : observation j de l'échantillon aléatoire ; x : v.a.

$f_n(x)$: fonction de densité empirique

$\lim_{n \rightarrow \infty} E[f_n(x)] = f(x)$

$\lim_{n \rightarrow \infty} V[f_n(x)] = 0$

$n =$ nombre d'observations

Ces estimateurs sont robustes et sans biais (Cacoullos, 1966).

Les modèles GRNN et RBFN s'appliquent surtout à la prévision des variables continues.

3.2 LE MODELE GRNN

Le modèle GRNN (Specht, 1991) est un réseau à fonction de base radiale qui ne requiert pas d'entraînement itératif. C'est un approximateur de fonction. Il ne présuppose aucune forme fonctionnelle de celle-ci mais plutôt l'exprime comme une fonction de densité de probabilité f.d.p. qui sera déterminée empiriquement à partir des données (X^i, Y^i) par l'estimateur non paramétrique de Parzen-Rosenblatt(1962) étendu au cas multidimensionnel par Cacoullos.

En considérant $f(x, y)$, la f.d.p. conjointe continue entre 2 variables aléatoires (v.a.) i.i.d. x et y . Etant donné X une valeur particulière de x , la régression de y sur X basée sur l'espérance conditionnelle (Dai,2010) s'exprime par :

$$E(y/X) = \frac{\int_{-\infty}^{+\infty} y f(X, y) dy}{\int_{-\infty}^{+\infty} f(X, y) dy} \quad (3.5)$$

$f(x,y)$, souvent inconnue, est estimée par Parzen-Rosenblatt à partir des données (X^i, Y^i) .

Dans le cas d'une fonction de Kernel gaussienne, celle-ci est estimée à :

$$\hat{f}(X, y) = \frac{1}{(2\pi)^{\frac{p+1}{2}} \sigma^{p+1}} \frac{1}{n} \sum_{i=1}^n \exp\left[-\frac{D_i^2}{2\sigma^2}\right] \exp\left[-\frac{(y - y^i)^2}{2\sigma^2}\right] \quad / p = \dim(x) \quad (3.6)$$

$$D_i^2 = (X - X^i)^T (X - X^i)$$

En substituant $\hat{f}(x, y)$ à $f(x,y)$ dans (3.5), nous obtenons :

$$\hat{y} = \frac{\sum_{i=1}^n \exp\left[-\frac{(X - X^i)'(X - X^i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} y \exp\left[-\frac{(y - y^i)^2}{2\sigma^2}\right] dy}{\sum_{i=1}^n \exp\left[-\frac{(X - X^i)'(X - X^i)}{2\sigma^2}\right] \int_{-\infty}^{\infty} \exp\left[-\frac{(y - y^i)^2}{2\sigma^2}\right] dy} \quad (3.7)$$

Puis après calculs (Hanan ,2010),

$$\hat{y}(X) = \frac{\sum_i y^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad \text{représentant la valeur estimée de } y \text{ sachant un vecteur } X. \quad (3.8)$$

$\sigma = \sigma(n)$ est choisi comme une fonction décroissante de n :

$$\lim_{x \rightarrow \infty} \sigma(n) = 0$$

$$\lim_{x \rightarrow \infty} n\sigma^2(n) = 0$$

L'expression (3.8) peut être remplacée par une expression de Parzen(1962) plus pratique:

$$\hat{y}(X) = \frac{\sum_i y^i \exp\left(-\frac{C_i}{\sigma}\right)}{\sum_i \exp\left(-\frac{C_i}{\sigma}\right)} \quad (3.9)$$

$$C_i = \sum_{j=1}^n |X_j - X_j^i|$$

Le choix de la valeur optimale de « σ » (ou spread) est réalisé par diverses méthodes :

Holdout, Wiggle , ... (Baeuer, 1995)

3.3 STRUCTURE

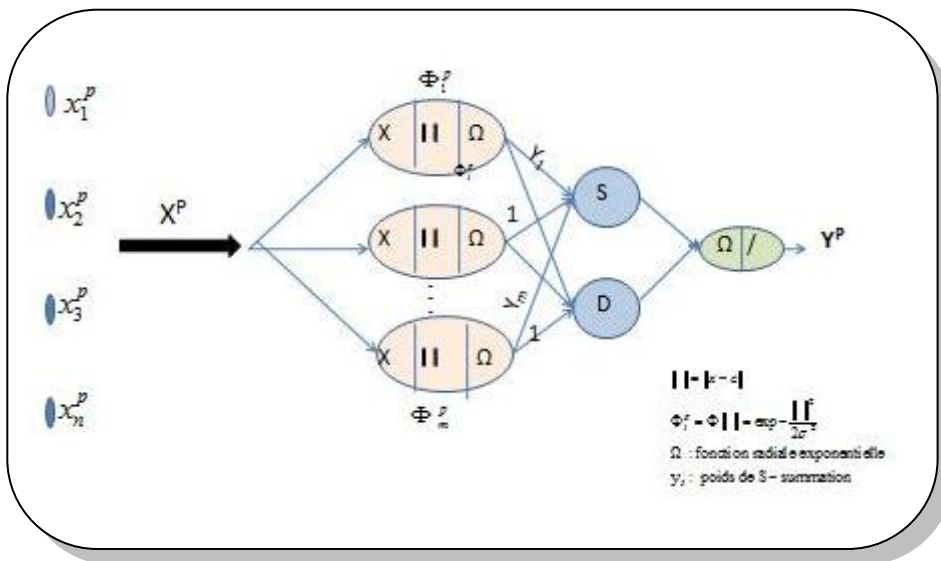
Le modèle GRNN, dont la structure¹³est représentée ci-dessous [fig 3.1], comprend 4 couches : la couche d'inputs reçoit le vecteur input X et le transfère à une couche d'unités radiales dénommée « pattern ». Chaque unité de celle-ci comprend un centred'entraînement X^i .

Elle calcule la norme $\|X-X^i\| = D_i = \sqrt{(x-x_i)^T(x-x_i)}$ puis produit, dans le cas d'une fonction Φ gaussienne, l'output $\Phi(D) = \exp\left(-\frac{D^2}{2\sigma^2}\right)$

La couche « summation » contient 2 unités: le « S-summation » neurone calcule la somme des outputs pondérées par les y^i , le « D-summation » calcule la simple somme des outputs.

Enfin, la couche d'output calcule le quotient $\frac{\text{output du «S - summation »}}{\text{output du «D - summation »}}$.

Figure 3. 1 Structure GRNN [k:m:2:1]



Source : l'auteur

¹³Baeuer (M.M.), *General regression neural network for technical use*, p.46,1995

3.4 PREVISION

3.4.1 DONNEES

Considérons le modèle GRNN pour prévoir les IPC en Algérie (π_t). Nous reprenons les mêmes données prétraitées et hypothèses énoncées dans le modèle FFNN.

3.4.2 STRUCTURE DU RESEAU

L'ensemble des variables explicatives $\{\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}\}$ constituent le vecteur input $X^k = \pi_t^k$, l'output $\hat{\pi}_t$ constitue la valeur prédite de la valeur actuelle (cible) du taux d'inflation π_t .

Chaque unité de la couche « pattern » est représentée par un vecteur de l'ensemble des vecteurs inputs $\{X^k \in R^k; k \leq 8\}$. La fonction radiale est gaussienne. La couche de sortie

comprend un seul neurone qui calcule le quotient $\frac{\text{output du «S – summation »}}{\text{output du «D – summation »}}$

3.4.3 METHODOLOGIE DE PREVISION

Nous appliquerons la méthodologie utilisée dans le modèle FFNN. Les entraînements s'effectueront à l'aide MATLAB 2011.

3.4.4 SPECIFICATION ET RESULTATS

Le taux d'inflation π_t^p au temps t est une fonction au plus des 8 taux d'inflation retardés soit

$$\pi_t^p = f(\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-p}); \quad p = \overline{1, 8}$$

3.4.4.1 CHOIX DE LA VARIANTE OPTIMALE

Nous choisirons d'abord la structure qui représentera chaque variante sur la base du choix de la valeur optimale de « σ » (ou spread) puis nous sélectionnerons la variante représentative du modèle GRNN sur la base des critères de performance retenus.

Sur la base des résultats obtenus et consignés dans le tableau ci-dessous, la variante π_t^{8*} [8 :148 :2:1] avec ($\sigma^*=0.1$) représentera le modèle GRNN.

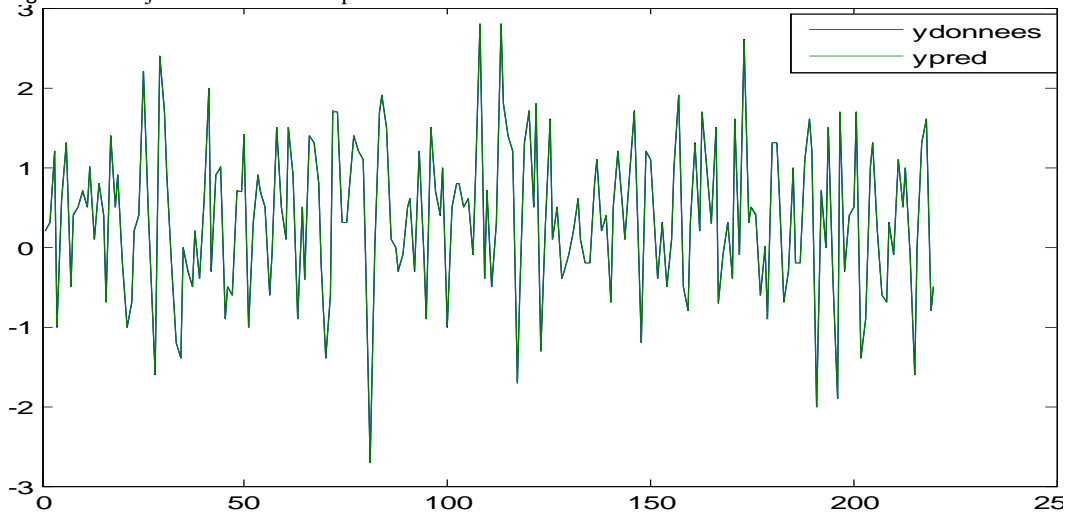
Tableau 3.1 Performance des variantes du modèle GRNNsource matlab

VARIANTE	spread	MSE	MAE	R ²
π_t^1	0.1	0.60	0.63	0.29
π_t^2	0.1	0.22	0.34	0.83
π_t^3	0.1	0.0175	0.048	0.9
π_t^4	0.1	6.00e-4	0.0041	0.9
π_t^5	0.1	2.53e-8	1.94e-5	0.9
π_t^6	0.1	7.66e-12	2.62e-7	0.9
π_t^7	0.1	8.92e-22	3.66e-12	0.9
π_t^8	0.1	4.88e-25	6.61e-14	0.9

3.4.4.2 PREDICTION PAR LE MODELE GRNN

La figure suivante [fig 3.2] illustre l'excellente qualité d'ajustement du modèle à la série Yt.

Figure 3.2 Ajustement d'YTSA par GRNN



Source : l'auteur

3.4.4.3 PREVISION

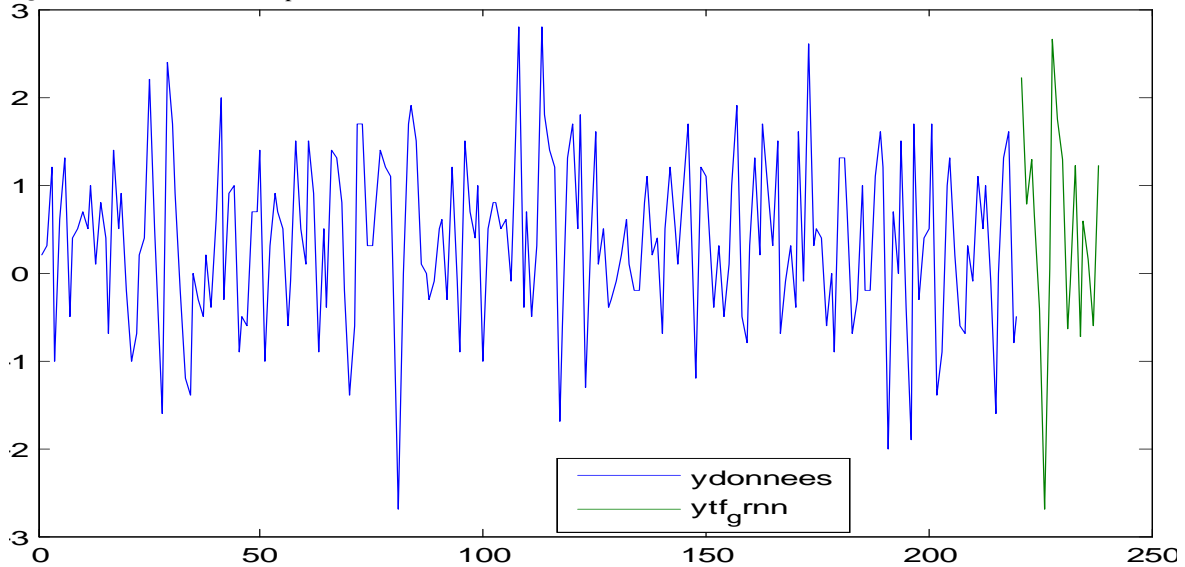
Le tableau ci-dessous (tab 3.2) donne les prévisions des IPC yt par ce modèle.

Tableau 3.2 IPC prévisionnels selon GRNNSource : l'auteur

Période	janv-21	févr-21	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
$\wedge ytsa$	1.36	0.55	0.97	0.79	0.03	-2.26	0.42	2.09	1.33
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
$\wedge yt$	2.21	0.78	1.28	0.66	-0.41	-2.68	0.03	2.66	1.74
Période	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
$\wedge ytsa$	1.09	-0.11	0.53	0.36	-0.95	0.27	0.29	-0.17	1.64
factors	0.40	1.00	-0.90	0.85	0.23	0.31	-0.13	-0.44	-0.42
$\wedge yt$	1.49	0.89	-0.37	1.21	-0.72	0.58	0.16	-0.61	1.22

La figure ci-dessous [fig.33] retrace les prévisions de la série y_t par ce modèle .

Figure 3. 3 Prévsiion de Y_t par GRNN



Source : l'auteur

3.5 PERFORMANCE COMPAREE DU MODELE GRNN

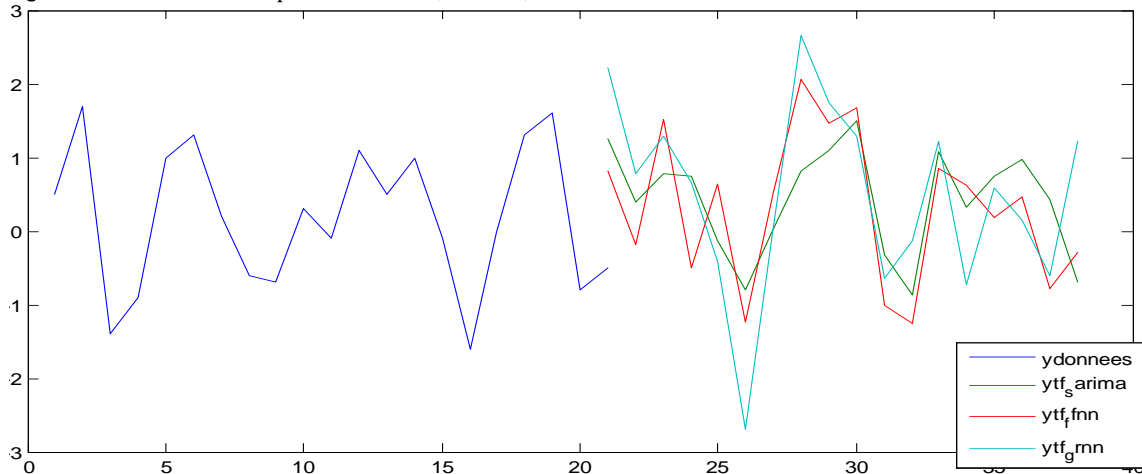
En conclusion, le modèle GRNN est plus performant relativement à la meilleure variante du modèle FFNN à savoir FFNN1 [tab 3.3].

Tableau 3. 3 Performance comparée de GRNNSource : l'auteur

Variante	RMSE	MAE	R ²
Sarima()()	0.81	0.65	0.29
FFNN1	0.63	0.48	0.34
FFNN2	0.77	0.61	0.30
FFNN3	0.69	0.57	0.42
GRNN	6.98e-13	6.66-14	0.9

Le graphe ci-dessous reproduit les prévisions du modèle GRNN relativement à celles des modèles SARIMA et FFNN1.

Figure 3. 4 Prévisions comparées SARIMA, FFNN1,GRNN



Source : l'auteur

L'ajustement de ce modèle est excellent (fig 3.2). Les performances de prévision sont très bonnes ($RMSE=6.98e-13$, $MAE=6.66e-14$). Ce modèle neuronal basé sur une approche statistique est de loin meilleur relativement au modèle neuronal « numérique » FFNN. Voyons maintenant si cette hypothèse se confirme en étudiant un autre modèle de la même classe, le modèle RBFN.

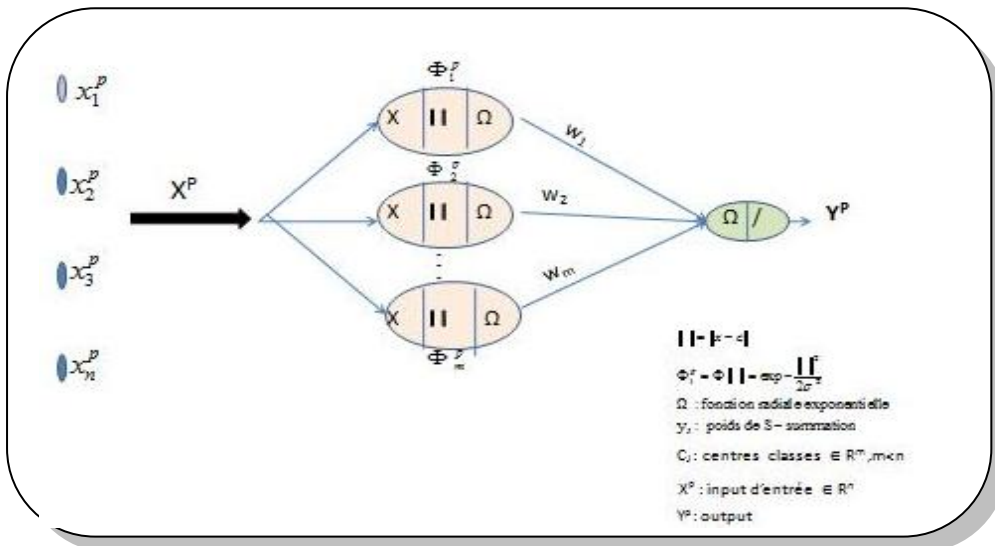
CH4 PREVISION DES IPC PAR LE MODELE RBFN

Le modèle RBFN consacre le passage des modèles neuronaux à fonction de base « sigmoïde » aux modèles à fonction de base radiale généralement gaussienne.

4.1 LE MODELE RBFN

De structure comparable à celle du FFNN, le modèle RBFN est un approximateur universel statique et feedforward mais contrairement au FFNN, il ne comporte qu'une seule couche cachée ; son processus d'apprentissage se limite aux neurones (poids) de la couche d'output ; ce qui lui confère une rapidité plus grande (Howlet, 2001). La figure 4.1 ci-dessous présente la structure de base du modèle RBFN gaussien.

Figure 4. 1 Structure RBFN [k :m:1]



Source : l'auteur

Soit un échantillon de N vecteurs inputs à étudier $\{X^t \in R^n ; t = \overline{1, N}\}$. Dans sa forme générale « regularization network »¹⁴, le RBFN crée autant de vecteurs centres C_j (et donc d'unités cachées H) que d'inputs à étudier, soient $C_j = [c_1, \dots, c_n]$; $j = \overline{1, N}$ (et $H = H_1, \dots, H_N$).

¹⁴Satish (K.), *Neural Networks: a classroom approach*, p.323, 2004

Quand N s'agrandit, la réduction du nombre de centres C_j s'impose (et par suite le nombre de couches cachées H) par le rassemblement des inputs en m vecteurs ($m < N$) au moyen des « clustering methods ». La recherche des C_j constitue la 1^{ère} tâche de ce modèle.

Par suite, chaque unité de la couche cachée H s'attribue un vecteur centre C_j et calcule la distance entre le vecteur input X^l et le vecteur centre C_j . Dans le cas d'une norme euclidienne, nous aurons :

$$\text{net}_j = \|X - C_j\| = \left[\sum_{i=1}^n (x_i - c_{ij})^2 \right]^{1/2}; \text{net}_j \text{ est ensuite transformé par}$$

une fonction d'activation de type « shappedfunction »: Gaussian, thin plate spline,.... Dans le cas de la fonction Gaussienne, nous aurons: $\Phi(\text{net}_j) = G(\text{net}_j) = G(\|X - C_j\|) = \text{EXP} \left[-\|x - c_j\|^2 / 2\sigma^2 \right]$.

$G(\text{net}_j)$ est appelée RBF. σ est fixé à $d_{\max} / \sqrt{2m}^{15}$. La recherche du spread σ constitue la 2^{ème} tâche.

Enfin, Les outputs de l'unité de la couche cachée sont alors multipliés par leurs poids d'outputs et transformés par la fonction d'output « purelin » générant les outputs finals:

$$y_h = \sum_j w_{kj} G(\|X - C_j\|). \text{ Le calcul de l'erreur de prévisibilité "e" amorce alors}$$

l'apprentissage qui se limite à la couche d'output et dont l'objectif est de trouver les poids d'output W_{kj} qui minimisent l'erreur. Ceci constitue la 3^{ème} tâche.

4.2 REGLES D'APPRENTISSAGE

Trois stratégies d'apprentissage sont utilisées dans le modèle RBFN : la « pseudo inverse technique »¹⁶, le « gradient descent learning »¹⁷ et l'« hybrid learning ».

¹⁵d= distance euclidienne= $\|x - c_i\|$

¹⁶ Méthode « offline »

¹⁷ Méthode « online »

4.2.1 PSEUDO-INVERSE TECHNIQUE

On choisit¹⁸, au hasard, parmi les N vecteurs inputs $X^t; X \in R^n$, m vecteurs centres C_j ($m \leq N$). En outre, on fixe, dans le cas d'une fonction gaussienne $G(\|x - c_i\|) = \exp(-\frac{1}{2\sigma^2}\|x - c_i\|^2)$, σ à $\frac{d_{\max}}{\sqrt{2m}}$. La fonction de base radiale devient $G(\|x - c_i\|) = \exp(-\frac{m}{d^2}\|x - c_i\|^2)$. Le training, une fois les centres C_j fixés, se limite alors à l'ajustement des poids d'output W_{kj} de la couche cachée par la méthode pseudo inverse (Chow, 2007).

Etant donné un échantillon de vecteurs inputs X^t et un vecteur cible Y^d , l'objectif est de déterminer les poids W_{kj} tels que $SSE = 1/2 \sum_{i=1}^k (y_i^d - \hat{y}_i)^2$ soit minimale. Pour un vecteur X^t donné, l'équation à résoudre est :

$$\sum_{j=1}^k \phi_j w_j = y^d \text{ soit } \phi^T w = y^d. \tag{4.1}$$

$$\phi = [\phi_1^t \ \phi_2^t \ \phi_3^t \ \dots \ \phi_m^t \ 1]^T ; w = [w_1 \ w_2 \ \dots \ w_m, \theta]^T ; y^d = y_t^d$$

Pour les N vecteurs $\{X^1, X^2, \dots, X^N\}$, le système d'équations est :

$$\begin{pmatrix} \phi_1^1 & \phi_2^1 & \dots & \phi_m^1 & 1 \\ \phi_1^2 & \phi_2^2 & \dots & \phi_m^2 & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi_1^N & \phi_2^N & \dots & \phi_m^N & 1 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \\ \theta \end{pmatrix} = \begin{pmatrix} y_1^d \\ y_2^d \\ \vdots \\ y_N^d \end{pmatrix} \tag{4.2}$$

Soit $\Phi .W = Y$ d'où $W = \Phi^{-1} Y$.

La matrice Φ n'étant pas carrée, elle n'est pas inversible ; la solution du système d'équations passe par la technique du pseudo inverse voire la décomposition en valeurs singulière (SVD) :

¹⁸Chow(T.W.S.) & Cho(S.Y.), *Neural network and computing: Learning algorithms and applications*, p.152, 2007

Si l'on considère l'équation (4.2) transformée: $\Phi^T \Phi .W = \Phi^T Y$, la matrice $\Phi^T \Phi$ étant carrée symétrique, alors $W = (\Phi^T \Phi)^{-1} .\Phi^T Y = \Phi^+ Y$. Si la matrice de variance covariance $(\Phi^T \Phi)$ n'est pas inversible, on décompose Φ en valeurs singulières, c.à.d. on diagonalise Φ tel que :

$$\begin{aligned}
 U^T \Phi V &= \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_k) \\
 \sigma_1 &\geq \sigma_2 \geq \dots \geq \sigma_k > 0 \\
 k &= \min (N, m+1) \\
 U &= [\vec{u}_1, \dots, \vec{u}_N] \\
 V &= [\vec{v}_1, \dots, \vec{v}_m, \vec{v}_{m+1}] \\
 \vec{u} &= \text{vect. propres de } \phi\phi^t \\
 \vec{v} &= \text{vect. propres de } \phi^t\phi
 \end{aligned}
 \tag{4.3}$$

On en déduit :

$$\begin{aligned}
 \Phi^+ &= V \Sigma^+ U^T \\
 \Sigma^+ &\text{ est une matrice } (m, N) \\
 \Sigma^+ &= \text{diag} (1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_k, 0, \dots, 0)
 \end{aligned}
 \tag{4.4}$$

4.2.2 GRADIENT DESCENT LEARNING

L'algorithme d'apprentissage « gradient descent » permet d'ajuster les 3 paramètres C_j , σ et w_j simultanément. Au temps n , nous aurons¹⁹ :

$$\begin{aligned}
 \Delta W_{kj}(n) &= \eta_w \frac{\partial E(n)}{\partial w_{kj}(n)} = \dots = \eta_w \sum_{k=1}^N e_k(n) G(\|x - c_j(n)\|) \\
 \Delta C_j &= \eta_c \frac{\partial E}{\partial c_j} = \dots = 2 \eta_c w_{kj} \sum_k e_k G'(\|x - c_j\|) \sigma_j^{-2} (x - c_j)
 \end{aligned}
 \tag{4.5}$$

¹⁹ Chow (T.W.S.), *op.cit.*, p.154

$$\Delta\sigma_j^{-1} = \eta_\sigma \frac{\partial E}{\partial \sigma_j^{-1}} = \dots = -\eta_\sigma w_{kj} \sum_k e_k G'(\|x - c_j\|) (x - c_j)(x - c_j)^t$$

La fonction de coût, n'étant pas convexe vis-à-vis des paramètres C_i et σ , fait que la procédure « gradient descent » souffre du problème du minima local (Deepa, 2006).

4.2.3 HYBRID LEARNING

Elle consiste en un entraînement non supervisé des paramètres C_j et σ par les « clustering methods » puis supervisé des W_{kj} par une des méthodes: moindre carrés des erreurs LMS, moindres carrés récursives RLS , « gradient descent »,...(Gupta, 2003).

4.2.3.1 CHOIX DES 'C_j' PAR L'ALGORITHME « K-MEANS »

Cette méthode²⁰ partitionne les X en m sous groupes S_j en minimisant la fonction de perte suivante :

$$I = \sum_{j=1}^m \sum_{i=1}^N a_{ij} \|X_i - C_j\|^2 \rightarrow \min_{c_j} \quad (4.6)$$

- 1- on démarre avec un échantillon aléatoire $C_j^{(1)}$ tiré à partir de l'ensemble des X
- 2- on assigne chaque donnée x_i au groupe-centre le plus proche par le choix des a_{ij} pour C_j

fixés, soient : $a_{ij} = \begin{cases} 1 & \text{si } j = \operatorname{argmin} \|X_i - C_j\|^2 \\ 0 & \text{sinon} \end{cases}$

- 3- on calcule le centre du nouveau groupe $C_j^{(2)}$ relativement à ses nouvelles données \underline{X}_i :

$$C_j^{(2)} = \frac{\sum_i a_{ij} \underline{X}_i}{\sum_i a_{ij}}$$

- 4- on répète 1 et 2 jusqu'à convergence.

²⁰Nelles (O.), *Non linear system identification: from classical approaches to neural networks and fuzzy models*, p.143, 2001

4.2.3.2 CHOIX DES ‘ Σ_j ’ PAR L’ALGORITHME « K-NEAREST NEIGHBOURS »

La valeur ‘ σ_j ’ est choisie égale à la racine de la moyenne des carrés des distances de chaque centre C_j par rapport à ses K voisins C_i ²¹ :

$$\sigma_j = \sqrt{\frac{1}{k} \sum_{i=1}^k (C_j - C_i)^2} \quad (4.7)$$

4.2.3.3 CHOIX DES W_{kj}^* PAR L’ALGORITHME « STEEPEST DESCENT »

Une fois C_j et σ_j calculés, les activations $\phi_j(x^p, c_j, \sigma_j)$ sont fixées; il ne reste qu’à chercher itérativement les poids w_{kj} qui minimisent la somme des carrés des erreurs d’output :

$$E = \frac{1}{2} \sum_t \sum_k (y_k(x^t) - t_k^t)^2 = \frac{1}{2} \sum_t \sum_k \left(\sum_j w_{kj} \phi_j(x^t, c_j, \sigma_j) - t_k^t \right)^2 \quad (4.8)$$

Au minimum de E , les gradients par rapport à tous les poids w_{kj} seront nuls:

$$\frac{\partial E}{\partial w_{kj}} = \sum_t \sum_j (w_{kj} \phi_j(x^t) - t_k^t) \phi_j(x^t) = 0, \text{ la solution s'obtient par la résolution de ce système}$$

(Chong, 2011).

4.3. PREVISION

4.3.1 DONNEES

Considérons le modèle RBFN pour prévoir les IPC en Algérie (π_t). Nous reprenons les mêmes données prétraitées et hypothèses énoncées dans le modèle FFNN.

4.3.2 ARCHITECTURE

L’ensemble des variables explicatives $\{\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}\}$ constituent le vecteur input X^k , l’output $\hat{\pi}_t$ constitue la valeur prédite de la valeur actuelle (cible) du taux d’inflation π_t , le

²¹Howlet (R.J.), Jain (L.C.), *Radial Basis Function Networks 2: new advances in design*, Physica-Verlag, 11, 2001

nombre des centres C_j de la couche cachée est choisi par la méthode 'Rols' de matlab, après avoir saisi une valeur de σ entre 0 et 1. La fonction radiale est gaussienne. La couche de sortie comprend un seul neurone et sa fonction de transfert est «purelin».

4.3.3 METHODOLOGIE DE PREVISION

Nous appliquerons la méthodologie utilisée dans les modèles FFNN et GRNN. RBFN désigne le modèle RBFN à fonction radiale gaussienne. Les entrainements seront effectués à l'aide MATLAB 2011.

4.3.4 SPECIFICATION ET RESULTATS

4.3.4.1 CHOIX DE LA VARIANTE OPTIMALE

Le taux d'inflation π_t^p au temps t est une fonction au plus des 8 taux d'inflation retardés soit $\pi_t^p = f(\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-p})$; $p=1,8$

Nous choisirons d'abord la structure qui représentera chacune des huit variantes π_t^p sur la base du choix de la valeur optimale du spread « σ » puis nous sélectionnerons la variante représentative du modèle RBFN sur la base de l'erreur de prévisibilité. Elle représentera notre modèle RBFN qui sera entraîné par l'algorithme LM sous le logiciel MATLAB 2011.

Sur la base des résultats obtenus dans le tableau ci-dessous, la variante π_t^{7*} [7 :221 :1] ($\sigma=0.5$) représentera le modèle RBFN.

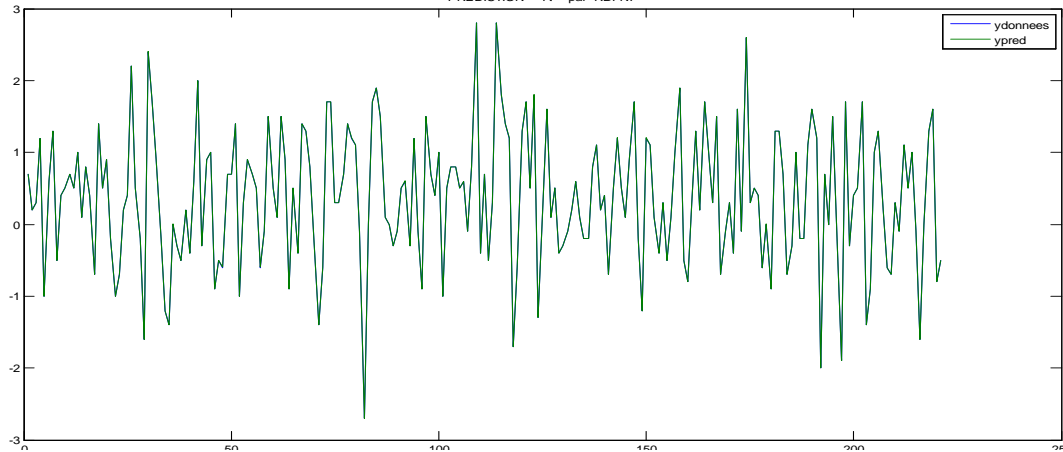
Tableau 4.1 Performance des variantes du modèle RBFN source matlab

VARIANTE	Nombre unités dans couche cachée	Spread σ	MSE	MAE	R ²
π_t^1	227	0.1	0.35	0.45	0.67
π_t^2	226	0.1	0.0009	0.0035	0.9
π_t^3	225	0.1	0.37 e-12	1.43e-6	0.9
π_t^4	224	0.2	5.67e-30	7.64e-16	0.9
π_t^5	223	0.2	1.10e-30	1.55e-15	0.9
π_t^6	222	0.4	9.08e-31	6.81e-16	0.9
π_t^7	221	0.5	7.21e-31	6.12e-16	0.9
π_t^8	220	1.1	8.79e-31	6.16e-16	0.9

4.3.4.2 PREDICTION PAR LE MODELE RBFN

Le graphe suivant (fig 4.2) illustre l'excellente qualité d'ajustement du modèle à la série YTSA.

Figure 4. 2 Ajustement de YTSA par RBFN



Source : l'auteur

4.3.4.3 PREVISION

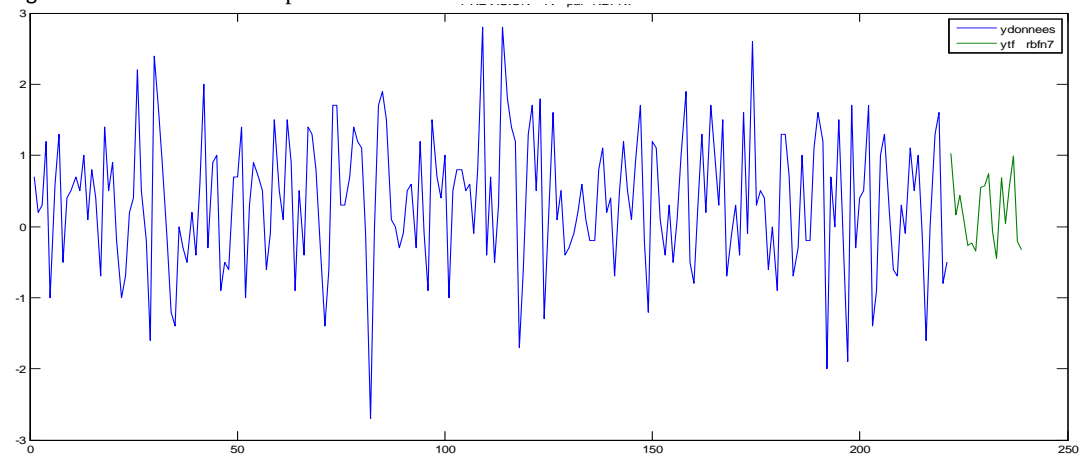
Le tableau ci-dessous (tab 4.2) donne les prévisions des IPC_{y_t} par ce modèle.

Tableau 4. 2 IPC prévisionnels par RBFN Source : l'auteur

Période	janv-21	févr-21	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
\hat{y}_{tsa}	0.16	-0.06	0.12	0.23	0.17	0.19	0.04	-0.02	0.16
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
\hat{y}_t	1.01	0.17	0.43	0.10	-0.27	-0.23	-0.35	0.55	0.57
Période	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
\hat{y}_{tsa}	0.53	0.45	0.22	-0.17	-0.19	0.26	1.12	0.23	0.10
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
\hat{y}_t	0.74	-0.08	-0.44	0.68	0.04	0.57	0.99	-0.21	-0.32

Le graphe suivant [fig 4.3] exposera la série des taux d'inflation prévisionnels ' ytf ' par ce modèle.

Figure 4. 3 Prédiction de Yt par RBFN



Source : l'auteur

4.4 PERFORMANCE DU MODELE RBFN

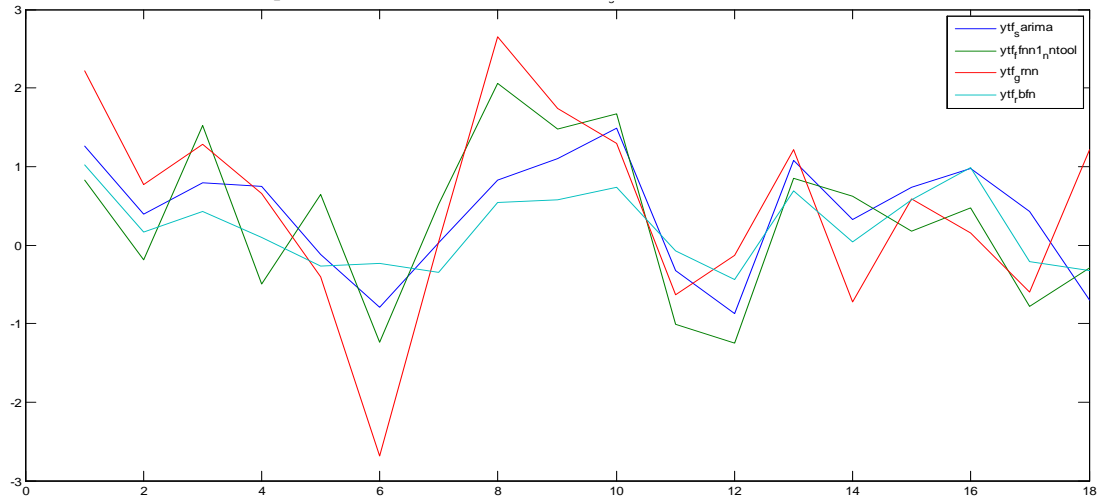
Le modèle RBFN est plus performant que le modèle GRNN [tab 4.3]

Tableau 4. 3 performance comparée du RBFN Source : l'auteur

Variante	RMSE	MAE	R ²
Sarima() ()	0.81	0.65	0.29
FFN1	0.63	0.48	0.34
GRNN	6.98e-13	6.66-14	0.9
RBFN	8.49e-16	6.11e-16	0.9

Le graphe 4.4 reproduit les prévisions du modèle RBFN relativement aux modèles précédents SARIMA, FFNN1 et GRNN.

Figure 4. 4 Prévisions comparées SARIMA, FFNN1, GRNN, RBFN



Source : l'auteur

La supériorité relative des modèles neuronaux « statistiques » est ainsi confirmée. Toutefois, les modèles statiques (FFNN, GRNN, RBFN) sont inadaptés aux systèmes dynamiques (à variables retardées). Par conséquent, explorons présentement les modèles neuronaux dynamiques (TDNN, NNAR, ...).

CH5 PREVISION DES IPC PAR LEMODELE TDNN

La classe des modèles neuronaux à délais temporels TDNN, NNAR,...servent à spécifier et prévoir les systèmes dynamiques.

5.1 LES MODELES DYNAMIQUES

Les réseaux dynamiques (Norgaard, 2003) non récurrents sont une extension des réseaux de neurones statiques via la considération du temps ; ils servent à modéliser les systèmes dynamiques non linéaires :

$$Y_k = f(X_k, X_{k-1}, \dots, Y_{k-1}, Y_{k-2}, \dots) + \varepsilon_k \quad (5.1)$$

X_{k-1}, X_{k-2} , vecteurs inputs retardés

Y_{k-1}, Y_{k-2}, \dots : vecteurs outputs retardés

ε_k : perturbation

f : non linéaire

Les réseaux dynamiques non récurrents (NNFIR²², NNAR²³, TDNN²⁴, ...), sont des modèles à variables retardées. Ils sont appropriés à l'étude des séries temporelles.

5.2 LE MODELE TDNN

5.2.1 STRUCTURE TDNN

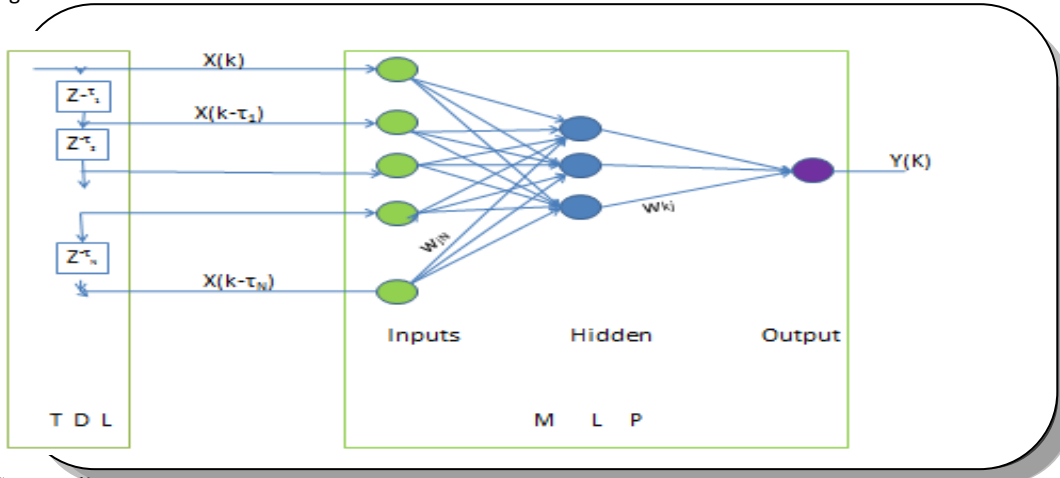
Le modèle TDNN (Abrahart, 2004) est un modèle FFNN à qui est rajouté un filtre temporel TDL en input. Dans ce modèle, on se limite, contrairement aux modèles NNFIR et NNAR, à la présentation de la chronique $X(k)$ et au choix de la dimension de la TDL. Cette dernière s'occupe de l'agencement de la chronique en vecteurs retardés $X(k), X(k-\tau_1), \dots, X(k-\tau_N)$ [figure 5.1].

²²Wan(E.A.), *FIR neural network with application in time series prediction*, Stanford university, p.22,1993

²³Connor(J.T.), Martin(R.D.)&Atlas(L.E.), *Recurrent neural networks and Robust time series prediction*, p.240, 1994

²⁴Lang(K.J.), Waibel(A.) and Hinton(G.E.), *A time delay neural network for isolate word recognition*, p.23, 1990

Figure 5. 1Structure TDNN



Source : l'auteur

Ce modèle, caractérisé par un filtre TDL localisé seulement en input, ne requiert pas pour son entraînement de rétro-propagation dynamique (BPTT²⁵, RTRL²⁶). Il est entraîné, généralement, par l'algorithme retro propagation par descente du gradient **GDBP** (Sivanandam, 2006).

Comme le montre la figure 5.1, les inputs d'entrée incluent non seulement la valeur courante du vecteur input X(k) mais aussi les valeurs retardées de ce vecteur : X(k-τ₁), X(k-τ₂),..... X(k-τ_N). L'output du neurone j est alors²⁷:

$$Y_j(k) = \Psi_1\left(\sum_{i=0}^N w_{ji} x(k - \tau_i) \pm \theta_j\right)$$

$\theta_j = \text{biais du neurone } j$

$\Psi = \text{function d'activation}$ (5.2)

$\tau_i = \text{retard}$

L'output final Y(k) est :

$$Y(k) = \Psi_2\left[\sum_j v_{kj} Y_j(k) \pm \theta_k\right] = \Psi_2\left[\sum_j v_{kj} \Psi_1\left(\sum_{i=0}^N w_{ji} x(k - \tau_i) \pm \theta_j\right) \pm \theta_k\right] \quad (5.3)$$

²⁵Mehrotra (K.), Mohan(C.K.) &Ranka(S.), *Elements of artificial neural networks*, p.71, 1997

²⁶Hassoun (M.H.), *Fundamentals in artificial neural networks*, p.274, 1995

²⁷Fa-Long (L.), *Digital Front-End in wireless communications and broadcasting :circuits and signal processing*, p.200, 2011

5.2.2PREVISION

5.2.2.1 DONNEES

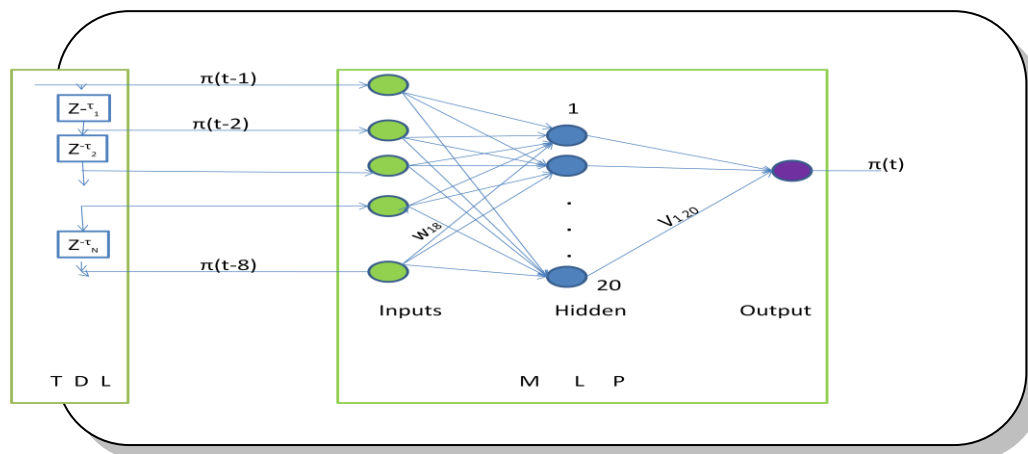
Considérons le modèle TDNN pour prévoir les IPC en Algérie (π_t). Nous reprenons les mêmes données prétraitées et hypothèses énoncées dans le modèle FFNN.

5.2.2.2 ARCHITECTURE DU RESEAU

Ce modèle comporte une seule couche cachée qui comprend au plus 40 neurones. Les fonctions d'activation et d'output sont 'tanh'. A partir du vecteur input X(t), sont générés, au moyen du filtre TDL, les vecteurs retardés X(t-1), X(t-2), ..., X(t-d) [fig5.2]. L'output Y(t) est :

$$\pi(t) = \Psi_2 \left[\sum_{j=1}^{20} v_{kj} Y_j(t) \pm \theta_k \right] = \Psi_2 \left[\sum_{j=1}^{20} v_{kj} \Psi_1 \left(\sum_{i=1}^8 w_{ji} \pi(t-i) \pm \theta_j \right) \pm \theta_k \right] \quad (5.4)$$

Figure 5. 2 Architecture du modèle



Source : l'auteur

5.2.2.3 METHODOLOGIE DE PREVISION

Nous appliquerons la méthodologie utilisée dans le modèle FFNN. Les entrainements seront effectués à l'aide MATLAB 2011a.

5.2.2.4 SPECIFICATION ET RESULTATS

L'IPC π_t^p au temps t est une fonction au plus des 8 taux d'inflation retardés, soit

$$\pi_t^p = f(\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-p}); p=1,8$$

Le choix de la variante représentative du modèle se fera de la même manière que précédemment.

Sur la base des résultats obtenus et consignés dans le tableau ci-dessous, la variante π_t^{8*} [8:25:1] représentera le modèle TDNN.

Tableau 5.1 Performance des variantes du modèle TDNNsource matlab

VARIANTE	Nombre unités dans couche cachée	MSE	MAE	R ²
π_t^1	40	0.4902	0.5322	0.3713
π_t^2	40	0.2856	0.3918	0.6328
π_t^3	40	0.0330	0.1028	0.9575
π_t^4	40	2.31E-5	0.0019	0.9
π_t^5	40	1.83E-12	9.54E-7	0.9
π_t^6	40	6.92E-12	1.70E-6	0.9
π_t^7	40	8.55e-14	2.16e-7	0.9
π_t^8	25	1.03e-14	6.93e-8	0.89

Les principaux paramètres d'entraînement retenus sont résumés dans le tableau suivant :

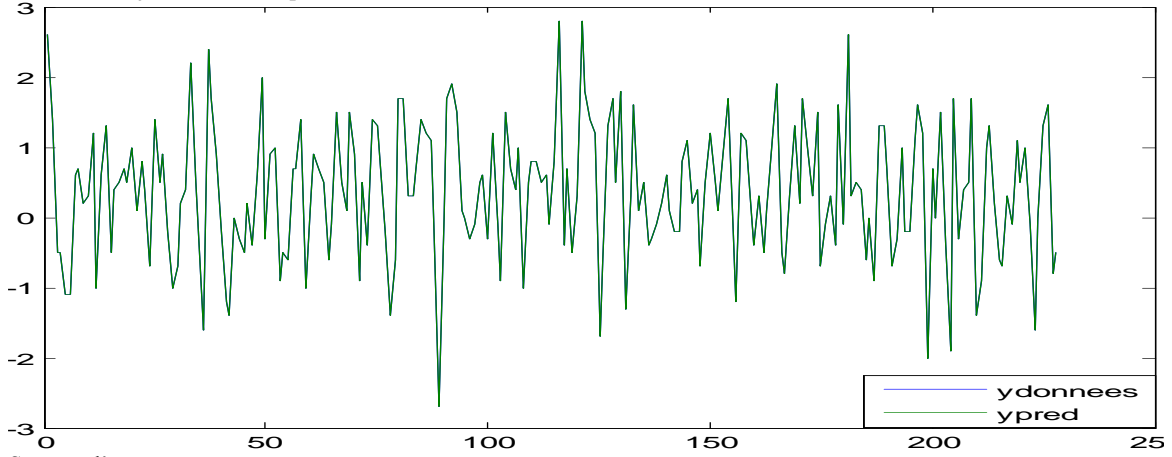
Tableau 5.2 Paramètres d'entraînement du modèle TDNN source matlab

Nombre limite d'époques	100
mem_reduc	1
fonction d'apprentissage	'learngdm'
momentum	0.001
Fonction d'entraînement	'traingdx'
Mse_goal	0
Min_grad	e-10
Mu_dec	0.1
Mu_inc	10
Max_fail	05

A. Prédiction

Le graphe 5.3 suivant illustre l'excellente qualité d'ajustement de ce modèle à la série YTSA.

Figure 5.3 Ajustement d'Yt par TDNN



Source : l'auteur

C. Prédiction

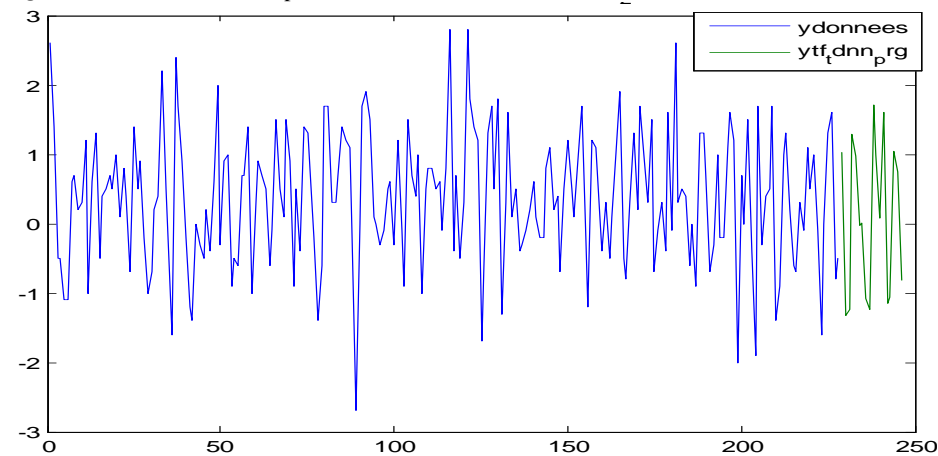
Le tableau 5.3 ci-dessous donne les prévisions des IPC y_t par ce modèle .

Tableau 5.3 IPC prévisionnels par TDNN

Periode	janv-21	févr-21	mars-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
ytsaf	0.16	-1.55	-1.55	1.42	1.42	0.39	0.39	-1.64	-1.64
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
ytf	1.01	-1.32	-1.24	1.29	0.98	-0.03	0.00	-1.07	-1.23
Periode	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
ytsaf	1.51	1.51	0.74	0.74	-1.37	-1.37	1.18	1.18	-0.39
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
ytf	1.72	0.98	0.08	1.59	-1.14	-1.06	1.05	0.74	-0.81

Le graphe 5.4 représente les IPC prévisionnels 'ytf' par ce modèle.

Figure 5.4 Prédiction de Yt par TDNN



Source : l'auteur

5.2.2.5 PERFORMANCE RELATIVE DU MODELE TDNN

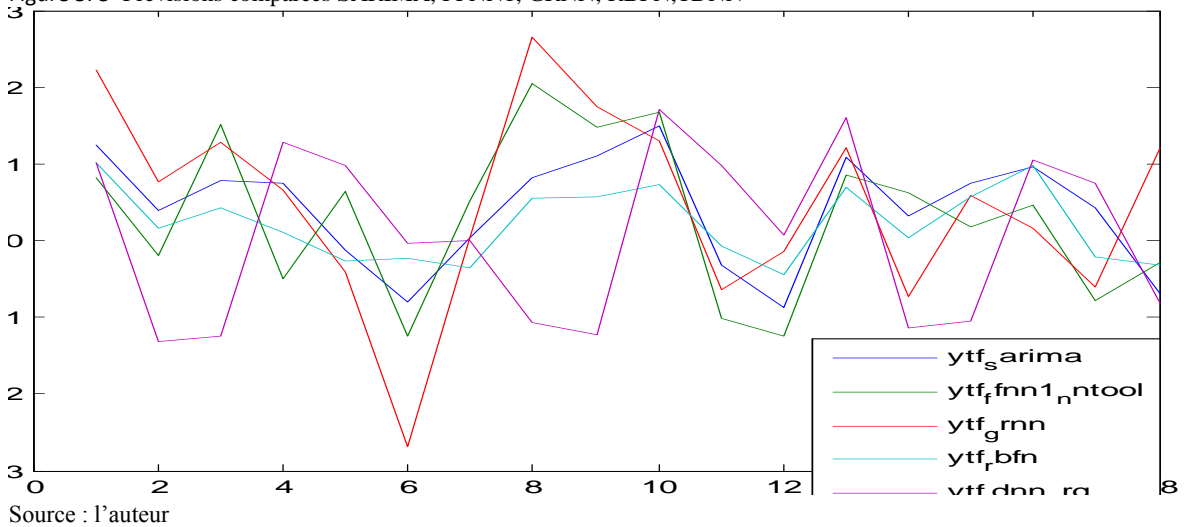
Le modèle TDNN s'avère plus performant que les modèles SARIMA et FFNN1 et presque aussi performant que le modèle GRNN [tab 5.4]

Tableau 5.4 Performance comparée du modèle TDNN Source : l'auteur

Variante	RMSE	MAE	R ²
Sarima()	0.81	0.65	0.29
FFNN1	0.63	0.48	0.34
GRNN	6.98e-13	6.66e-14	0.9
RBFN	8.49e-16	6.11e-16	0.9
TDNN	1.01e-7	6.93e-8	0.89

Le graphe (fig 5.5) expose les prévisions du modèle TDNN relativement à celles des modèles SARIMA, FFNN1, GRNN et RBFN.

Figure 5.5 Prévisions comparées SARIMA, FFNN1, GRNN, RBFN,TDNN



Source : l'auteur

5.3 LE MODELE NNAR²⁸

5.3.1 STRUCTURE

A l’instar du modèle statistique autorégressif non linéaire NARX(p), le prédicteur du modèle neuronal NNARX (Leontaris, 2007) [fig5.2] est défini par ²⁹:

$$\hat{Y}(t) = F [Y(t-1), Y(t-2), \dots, Y(t-n_Y), X(t-d), X(t-d-1), \dots, X(t-d-n_X)]$$

n_Y : retard du vecteur Y

n_X : retard du vecteur X (5.5)

F : non linéaire

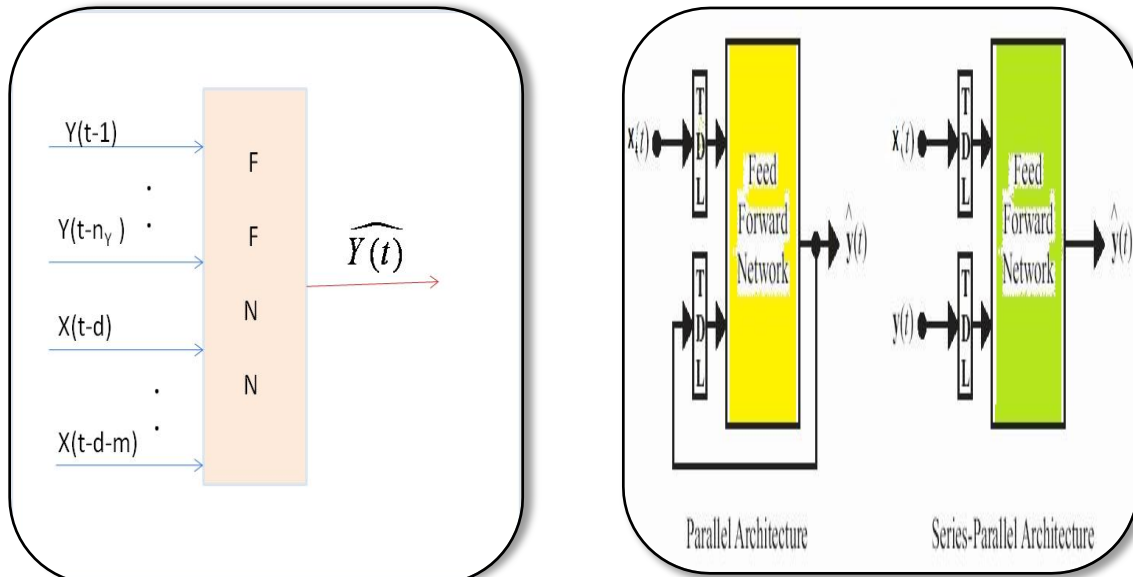
$X : X[n,1] ; Y : Y[n,1]$

d : opérateur de retard

La figure 5.6 illustre la forme canonique du modèle NNARX. L’architecture ‘ Parallel

Architecture’ est un excellent outil de prévision (Karbasi, 2009) (fig 5.7).

Figure 5.6 Structure du NNARX Figure 5.7 Architectures du modèle NNARX



Source : l’auteur Source: NARX, Matlab 2011a

²⁸Gharlegghi(B.)&Nor(A.H.S.M.), *Application of dynamic models for exchange rate prediction*, p.459-464, 2011

²⁹Matlab 2011a, *Neural Time Series Tool (ntstool)*, 2011

5.3.2 ARCHITECTURE DU RESEAU

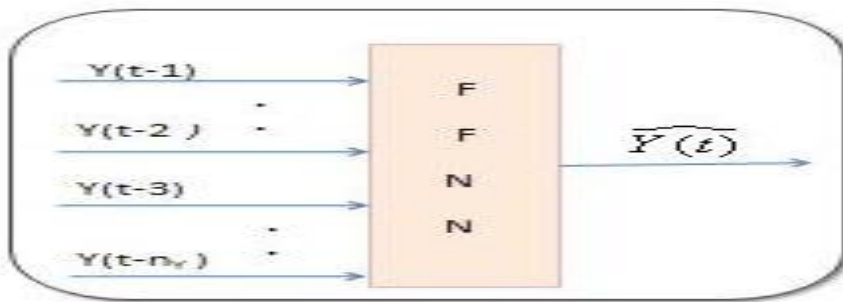
Le prédicteur de ce modèle est défini par :

$$\hat{Y}(t) = F(Y(t-1), Y(t-2), \dots, Y(t-n_y)) \quad (5.6)$$

n_y : retard du vecteur Y
 F : fonction non linéaire
 Y : $Y[n,1]$

La figure ci-dessous illustre l'architecture 'SeriesParallel Architecture' de notre modèle NNAR.

Figure 5. 8 'Seriesparallel architecture' du NNAR



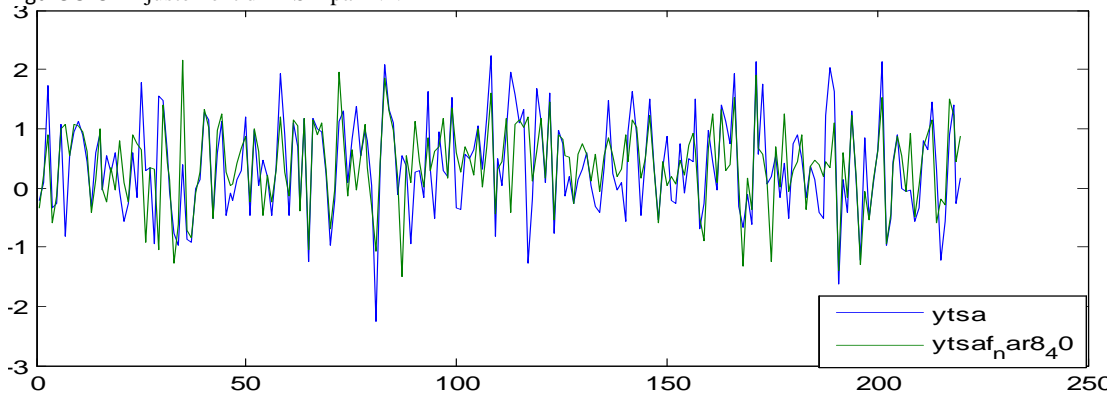
Source : l'auteur

5.3.3 PREVISION

5.3.3.1 PREDICTION

Le graphe 5.9 suivant illustre la qualité d'ajustement de ce modèle à la série Y TSA.

Figure 5. 9 Ajustement d'YTSA par NNAR



Source : l'auteur

5.3.3.2 PREVISION

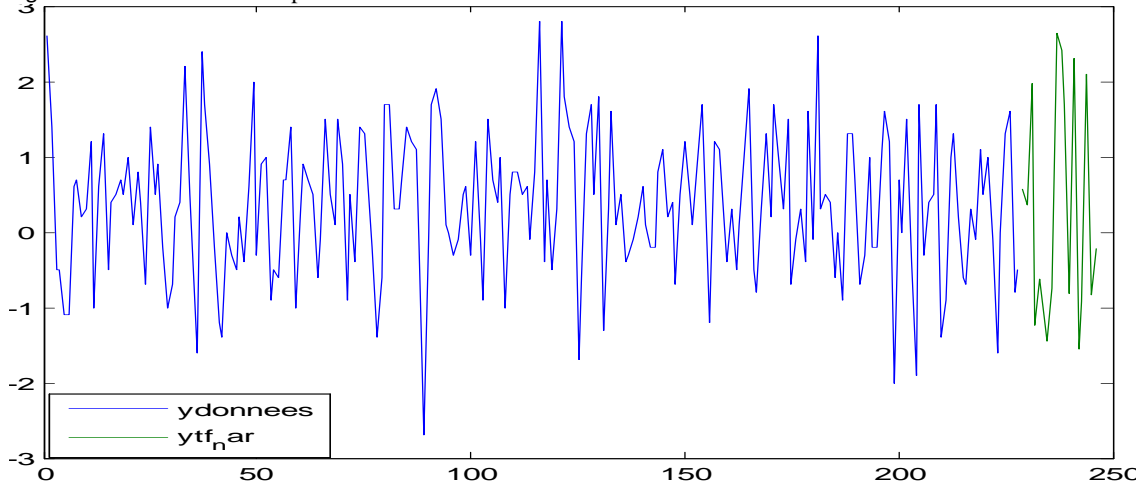
Le tableau 5.5 donne les prévisions des IPC yt par ce modèle .

Tableau 5.5 IPC prévisionnels par NNARSource : l'auteur

Periode	janv-21	févr-21	mar-21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
ytsaf	-0.36	-2.24	1.70	-0.67	0.34	-0.43	-0.25	1.15	1.51
ytf(-12)	-0.10	0.50	0.60	-0.30	1.20	-0.10	-0.90	1.50	0.70
ytf	-0.46	-1.74	2.30	-0.97	1.54	-0.53	-1.15	2.65	2.21
Periode	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
ytsaf	0.73	0.00	-2.00	-1.71	1.29	1.47	-2.11	-1.46	-0.55
ytf(-12)	0.40	1.00	-0.90	-0.46	-1.74	2.30	-0.97	1.54	-0.53
ytf	1.13	1.00	-2.90	-2.18	-0.46	3.77	-3.08	0.08	-1.08

Le graphe suivant affichera la série des IPC prévisionnels de ce modèle (fig 5.10).

Figure 5.10 Prédiction de Yt par NNAR



Source : l'auteur

5.3.4 PERFORMANCE RELATIVE DU MODELE NNAR

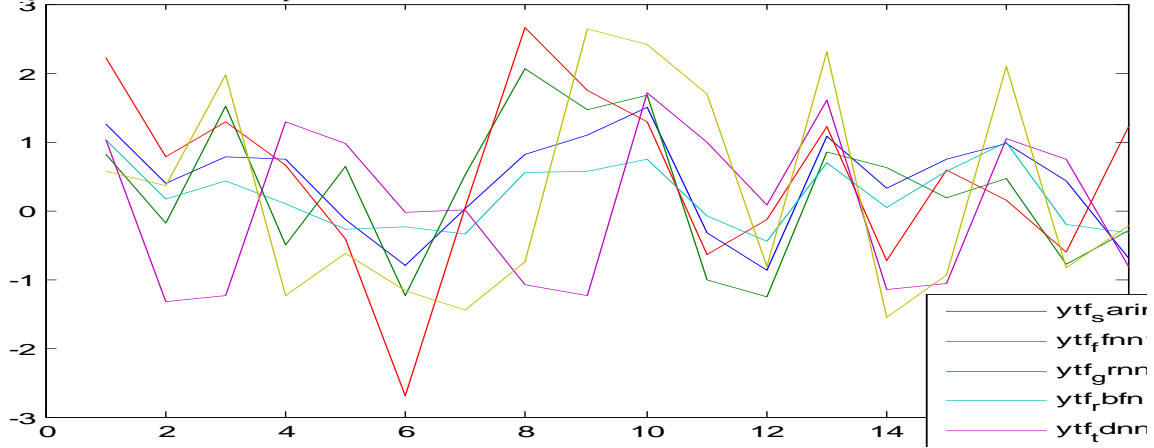
Le modèle NNAR s'avère plus performant que les modèles SARIMA et FFNN mais moins performant que les modèles GRNN, RBFN et TDNN [tab 5.6].

Tableau 5.6 Performance comparée du modèle NNARSource : l'auteur

Variante	RMSE	MAE	R ²
<i>Sarima()</i>	0.81	0.65	0.29
<i>FFNNI</i>	0.63	0.48	0.34
GRNN	6.98e-13	6.66-14	0.9
RBFN	8.49e-16	6.11e-16	0.9
TDNN	1.01e-7	6.93e-8	0.89
NNAR	0.27	0.41	0.67

Le graphe 5.11reproduit les prévisions du modèle NNAR relativement à celles desmodèles SARIMA, FFNN1, GRNN, RBFN et TDNN.

Figure 5. 11 Prévisions comparées SARIMA, FFNN1,GRNN,RBFN,TDNN&NNAR



Le modèle dynamique TDNN s'avère relativement aussi bon que les modèles neuronaux « statistiques » GRNN et RBFN. Mais comme tous ces modèles ne prennent pas en considération les résidus retardés e_{t-i} , examinons maintenant les modèles récurrents RNN. Le modèle RNN avec un feedback en output à partir de $(Y - \hat{Y}_t)$ s'identifie à un NNARMA (p,q)

CH6 PREVISION DES IPC PAR LE MODELE NNARMAX

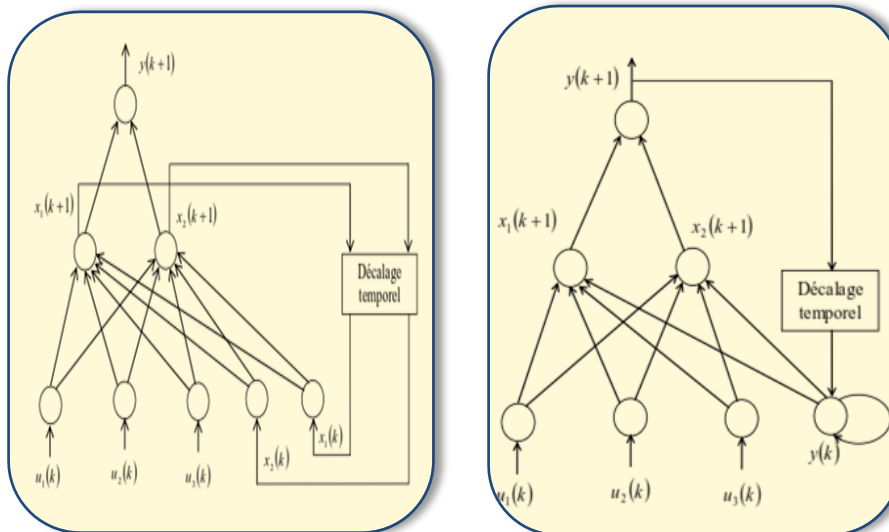
Les systèmes dynamiques récurrents sont modélisés et prévus par la classe de modèles neuronaux récurrents RNN.

6.1 LES MODELES RECURRENTS RNN

Les modèles récurrents (modèle d'Elman ENN³⁰, modèle de Jordan JNN³¹, NNOE³², NNARMAX,..) sont des modèles dynamiques qui disposent de boucles de rétroaction entre les couches via des mémoires à court terme appelées « contextes ou states ». Ces modèles sont adaptés à la prévision dynamique des séries temporelles (Medsker, 2010).

Le feedback interne des modèles récurrents peut relier, aux unités additionnelles d'input, les unités cachées [ENN, fig 6.1] ou bien les unités de sortie [JNN, fig 6.2]. Ces modèles sont caractérisés par l'adjonction d'une couche (ou plusieurs) récurrente (s) qui a (ont) le rôle de mémoire à court terme. Le prédicteur est composé par la partie FFNN du réseau.

Figure 6. 1 Réseau d'ELMAN (ENN) Figure 6. 2 Réseau de JORDAN (JNN)



Source: [70], p.188

Source: [70], p.189

³⁰Elman (J.L.), *Finding structure in time*, p.179, 1990

³¹Jordan (M.I.), *Attractor dynamics and parallelism in connectionist sequential machine*, p.531, 1986

³²Thiaw (L.), *Identification de systèmes dynamiques non linéaires par réseaux de neurones et multimodèles*, p.42, 2008

Un réseau récurrent à temps discret d'ordre K peut être représenté par K équations aux différences d'ordre 1. Sa dynamique est³³:

$$\begin{aligned} X(t+1) &= \Phi[U(t), X(t), w] \\ Y(t) &= \Pi[U(t), X(t), w] \end{aligned} \quad (6.1)$$

/ $U(t) = [U_1(t), \dots, U_N(t)]$: vecteur d'entrée
 $X(t) = [X_1(t), \dots, X_K(t)]$: vecteur d'état
 w : poids
 $Y(t)$: output
 Φ, Π : fonctions non linéaires

Ils ne prennent pas en considération les résidus retardés e_{t-i} . Le modèle RNN avec un feedback en output à partir de $(Y_t - \widehat{Y}_t)$ s'identifie alors à un NNARMA (p,q).

Les modèles NNARMAX (NNARX) sont des modèles statistiques ARMAX (ARX) implantés dans des RNA à fonctions d'activation non linéaires (Likothanassis, 1998).

6.2 LE MODELE NNARMAX (BECCALI, 2008)

6.2.1 STRUCTURE

Le modèle ARMAX, selon L. Ljung³⁴, peut s'écrire sous la forme : (Beccali, 2008)

$$y(t) = G(q, \theta) x(t) + H(q, \theta) \varepsilon(t) \quad (6.2)$$

$G; H$: fonctions d'activation,

$$G(q, \theta) = \frac{B(q)}{A(q)}$$

$$H(q, \theta) = \frac{C(q)}{A(q)}$$

$$A(q) = 1 + \sum_{i=1}^{n_a} a_i q^{-i}$$

$$B(q) = \sum_{i=1}^{n_b} b_i q^{-i}$$

$$C(q) = \sum_{i=1}^{n_c} c_i q^{-i}$$

³³Kolen (J.F.) & Kremer (S.C.), *A field guide to dynamical recurrent networks*, p.34, 2001

³⁴Ljung (L.), *System Identification: theory for user*, p.73, 1987

Son prédicteur optimal est : $\hat{y}(t / \theta) = H^{-1}(q)G(q)x(t) + [1 - H^{-1}(q)]y(t)$ (6.3)

L'erreur de prévision est : $e(t) = -H^{-1}(q)G(q)x(t) + H^{-1}(q)y(t)$ (6.4)

Partant des fonctions d'activation H et G ci-dessus, (6.3) se reformalise ainsi :

$$\hat{y}(t/\theta) = \frac{B(q)}{C(q)}x(t) + \left[1 - \frac{A(q)}{C(q)}\right]y(t) = B(q)x(t) + [1 - A(q)]y(t) + [C(q) - 1]e(t, \theta) \quad (6.5)$$

$$= \phi^T(t, \theta) \theta$$

$\phi(t, \theta) = [-y(t-1), \dots, -y(t-n_a), x(t-1), \dots, x(t-n_b), e(t, \theta), \dots, e(t-n_c, \theta)]^T$ est son vecteur de regression

$\theta = [a_1, \dots, a_{n_a}, b_1, \dots, b_{n_b}, c_1, \dots, c_{n_c}]^T$ est son vecteur de poids.

NNARMAX est la version non linéaire du modèle ARMAX ci-dessus.

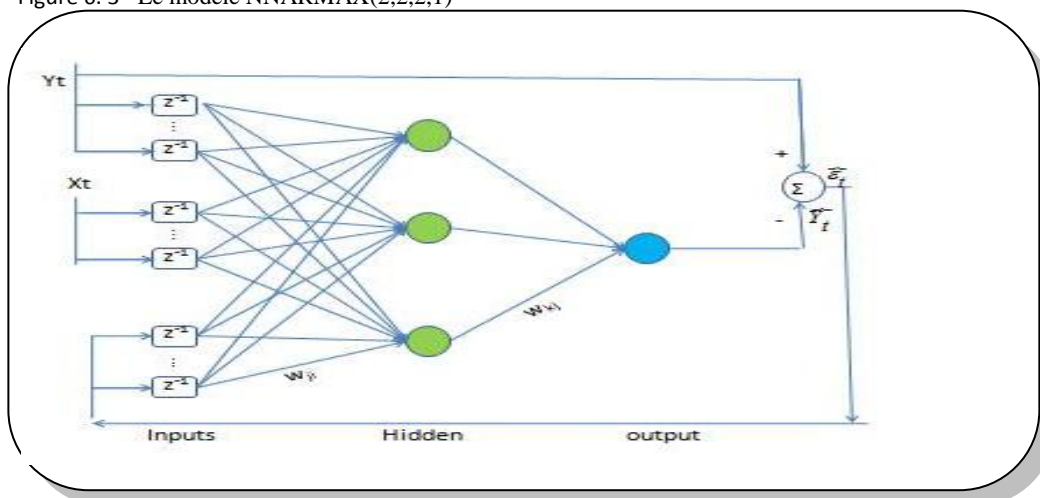
Son modèle est $y(t) = g[\varphi(\theta, t), \theta] + \varepsilon(t);$ (6.6)

Son prédicteur est $\hat{y} = g[\varphi(t, \theta), \theta].$

$\varphi(t, \theta); \theta$ sont respectivement le vecteur de régression et de poids.

La figure 6.3 ci-dessous présente la structure de base du modèle NNARMAX(2,2,2,1).

Figure 6.3 Le modèle NNARMAX(2,2,2,1)



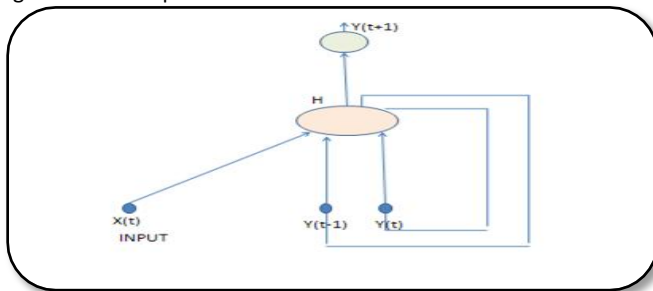
Source : l'auteur

6.2.2 L'ALGORITHME D'ENTRAINEMENT BPTT

Les principaux algorithmes d'entraînement des réseaux récurrents sont le BPTT (Werbos, 1990) et le RTRL (Cloette, 1997). BPTT est une adaptation de l'algorithme BP aux spécificités du réseau récurrent. Il procède à la formulation de la réponse forward puis au dépliage du réseau récurrent avant de lui appliquer la procédure BP.

Illustrons ceci sur le modèle récurrent simple suivant (Laxmidhar, 2008): $y(t+1) = w_1 y(t) + w_2 y(t-1) + w_3 x(t)$ [fig 6.4]

Figure 6. 4 Exemple du réseau récurrent



Source : l'auteur

1- écrivons la réponse forward de ce réseau :

$$Y(2) = w_1 Y(1) + w_2 Y(0) + w_3 X(1) \quad (6.7)$$

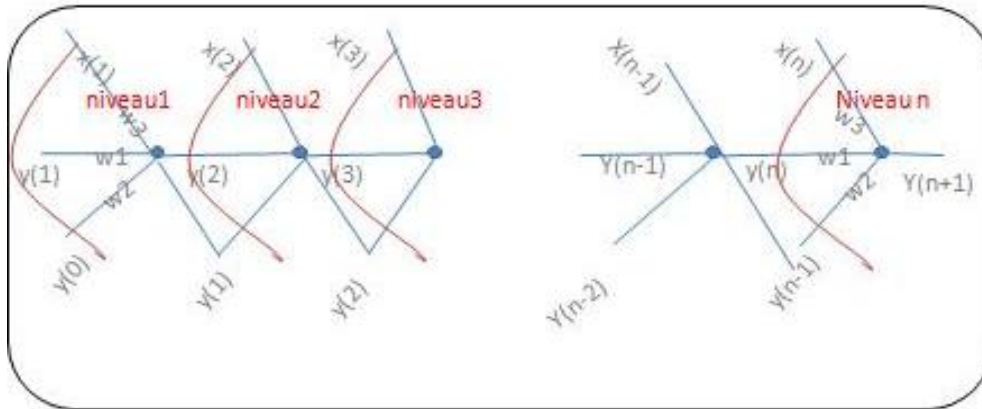
$$Y(3) = w_1 Y(2) + w_2 Y(1) + w_3 X(2)$$

.....

$$Y(t+1) = w_1 Y(t) + w_2 Y(t-1) + w_3 X(t)$$

2- Déplions ce réseau

Figure 6. 5 Réseau récurrent déplié



- 3- Calculons les valeurs $y(2), \dots, y(t+1)$, étant donnés $y(0)$ et les inputs $x(1) ; \dots, x(t)$,
- 4- calculons pour le niveau n l'erreur $e(t+1) = y^d(t+1) - y(t+1)$ puis la variation des poids:

$$\Delta w_3(t+1) = \eta \delta_{t+1} X(t) ; \Delta w_1(t+1) = \eta \delta_{t+1} y(t) ; \Delta w_2(t+1) = \eta \delta_{t+1} y(t-1)$$

$$\delta_{t+1} = y(t+1) [1 - y(t+1)] e(t+1)$$

- 5- procédons de la même manière pour les niveaux n-1 à 1.

La règle de mise à jour des poids est alors : $w_k^{new} = w_k^{old} + \sum_{i=1}^{n+1} \Delta w_k(i) \quad \backslash k = \overline{1,3}$

Appliquons maintenant l'algorithme BP (décrit dans 2.2.2.).

6.3 PREVISION

6.3.1 DONNEES

Soit $Z^N = \{ [X(t), Y(t)] ; t = \overline{1, N} \}$ / $X(t)$ = les inputs au temps t, $Y(t)$ =l'output objectif
 Ces données sont divisées en 2 groupes: entraînement (X_1, Y_1) et validation (X_2, Y_2) .

6.3.2 ARCHITECTURE DU MODELE

Les régresseurs sont ceux utilisés auparavant par le modèle ARMA(p,q):

- choix de la structure des régresseurs : Nous utiliserons le vecteur de régression suivant : $\Phi(t, \Theta) = [Y(t-1), Y(t-2), X(t-1), X(t-2), e(t-1), e(t-2)]^T$ réduit à $\Phi(t, \Theta) =$

$[\pi(t-1), \pi(t-2), e(t-1), e(t-2)]^T$, Le lag de chaque vecteur d'input est fixé selon la méthode 'trial and error'. Le vecteur de poids est $\Theta = [w_1, w_2]^T$ / $w_1 = \text{rand}(5,5)$; $w_2 = \text{rand}(1,6)$

- Choix de l'architecture du réseau : Nous utiliserons un réseau composé, en plus de la couche d'inputs et d'inputs additionnels, de 5 unités cachées munies de fonctions « tanh » plus une unité d'output munie de fonction « purelin ».
- Nous utiliserons la fonction NNARMAX2, l'algorithme d'entraînement LM et l'algorithme d'apprentissage « traingdm » (Norgaard, 2000).

Les principaux paramètres d'entraînement retenus sont résumés dans le tableau suivant

Tableau 6. 1 Paramètres d'entraînement du modèle NNARMA source matlab

<i>Nombre limite d'époques</i>	<i>130</i>
<i>Méthode</i>	<i>'ff'</i>
<i>fonction d'apprentissage</i>	<i>'traingdm'</i>
<i>momentum</i>	<i>0.001</i>
<i>Fonction d'entraînement</i>	<i>'trainlm'</i>
<i>Mse_goal</i>	<i>0</i>
<i>Min_grad</i>	<i>1.0 e-04</i>
<i>Mu_dec</i>	<i>0.1</i>
<i>Mu_inc</i>	<i>10</i>
<i>Max fail</i>	<i>05</i>

6.3.3 METHODOLOGIE DE PREVISION

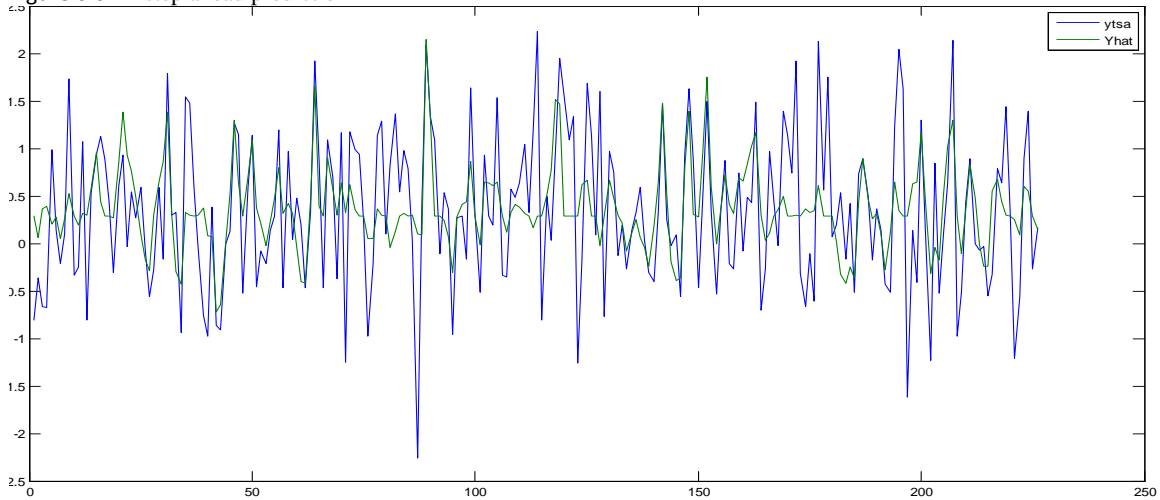
Nous utiliserons la méthodologie de prévision appliquée au modèle TDNN hormis que les prédictions se feront à 1,3 et 12 step-ahead. Les entraînements s'effectueront à l'aide MATLAB 11.

6.3.4 SPECIFICATION ET RESULTATS

6.3.4.1 PREDICTION

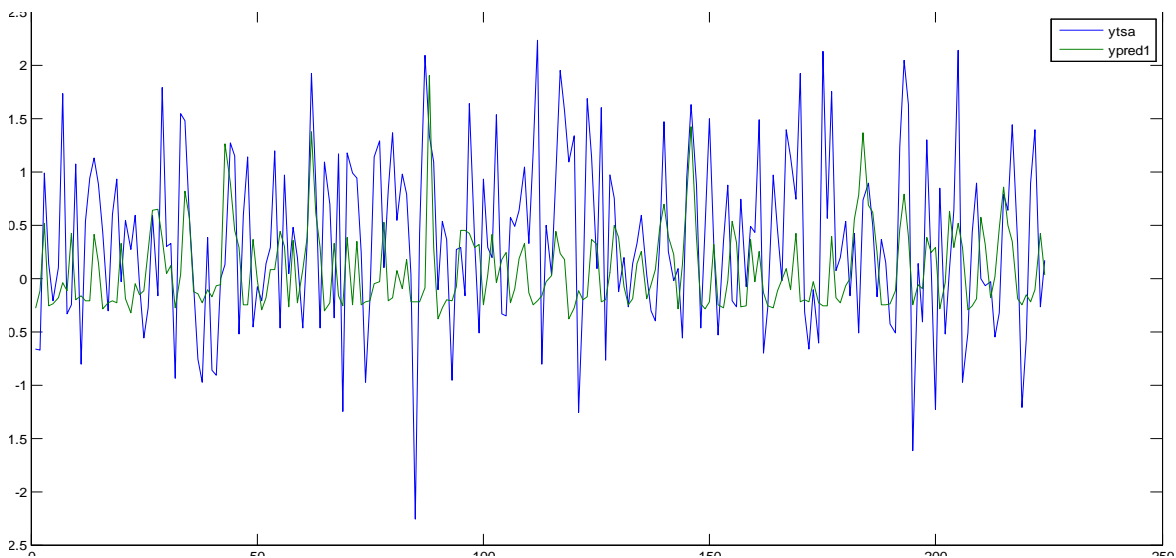
Les graphes suivants 6.6, 6.7 et 6.8 illustrent la qualité des prédictions 1,3 et 12 steps ahead de la série YTSA par ce modèle.

Figure 6.6 1-step ahead prediction



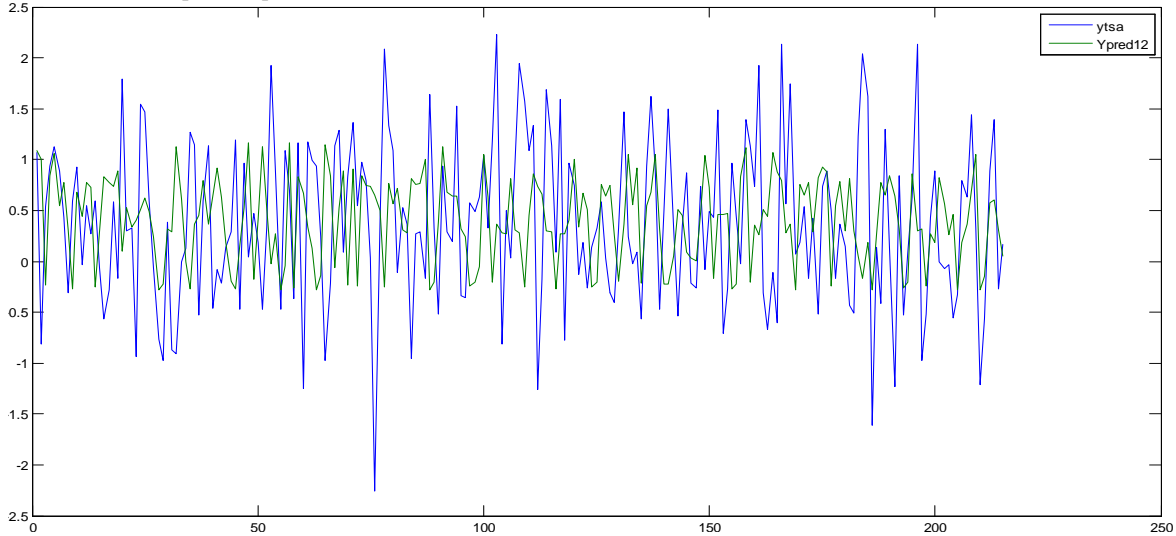
source: Matlab 2011

Figure 6.7 3-step ahead prediction



source: Matlab 2011

Figure 6.8 12-step ahead prediction source: Matlab 2011



source: Matlab 2011

6.3.4.2 PREVISION

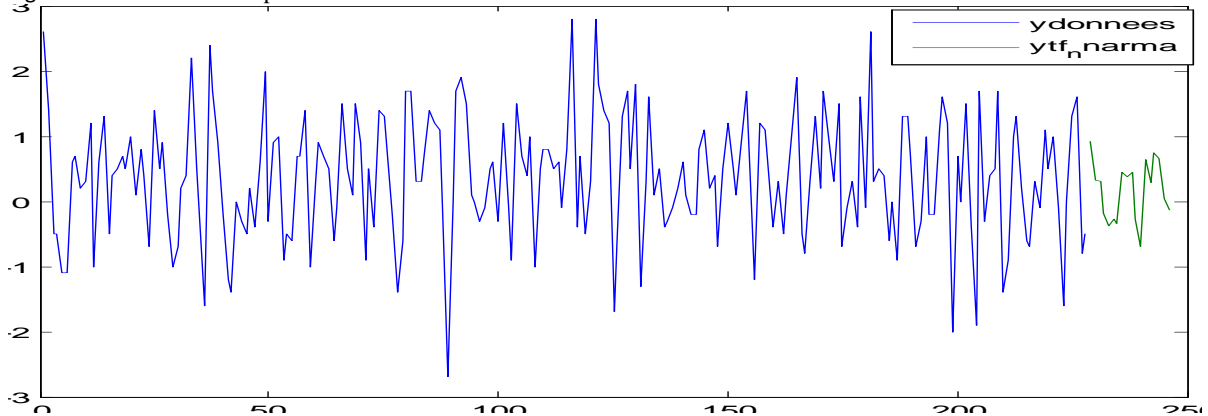
Le tableau 6.2 suivant donne les prévisions des IPC yt par ce modèle .

Tableau 6.2 IPC prévisionnels par NNARMA source: l'auteur

Periode	janv-21	févr-21	mars21	avr-21	mai-21	juin-21	juil-21	août-21	sept-21
ytsaf	0.07	0.09	0.00	-0.05	0.05	0.16	0.05	-0.11	-0.03
factors	0.85	0.23	0.31	-0.13	-0.44	-0.42	-0.39	0.57	0.41
ytf	0.92	0.32	0.31	-0.18	-0.39	-0.26	-0.34	0.46	0.38
Periode	oct-21	nov-21	déc-21	janv-22	févr-22	mars-22	avr-22	mai-22	juin-22
ytsaf	0.23	0.26	-0.03	-0.022	0.06	0.44	0.79	0.49	0.29
factors	0.21	-0.53	-0.66	0.85	0.23	0.31	-0.13	-0.44	-0.42
ytf	0.44	-0.27	-0.69	0.63	0.29	0.75	0.66	0.05	-0.13

Le graphe 6.9 représente les IPC prévisionnels 'ytf' par ce modèle .

Figure 6.9Prévision d' Yt par NNARMA



Source : l'auteur

6.4 PERFORMANCE COMPAREE DU MODELE NNARMA

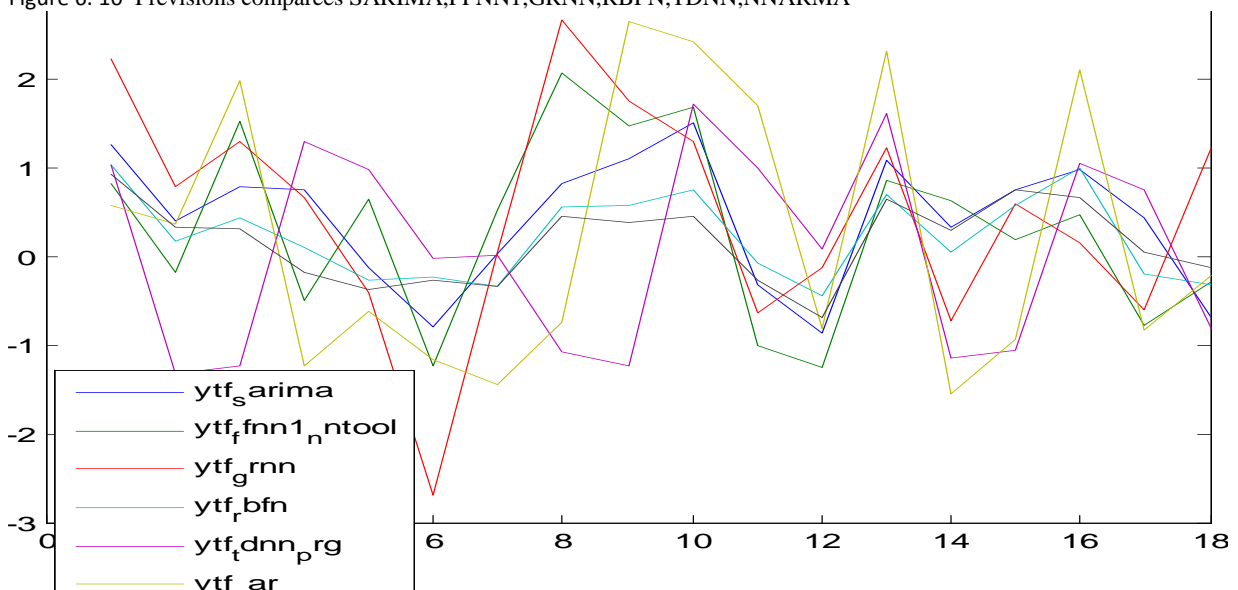
En conclusion, Le modèle NNARMA est moins performant relativement aux modèles FFNN1, GRNN, RBFN et TDNN (tableau6.3).

Tableau 6. 3 Performance comparée du modèle NNARMA source: l'auteur

Variante	RMSE	MAE	R ²
Sarima()()	0.81	0.65	0.29
FFNN1	0.63	0.48	0.34
GRNN	6.98e-13	6.66e-14	0.9
RBFN	8.49e-16	6.11e-16	0.9
TDNN	1.01e-7	6.93e-8	0.89
NNAR	0.27	0.41	0.67
NNARMA	0.46	0.52	0.28

Le graphe ci-dessous illustre les prévisions du modèle NNARMA relativement à celles des modèles SARIMA, FFNN1, GRNN, RBFN, TDNN et NNAR.

Figure 6. 10 Prévisions comparées SARIMA,FFNN1,GRNN,RBFN,TDNN,NNARMA



source: Matlab 2011

Le modèle NNARMA (2,2,1), bien que plus étoffé dans son architecture, se révèle paradoxalement être moins performant en prévisibilité que prévu (RMSE=0.46, MAE=0.52). Ceci serait probablement lié aux nombres de retard encore modestes.

L'impossibilité d'expliquer les résultats obtenus par les modèles neuronaux ont fait qualifier ces modèles de «boîtes noires» et ouvert la voie aux modèles de logique floue FL (Zadeh,65).

CH7 PREVISION DES IPC PAR LE MODELE FL

La logique floue (LF) est une généralisation de la logique booléenne classique. Dans ce cadre, un sous ensemble flou (SEF) est défini par une fonction d'appartenance (MF) à valeur dans [0,1].

7.1 THEORIE DES ENSEMBLES FLOUS

7.1.1 SOUS ENSEMBLE FLOU³⁵

Un SEF A de X se note : $A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\}$ / $\mu_A(x)$ son MF index

Dans un référentiel X, un SEF peut se représenter par diverses MF (trapmf, trimf,...)

$$\text{- trapmf}(\alpha, \beta) = \begin{cases} 0 & x \leq \alpha \\ (x - \alpha) / (\beta - \alpha) & \alpha \leq x \leq \beta \\ 1 & \beta \leq x \leq \delta \\ (\gamma - x) / (\delta - \gamma) & \delta \leq x \leq \gamma \\ 1 & \gamma \leq x \end{cases}$$

$$\text{- trimf}(\alpha, \beta, \gamma) = \begin{cases} 0 & 0 \leq x \\ \frac{x - \alpha}{\beta - \alpha} & \alpha \leq x \leq \beta \\ \frac{x - \beta}{\gamma - \beta} & \beta \leq x \leq \gamma \\ 0 & \gamma \leq x \end{cases}$$

7.1.2 OPERATEURS LOGIQUES

Des fonctions logiques (ou, et, non, xor, nor...) sont effectuées sur les SEF. Elles correspondent aux opérateurs (\cup, \cap, C_x, \dots) dont les principales propriétés sont:

³⁵Buckley(J.J.) &Eslami(E.),Introduction to fuzzy logic and fuzzy sets, p.21,2002.

$$\mu_{A \cup B}(x) = \max[\mu_A(x), \mu_B(x)] \quad (\text{Zaddeh})$$

$$\mu_{A \cap B}(x) = \min[\mu_A(x), \mu_B(x)]$$

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \times \mu_B(x) \quad (\text{Prod/Probor})$$

$$\mu_{A \cap B}(x) = \mu_A(x) \times \mu_B(x)$$

(Nguyen, 2006)

7.1.3 INFERENCE FLOUE

A l'instar de la logique classique, la règle d'implication est à la base de l'inférence³⁶. Elle revêt la forme de la proposition 'SI « x est A_i » (prémisse) ALORS « y est B_i » (conclusion)'

et s'identifie par sa matrice (relation) $R(x, y) = \frac{\mu_R(x, y)}{(x, y)}$:

Il existe plusieurs MF de l'implication en littérature :

$$- \mu_R(x, y) = \max[1 - \mu_A(x), \mu_B(y)] \quad (\text{Zadeh})$$

$$- \mu_R(x, y) = \min[\mu_A(x), \mu_B(y)] \quad \text{ou} \quad (\text{Mamdani})$$

$$- \mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) \quad (\text{Larsen})$$

L'inférence floue s'effectue au moyen de la composition des règles. Elle affirme que connaissant une règle R₁ et la prémisse A' d'une règle R₂, on peut inférer la conclusion B' de la règle 2 par composition : B' = A' o R₁.

On trouve plusieurs règles de composition en littérature:

$$\text{Max - min : } \mu_R(x, y) = \max_{x \in X} \{ \min[\mu_A(x), \mu_R(x, y)] \}$$

$$\text{Max - Product : } \mu_R(x, y) = \max_{x \in X} \{ \mu_A(x), \mu_R(x, y) \}$$

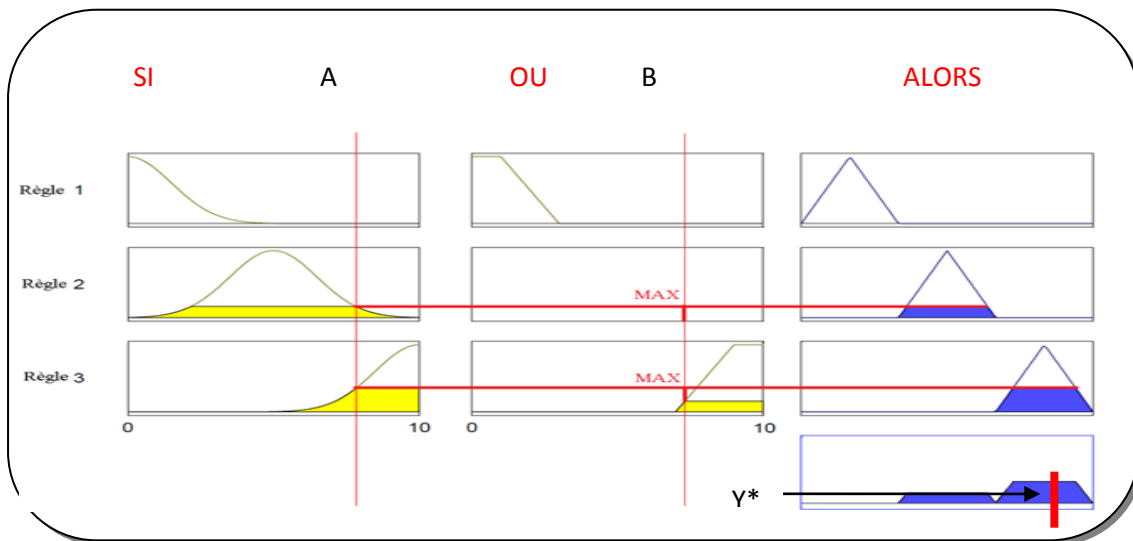
Le raisonnement approximatif de la LF s'effectue différemment selon les propriétés de la SEF et de la règle. Illustrons le cas d'une SEF continue et d'une règle multiple.

³⁶Sivanandam(S.N.), Sumathi(S.) and Deep(S.N.), introduction to fuzzy logic using matlab, p.118, 2007

Soit un système flou continu, à 2 antécédents ($x_1; x_2$) et 1 output y , représenté par :
 SI « x_1 est A_1^k » ET « x_2 est A_2^k » ALORS « y^k est B^k » $k=1;2$
 quel serait la conséquence finale y^* d'un antécédent quelconque (x_1^*, x_2^*), ?

On effectue d'abord l'opération "et" puis l'implication. Par agrégation des outputs nous obtenons, par la méthode de la coupure, la surface en bas du graphe 6.1

figure 7.1 Fuzzy Inférence Système (Mamdani)



source : Matlab

La valeur y^* s'obtient alors par défuzzification par la méthode du centroïde COG =
$$\frac{\sum_{x=a}^b \mu(x) x}{\sum_{x=a}^b \mu(x)}$$

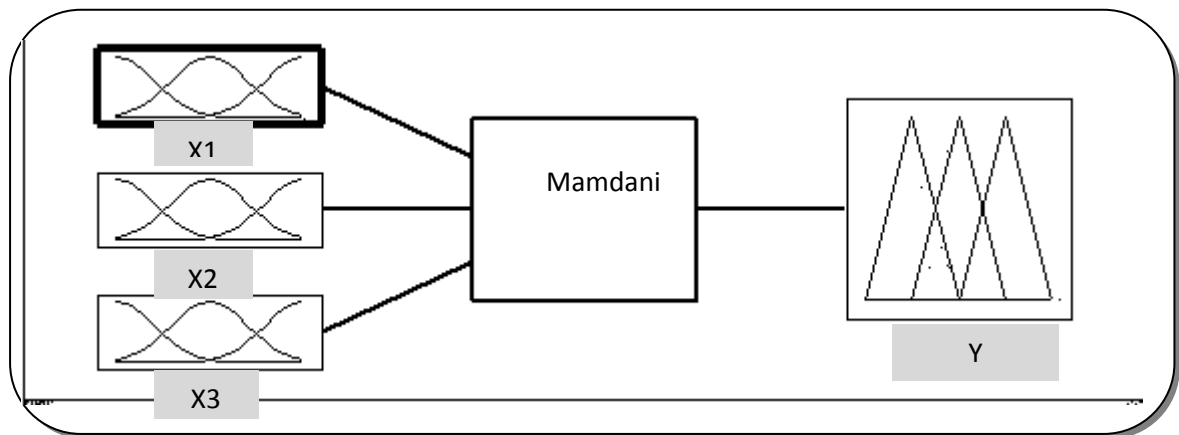
7.2 STRUCTURE

Le modèle FL est constitué de 3 couches : input, inférence et output (fig. 2).

Les données X_i sont présentées, après leur « fuzzification », à la couche inférence qui comprend les règles floues constituées des opérateurs « ou, et, ... ». Dans la couche d'inférence, on procède à l'opération floue d' « implication » puis d' « agrégation ». A l'issue de l'opération de « défuzzification », l'output final « Y_j » est enfin déterminé. Les différents opérateurs logiques relèvent des principaux types d'inférence : Mamdani ou Sugeno.

Les fonctions triangulaire, trapézoïdale, gaussienne,.... constituent les principales fonctions d'appartenance.

figure 7.2 Structure du modèle FL



source : Matlab/FL

7.3 PREVISION

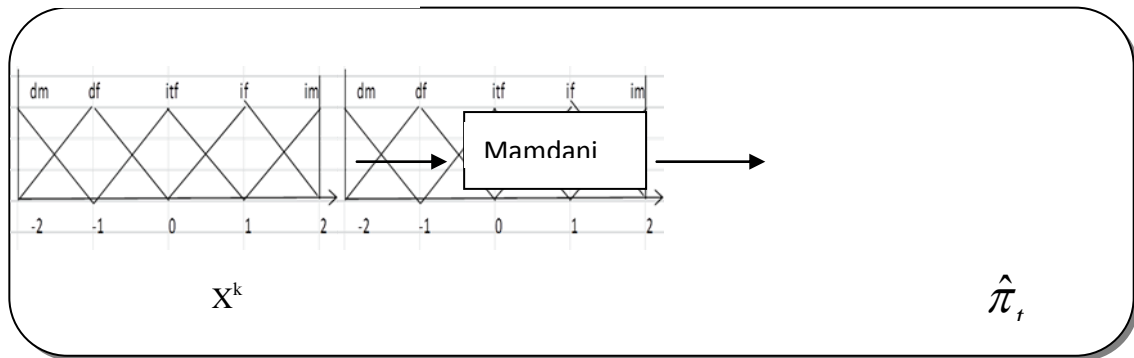
7.3.1 DONNEES

Les données sont tirées des séries « collections statistiques : IPC » publiées par l'ONS et citées dans le modèle SARIMA. La période totale s'étale sur 228 mois, de 2002 :01 à 2020:12; la période de prévision couvre 18 mois. Le prétraitement des données se restreint à leur dessaisonalisation. Parce que nous considérons un horizon de court terme, nous nous contenterons des valeurs historiques du taux d'inflation $\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}$; $k \in \mathbb{N}$ comme variables explicatives.

7.3.2 ARCHITECTURE DU SYSTEME

L'ensemble des valeurs explicatives $\{\pi_{t-1}, \pi_{t-2}, \dots, \pi_{t-k}\}$ constitue le vecteur input X^k ; le nombre de décalage k est déterminé par la variante FFNN du modèle ANN sinon par la méthode heuristique. L'output $\hat{\pi}_t$ constitue la valeur prédite de la valeur actuelle de l'IPC π_t . Le type d'inférence appliqué est celui de Mamdani, L'output $\hat{\pi}_t$ est calculé à l'issue des opérations d'agrégation par la méthode de la « coupure » et de défuzzification par la méthode « centroïde ». L'architecture du système est présentée dans la figure 7.3

figure 7. 3 Architecture du système



source : l'auteur

7.3.3 METHODOLOGIE DE PREVISION

Les prévisions s'effectuent par inférence « Mamdani », les procédures statique et dynamique. Les critères d'appréciation des prévisions sont le RMSE, MAE et R^2 . La période de prévision s'étale de 2021:01 à 2022:06. Les prédictions se feront au plus à 8-step-ahead.

7.3.4 SPECIFICATION ET RESULTATS

7.3.4.1 LE MODELE FL

La partition du range (ou univers) U des données est effectuée par la méthode heuristique. On en fixe cinq classes u_i d'amplitudes identiques égales à 1.

$$u_1=[-2.5 ; -1.5] \quad u_2=[-1.5 ; -0.5] \quad u_3=[-0.5 ; 0.5] \quad u_4=[0.5 ; 1.5] \quad u_5=[1.5 ; 2.5]$$

La fonction d'appartenance choisie est la fonction triangulaire « trimf ». Il en résulte 5 variables floues

$$\text{-déflation moyenne : } dm=A_1=[-3 \quad -2 \quad -1]$$

$$\text{-déflation faible : } df=A_2=[-2 \quad -1 \quad 0]$$

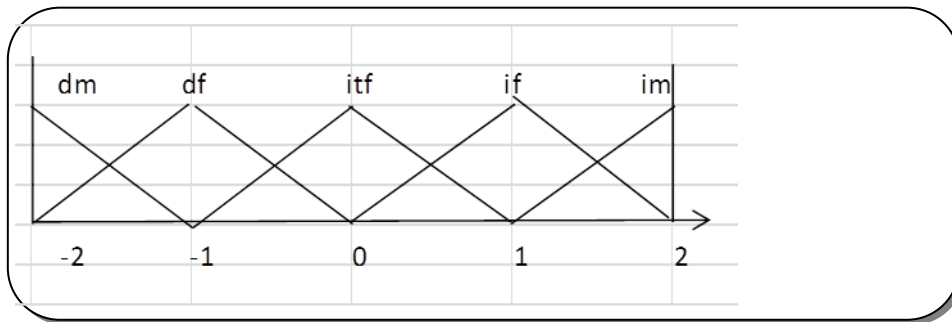
$$\text{-inflation très faible : } itf=A_3=[-1 \quad 0 \quad 1]$$

$$\text{-inflation faible : } if=A_4=[0 \quad 1 \quad 2]$$

$$\text{-inflation moyenne : } im=A_5=[1 \quad 2 \quad 3]$$

Ces cinq variables floues sont illustrées par le graphe suivant :

Figure 7. 4 Représentation des variables floues



Source : l'auteur

L'ordre de la relation entre variables est fixé à 1 par la méthode heuristique. Aussi, Les couples de données (π_{t-1}, π_t) constitueront les inputs du modèle FL .

Après fuzzification des données par la méthode d'" u_{\max} " et codification des valeurs floues dm, df, itf, if et im par respectivement 1, 2, 3, 4 et 5. Les relations entre valeurs floues sont calculées puis regroupées :

$$\text{groupe1 (dm) : } 1 \longrightarrow 3$$

$$\text{groupe2 (df) : } 2 \longrightarrow 2 ; 2 \longrightarrow 3 ; 2 \longrightarrow 4 ; 2 \longrightarrow 5$$

$$\text{groupe3 (itf) : } 3 \longrightarrow 1 ; 3 \longrightarrow 2 ; 3 \longrightarrow 3 ; 3 \longrightarrow 4 ; 3 \longrightarrow 5$$

$$\text{groupe4 (if) : } 4 \longrightarrow 2 ; 4 \longrightarrow 3 ; 4 \longrightarrow 4 ; 4 \longrightarrow 5$$

$$\text{groupe5 (im) : } 5 \longrightarrow 1 ; 5 \longrightarrow 2 ; 5 \longrightarrow 3 ; 5 \longrightarrow 4 ; 5 \longrightarrow 5$$

Puis, après traitement, ces relations sont synthétisées en 5 groupes :

$$\text{groupe1 (dm) : } 1 \longrightarrow 3(1)$$

$$\text{groupe2 (df) : } 2 \longrightarrow 3(0.9)$$

$$\text{groupe3 (itf) : } 3 \longrightarrow 4(1)$$

$$\text{groupe4 (if) : } 4 \longrightarrow 3(1)$$

groupe5 (im) : 5 \rightarrow 2(0.9)

Ces relations munies de leur ordre amorcent les inférences au moyen des règles « If...Then » suivantes.

1.if π_{t-1} is dm then π_t is itf (1)

2.if π_{t-1} is df then π_t is itf (0.9)

3.if π_{t-1} is itfthen π_t is if (1)

4.if π_{t-1} is if then π_t is itf (1)

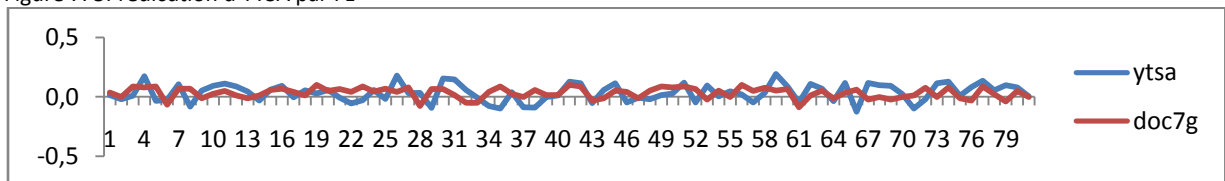
5.if π_{t-1} is im then π_t is df (0.9)

Après agrégation des inférences par la méthode de la « coupure » et de leur défuzzification par la méthode du « centroïde », les prédictions des outputs sont déterminées.

7.3.4.2 PREDICTION ET PREVISION

La prédiction des IPC est représentée par la figure ci-dessous.

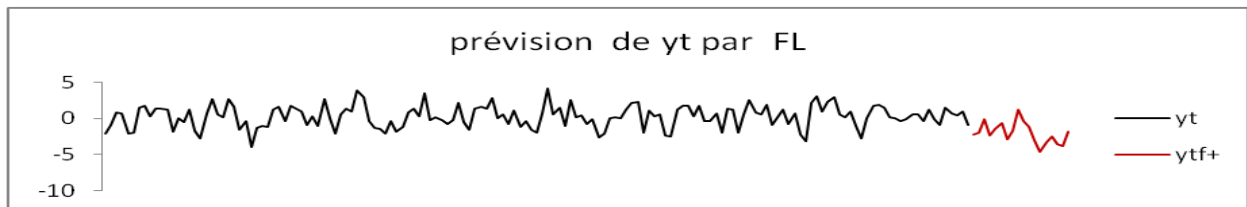
Figure 7. 5Prédiction d'YTSA par FL



source:l'auteur

La figure ci-dessous illustre les prévisions des IPC sur la période 2022:01 - 2022 :06.

figure 7. 6Prévision de Yt par FL



source: l'auteur

7.3.5 PERFORMANCE COMPAREE

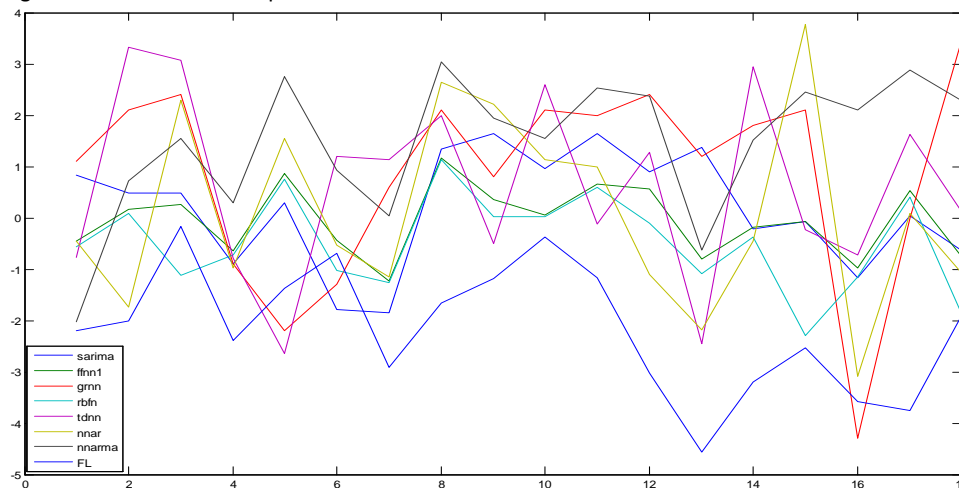
D'après les résultats des critères de performance retenus, les modèles ANN sont plus perfor-

nants que le modèle FL (tableau 7.1). Le graphe ci-dessous (figure 7.7), illustre les prévisions du modèle FL relativement à celles des modèles SARIMA, FFNN1, GRNN, RBFN, TDNN, NNAR et NNARMAX.

Tableau 7. 1 Performance comparée du modèle flousource: Matlab 2011a

Variante	RMSE	MAE	R ²
Sarima(())	0.81	0.65	0.29
FFNN1	0.63	0.48	0.34
GRNN	6.98e-13	6.66e-14	0.98
RBFN	8.49e-16	6.11e-16	0.98
TDNN	1.01e-7	6.93e-8	0.89
NNAR	0.27	0.41	0.67
NNARMA	0.46	0.52	0.28
FL	1.00	0.82	0.42

Figure 7. 7 Prévisions comparés des modèles



source: Matlab 2011

CONCLUSION

Dans le cadre de notre tentative de modélisation de l'inflation en Algérie, notre travail a consisté à confronter trois modèles : SARIMA, ANN et FL du point de vue performance en prévisibilité. Les critères de performance retenus étaient le RMSE, le MAE et R^2 .

Au cours de notre étude, nous avons trouvé utile de subdiviser les modèles neuronaux en statiques\dynamiques puis dynamiques purs\récurrents. Les modèles neuronaux statiques ont été scindés, quant à eux, en numériques\statistiques.

Au moyen de la stratégie de Box-Jenkins de l'économétrie des séries temporelles, les prévisions ont été entamées à l'aide du modèle statistique SARIMA (47,0,4) (12,2,0,2). Ce modèle a généré une faible performance en prévisibilité (RMSE= 0.81) qui est probablement dûe à son hypothèse fondamentale de linéarité.

L'application des modèles de modélisation neuronaux a conclu aux faits suivants :

Les modèles neuronaux statiques « statistiques » RBFN et GRNN ont été de loin plus performants que le modèle statique « numérique » FFNN. Les RMSE étaient respectivement de $8.49e-16$, $6.98e-13$ contre 0.63. Ceci serait attaché au fondement « statistique » de ces modèles.

Le modèle dynamique TDNN s'est avéré presque aussi concurrentiel que les modèles statiques « statistiques ». Son RMSE était de $1.01e-7$

Bien que le modèle récurrent NNARMA (2, 2,1) ait pris en compte, dans son architecture et ses calculs, les résidus retardés e_{t-1} , ses résultats se sont montrés relativement moins précis que prévus (RMSE=0.46). Ceci serait lié aux nombres encore modestes de ses retards.

Globalement, les modèles neuronaux se sont révélés plus compétitifs que le modèle statistique SARIMA.

D'autre part, pour pallier au côté « boîte noire » des modèles neuronaux, les modèles FL (Mahabir, 2003) ont été à leur tour prospectés. Basés sur la théorie des ensembles flous et la théorie des possibilités, ces derniers prennent en considération les données linguistiques (trop fort, assez fort,...) et ainsi sont proches du langage et raisonnement humains. Malheureusement, leurs performances ont été relativement faibles (RMSE=1.00). Plusieurs

raisons pourraient être invoquées : réglage manuel non réussi: mauvaise spécification, architecture non satisfaisante du modèle (partition sous optimale, nombre insuffisant des variables flous, fonction d'appartenance inefficace, ...), secteur d'application inapproprié (la macro-économie semble ne pas conforter ce modèle),...

En conclusion de notre travail, nous affirmons que les modèles numériques et a-théoriques ANN, tant statiques que dynamiques, ont témoigné, en pratique, de leur supériorité en prévisibilité relativement au modèle statistique SARIMA (p, d, q)(s, P,D, Q) et au modèle FL. Curieusement parmi les modèles ANN, les modèles statiques « statistiques » GRNN et RBFN ont été les variantes les plus performantes et le modèle le plus étoffé NNARMA le moins compétitif.

La suprématie des modèles ANN relativement aux modèles SARIMA et FL conforte l'hypothèse de précellence des modèles neuronaux à laquelle nous nous sommes ralliés.

En conséquence de notre étude, les modèles ANN constituent un meilleur outil de modélisation et de prévision de l'inflation à proposer aux décideurs algériens. Cependant, beaucoup de travail reste à faire pour surmonter les limites des modèles neuronaux (modèles à boîte noire, à variables numériques seulement, gourmandise en données, problèmes de mise au point,...). Ainsi, nous envisageons de travailler prochainement dans deux principales directions: les modèles hybrides statistiques (SARIMA/ANN) et hybrides logiques (ANFIS, SANFIS,...).

Les modèles hybrides statistiques (S)ARIMA/ANN (Khashei, 2009) seront explorés dans le but de cumuler les avantages de chacun des modèles neuronaux et statistiques. Une multitude de variantes est présentée depuis quelques années en littérature : ARMA-RBF [Wedding and Cios,1996]³⁷, SARIMA-BP [Tseng& al.,2002]³⁸, ARIMA-ANN [Zhang , 2003]³⁹,.....

D'autre part, pour pallier au côté « boîte noire » des modèles ANN et aux problèmes de détermination des règles et des index (des fonctions d'appartenance) appropriés des modèles FL, trois grands types de combinaison des modèles ANN aux modèles FL sont proposés en

³⁷Wedding (D.K.), Cios (K.J.), *Time series forecasting by combining RBF networks, certainty factors, and Box-Jenkins* , p.149, 1996

³⁸Tseng (F.M.), Yu (H.C.), Tseng (G.H.), *Combining neural network model with seasonal time series ARIMA model*, p. 71, 2002

³⁹Zhang (G.P.), *Time series forecasting using hybrid ARIMA and neural network model*, p.159, 2003

littérature : les systèmes neuro-flous NFS (Shi& Mizumoto, 2000, ...), les systèmes flous_neuronaux FNS (Lin & Lee, 1996,...) et les systèmes hybrides HNFS (ARIC [Berenji⁴⁰ 1992], GARIC [Bersini⁴¹ 1993], NNDFR [Takagi&Hayashi⁴²1991], ANFIS[Jang,1993], SANFIS [Lee,2000], NEFCLASS, NEFCON,...

Nous nous proposons de prospecter, dans un prochain travail, quelques modèles parmi ces derniers.

⁴⁰Berenji (R.H.). *A reinforcement learning-based architecture for fuzzy logic control*. International Journal of Approximate Reasoning, Vol. 6, Issue 2,1992

⁴¹Bersini (H), Nordvik (J.P) & Bonarini (A.). *A simple direct adaptive fuzzy controller derived from its neutral equivalent*, *Proceedings of 2nd IEEE International Conference on Fuzzy Systems*, Vol. 1, pp. 345-350,1993

⁴²Takagi (H.) & Hayashi (I.). *NN-driven fuzzy reasoning*. International Journal of Approximate Reasoning, Vol. 5, Issue 3,1991

BIBLIOGRAPHIE

Abrahart (R.J.), Kneale (P.E.) & See (L.M.). Neural networks for hydrological modeling, Taylor & Francis, 83-85, 2004.

Babu (N.R.), Babu (C.B.A.), Reddy (D.P.) & al. *Comparison of ANFIS and ARIMA model for weather forecasting*, *India journal of science and technology*, 8(S2):70-73, 2015.

Baeuer (M.M.). *General regression neural network for technical use*, thesis, university of Wisconsin-Madison, 1995.

Beccali (G.), Kaluria (M.) & Cullota (S.) & al. *A web-based autonomous weather monitoring system of the town of Palermo and its utilization for temperature nowcasting*, in *Computational science and its applications*, ICCSA 2008, Italy, Springer, 2008.

Binner (J.M.), Tino (P.) & Tepper (J.) & al. *Does money matter in forecasting inflation?*, *Research Division, Federal Reserve Bank of St. Louis*, working paper, St Louis, 2009-030B.

Binner (J.M.), Elger (C.T.), Nilson (B.) & al. *Predictable non-linearities in U.S. inflation*, *Economics Letters*, 93(3):323-328, 2006.

Box (G.E.P.) & Jenkins (G.M.). *Time Series analysis, forecasty and control*, Prentice Hall, Englewood Cliffs, 1976.

Buckley (J.J.) & Eslami (E.). *Introduction to fuzzy logic and fuzzy sets*, Springer- Velag, p.5-52, 2002.

Cacoullos (T.). Estimation of multivariate density, *Ann. Inst. Statist. Math.*, 18(2): 179-189, 1966.

Chong (E.K.P.) & Zak (S.H.). *An introduction to optimization*, John Wiley, 127, 2011.

Chow (T.W.S.) & Cho (S.Y.). *Neural network and computing: Learning algorithms and applications*, Imperial College Press, 152, 2007.

Cloette (I.) & Zurada (J.M.). *Knowledge-based neurocomputing*, MIT Press, 92, 1997.

Connor(J.T.), Martin(R.D.)&Atlas (L.E.). Recurrent neural networks and Robust time series prediction, IEEE Transactions on neural networks 5(2): 240-253, march 1994.

Dai (J.), Liu(X.), Zhang(S.) and al. Continuous neural decoding method based on general regression neural network, International Journal of Digital Technology and its Applications, 4(8), 216-220, Nov 2010.

Dasgupta (P.). Course '*Artificial Intelligence: BP learning*', IIT Kharagpur, India, 2004.

Deepa (S.N.), Sivanandam (S.N.) &Sumathi(S.). *Introduction to neural networks using matlab 6.0*, Tata McGraw Hill Education, 213, 2006.

Djuravic (N.), Domazet (M.) Stricevic (R.) & al. Comparison of groundwater level models based on ANN and ANFIS, Scientific World Journal, 2015.

Duzgun(R.). *GRNN for inflation forecasting*, International journal of research of finance and economics, (51):59-70, 2010.

Elman (J.L.). Finding structure in time, cognitive science 14(1):179-211, 1990.

Engle (R.F.). Autoregressive conditional heteroscedasticity with estimates of variance of united kingdom inflation, 50(4) : 987-1007, 1982.

Eyden (Robert J.V.). The application of neural network in the forecasting of share prises, Finance & Technology Publishing, 1996.

Fa-Long (L.). *Digital Front-End in wireless communications and broadcasting:circuits and signal processing*, Cambridge University Press, 200, 2011.

Fine(T.L.). *Feedforward neural network methodology*, Springer, 1999.

Gajarati(D.N.). '*Basic econometrics*', McGraw-Hill Education, India, 2004.

Garibaldi(J.)&Lotfi(A.).*Applications and scientific soft computing*, Springer, 34, 2004.

Gharlegghi(B.)&Nor(A.H.S.M.). *Application of dynamic models for exchange rate prediction*,international journal of innovation, management and technology, 2(6):459-464, 2011.

Ghodsi(R.), Zakarinia(M.S.) &Jokar(M.). *NN and FL model for forecasting short term price in Ontario electricity market*, 41international conference on computers and industrial engineering, 2011.

Graupe (D.). Principles of artificial neural networks, World Scientific, 2007.

Griva(I.), Nash (S.G.)&Sofer(A.).*Linear and non linear optimization*, Siam, 402, 2009.

Gupta (M.M.), Jin (L.) &Homma(N.). *Static and dynamic neural networks: from fundamentals to advanced theory*, John Willey & Sons, 245, 2003.

Hanan(S.A.),Manza (R.R.) and Ramteke (R.J.). GRNN and RBF for heart disease diagnostic, International Journal of Computer Applications, 7(13):7-13, oct 2010.

Hassoun (M.H.).*Fundamentals in artificial neural networks*, MIT Press, 274, 1995.

Howlet (R.J.)& Jain (L.C.).*Radial Basis Function Networks 2: new advances in design*, Physica-Verlag, 2001.

Hu (T.F.), Luja (I.G.), SU (H.C.) and al. *Forecasting inflation underglobalization with ann-based thin and thick models*, WCECS, San Francisco, USA, Oct 2007.

Hurlin (C.). *Cours d'économétrie appliquée: séries temporelles*, université Paris Dauphine, chapitre 1-4, 2004.

Jang (J.-S.R.). *Adaptive Network Fuzzy Inference System*, IEEE Journal, 23(3):665-685, 1993.

Jang(J.-S.R.)& Sun (C.T.).*Functional equivalence between RBFN and FLS*, IEE Trans Neural Networks. 4(1):156-9, 1993.

Jang(J.-S.R.). *Rule extraction using generalized neural network*, proc. Of the 4th IFSA world congress, volume of Artificial Intelligence, 82-86, 1991

Jordan(M .I).Attractor dynamics and parallelism in connectionist sequential machine, Proc. cognitive science, 1986.

Karbasi (A.) Laskukalayeh(S.S.)&Fahimifard(M.).*Comparaison of NNARX, ANN and ARIMA: Techniques to poultry retail price forecasting*, University of Zobol (Iran), 6, 2009.

Khashei (M.), Bijari (M.), Gholam (A.) et al.Improvement of Auto-regressive Integrated Moving Average models using Fuzzy logic and Artificial Neural Networks (ANNs), *Neurocomputing*, 72: 956-967, 2009.

Kirchgassner (G.), Volters (J.R.)&Hassler (U.).*Introduction to modern time series*, Springer, 127-135, 2012.

Kolen (J.F.) & Kremer (S.C.).A field guide to dynamical recurrent networks, IEEE Press, 34, 2001.

Kuamr (K.R.) &Kalvathi (M.S.).*ANN-ANFIS based forecast model for predicting PV and wind energy generation*, world congress, 1, London U.K., 2016.

Lang (K.J.), Waibel (A.) &Hinton (G.E.).A time delay neural network for isolate word recognition, *Neural Networks*, 3(1): 23-43, 1990.

Laxmidhar (B.). Course '*Recurrent Networks: Back Propagation Through Time*', INPTEL, India, 2008.

Lee (G.) &Wang(J.-S.). *SANFIS: structure and learning*, IEE/RSJ, International Conference on intelligent Robots and System, 2000.

Likothanassis (S.D.) &Demiris (E.N.).ARMAX model identification with unknown process order and time-varying parameters, 175-178, in *Signal analysis and prediction* Springer, 1998.

Lin (C.T.) and Lee(G.).*Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*.Ed. Prentice Hall, 1996

Liu (B.).Web data mining:exploring hyperlinks, contents and usage data, Springer, 2011.

Ljung (L.).*System Identification:theory for user*, Prentice Hall, Englewood Cliffs, New jersey, 73, USA, 1987.

Lontaritis(I.J.) & Billings(S.A.).Input output parametric models for non-linear systems. Part I: deterministic non-linear systems, *International Journal of Control* 41(2): 303-328, 2007.

Madsen (H.).Time series analysis and forecasting, Chapman & Hall/CRC, USA, 2008.

Mahabir (C.).*Application of Fuzzy Logic to forecast seasonal runoff*, *Hydrological Process*, 17(1): 3749-3762, 2003.

McCulloch (W.)& Pitts (W.).*A logical calculus of the ideas immanent in nervous activity*, *Bulletin of Mathematical Biophysics* 5, 115-133, 1943.

McNelis (P.) &McAdam (P.).*Forecasting inflation with thick models and neural network*, Working Paper Series, nr 352, Apr. 2004.

Medsker(L.) & Jain(L.C.).*Recurrent Neural Networks: Design and Applications*, Taylor& Francis, 2010.

Mehri(M.). *A comparison of NN models, FL and MLR for prediction of hatchability*, RCSDA, Zaboluniversity, Iran, 2013.

Mehrotra (K.), Mohan(C.K.) &Ranka(S.).*Elements of artificial neural networks*, MIT Press, 71, 1997.

Mehrotra(K.), Mohan(C.K.)&Ranka(S.).*Elements of ANN*, M.I.T. Press, 4-7, 1996.

Moshiri (S.).*Forecasting inflation using econometrics and ANN models*, Ph.D. thesis, Manitoba university, Canada, Nov.1997.

Moshiri (S.) & Cameron (N.). *Neural network versus econometrics models in forecasting inflation*, *Journal of forecasting*, 19(3):201-207, 2000.

Nakamura (E.).*Inflation forecast using neural network*, *Economics Letters*, 86(3):373-378, 2005.

Nelles (O.).*Non linear system identification: from classical approaches to neural networks and fuzzy models*, Springer, 143, 2001.

Neves (L.D.). *Forecasts inflation and NARMAX models*, VII encontro brasileiro de finanças, Brasil, 2007.

Ngo(T.H.D.)&Bros (W.).*The Box Jenkins methodology for time series models*,SAS global forum, paper 454, 2013.

Nguyen(H.T.) &Walker(E.A.). A first course in fuzzy logic, Chapman & Hall, p.6-36, 2006.

Norgaard (M.). Neural network for modeling and control of dynamic system: a practionner..., Springer, 19-20, 2003.

Norgaard (Magnus). *Neural Network Based System Identification version 2*, Technical University of Denmark, 2000.

ONS.*Indices des prix à la consommation*, 158, 161, 173, 178, janv.2009; avril 2009; avril 2010; sept.2010.

ONS.*Indices des prix à la consommation*, Collections statistiques, 137(4), 2008.

Papert (S.) &Minsky (M.).*Perceptrons, An introduction to Computational Geometry*, M.I.T. Press, Cambridge, 1969.

Parzen (E.).On estimation of probability density function and mode, *Annals of Mathematical Statistics*, 33(1):1065-1076, 1962.

Poo (J.M.R.).*Computer-aided introduction to econometrics*, Springer, 225, 2003.

Prankratz (A.). *Forecasting with dynamic regression models* , John Wiley & Sons, New York, 1991, published on line Jan 2012.

Priddy (K.L.) & Keller (P.E.).*Artificial neural network: an introduction*, SPIE Press, 107-109, 2005.

Rahmati (S.H.), Hadda (O.B.), Sedghi (H.) &al.A comparison of ANFIS, ANN, ARMA and Multivariate regression methods for urban water *consumption forecasting ...*, *India J. Sci.*, 7(1):870-880, 2014.

Refenes (A.N.), Zapranis (A.) & Francis (G.). *Stock performance using neural network a comparative study with regression models*, *Neural Networks*, 7(2): 375-388, 1994.

Rosenblatt(F.). *The perceptron: a probabilistic model of information storage and organization in the brain*, *Psychological Review* 65(6): 386-408, 1958.

Rumelhart (D.E.), Hinton (G.E.), & Williams (R.J.). *Learning representations by back-propagation errors*, *nature* 323():533-536, 1986.

Satish (K.). *Neural Networks: a classroom approach*, Tata McGraw-Hill Education 323, 2004.

Sengupta (S.). Course '*Neural Network and Application ;BP algorithm*', IIT Kharagpur, India, 2003.

Shi (Y.) & Mizumoto (M.). *A new approach of neurofuzzy learning algorithm for tuning fuzzy rules*. *Fuzzy sets and systems*, 112(1):99–116, 2000

Sims (Christopher.A.). *Macroeconomics and reality*, *Econometrica*, 48(1):1-48, janv.1980.

Singh(T.K.K.) & Nath(S.K.). *Maximum credible earthquake prediction, a neural network approach* in international workshop on earthquake hazards and mitigation, India, dec 2007.

Sivanandam (S.N.), Sumathi (S.) & Deepa(S.N.). *Introduction to neural networks using matlab 6.0*, Tata McGraw-Hill Education, 185-186, India, 2006.

Sivanandam(S.N.), Sumathi(S.) and Deep(S.N.). *Introduction to fuzzy logic using matlab*, p.113-149, 2007.

Specht (D.F.). *A general Regression Neural Network*, *IEEE transactions on neural network* , 2(6): 568-576, Nov 1991.

Specht (D.F.). *Probabilistic Neural Networks*, *Neural Networks*, 3 : 109-118, 1990.

Stock(J.H.) & Watson(M.W.). *Forecasting inflation*, *Journal of monetary economics*, 44(2): 293-335, 1999.

Thiaw(L.). *Identification de systèmes dynamiques non linéaires par réseaux de neurones et multimodèles*, thèse doctorat, Université Paris XII , 42-43, 2008.

Tseng (F.M.), Yu (H.C.), Tseng (G.H.). Combining neural network model with seasonal time series ARIMA model, *Technol. Forecast. Soc. Change*, 69: 71-87, 2002.

Ucenic(C.I.),&Atsalakis (Cr).*Forecasting CPI using a NN with FS*, 13th International Conference: Applied Stochastic Modelisation Data Analysis ASMADA, Vilnius, LITHUANIA, 2009

Wan(E.A.).FIR neural network with application in time series prediction, Stanford university, thesis of Phd, Stanford University, 1993.

Wang (W.).*Stochasticity ,nonlinearity and forecasting of streamflow processes*, IOS Press, 2006.

Wedding (D.K.), Cios (K.J.).Time series forecasting by combining RBF networks, certainty factors, andBox-Jenkins, *Neurocomputing*, 10: 149-168, 1996.

Werbos (P.J.).Back propagation through time: what it does and how to do it, *IEE*, 78 (10): 1550, October 1990.

Widrow (B.) & Hoff (M.).*Adaptative switching circuits*, IRE WESCON Convention Record, New York, 96-104, 1960.

Wooldridge (J.M.).Introductory to econometrics, a modern approach, South-Western Cengag Learning, USA, 2009.

Yaffee (R.A.) &McGee(M.).*An introduction to time series analysis and forecasting*, with applications of SAS and SPSS, Academic Press, New York, 2000.

Yegnanarayana (B.). Artificial neural network, PHI learning Pvt Ltd, 278, Tata McGraw-hill, New Delhi, 2004.

Zadeh(L.A.). *Fuzzy sets*, *information and control*, 8: 338-353, 1965.

Zhang (G.P.). Time series forecasting using hybrid ARIMA and neural network model, *Neurocomputing*, 50: 159-175, 2003

ANNEXES

ANNEXE A Test de saisonnalité de la série Yt

Le comportement de l'ACF aux périodes multiples de 12 confirme l'existence d'une saisonnalité

Tableau1. 24 Correlogramme de Yt source : Eviews

Date: 08/05/21 Time: 13:51

Sample: 2002M01 2020M12

Included observations: 228

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.083...	0.083...	1.6172...	0.203...
		2 -0.22...	-0.23...	13.149...	0.001...
		3 -0.22...	-0.19...	25.051...	1.506...
		4 -0.15...	-0.19...	30.683...	3.551...
		5 0.097...	0.021...	32.919...	3.905...
		6 0.132...	0.007...	37.060...	1.714...
		7 0.096...	0.058...	39.278...	1.729...
		8 -0.06...	-0.04...	40.142...	3.013...
		9 -0.08...	0.003...	41.802...	3.570...
		1... -0.11...	-0.09...	44.833...	2.329...
		1... 0.109...	0.116...	47.753...	1.579...
		1... 0.265...	0.202...	64.921...	2.819...
		1... -0.08...	-0.11...	66.829...	3.060...
		1... -0.02...	0.112...	66.990...	6.735...
		1... -0.15...	-0.10...	73.282...	1.154...
		1... -0.05...	0.008...	74.054...	1.925...
		1... 0.171...	0.096...	81.349...	2.206...
		1... 0.036...	-0.06...	81.679...	4.348...
		1... 0.099...	0.122...	84.177...	3.506...
		2... -0.07...	-0.04...	85.629...	4.263...
		2... -0.11...	-0.01...	88.880...	2.515...
		2... -0.21...	-0.21...	100.44...	5.397...
		2... 0.102...	0.058...	103.15...	3.995...
		2... 0.274...	0.133...	122.54...	3.441...
		2... 0.032...	-0.01...	122.81...	6.994...
		2... -0.05...	-0.02...	123.62...	1.143...
		2... -0.22...	-0.05...	136.58...	1.110...
		2... -0.05...	-0.02...	137.28...	2.220...
		2... 0.122...	0.051...	141.23...	1.110...
		3... 0.064...	-0.02...	142.34...	1.110...
		3... 0.092...	0.013...	144.61...	1.110...
		3... -0.02...	0.071...	144.82...	2.220...
		3... -0.16...	-0.11...	152.04...	0
		3... -0.18...	-0.07...	161.15...	0
		3... 0.092...	-0.03...	163.48...	0
		3... 0.229...	0.101...	177.90...	0

ANNEXE B Test de saisonnalité de la série YTSA

La disparition des pics de l'ACF aux périodes multiples de 12 confirment la disparition de la saisonnalité.

Tableau1. 25 Correlogramme d' YTSA source : Eviews

Date: 08/14/21 Time: 18:38

Sample: 2002M01 2022M12

Included observations: 228

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob	
		1	0.040	0.040	0.3745	0.541
		2	-0.18...	-0.19...	8.6697	0.013
		3	-0.10...	-0.09...	11.212	0.011
		4	-0.09...	-0.13...	13.497	0.009
		5	0.022	-0.01...	13.610	0.018
		6	0.088	0.035	15.432	0.017
		7	-0.01...	-0.04...	15.497	0.030
		8	-0.00...	0.010	15.503	0.050
		9	0.075	0.081	16.832	0.051
		1...	0.002	0.010	16.833	0.078
		1...	0.111	0.147	19.796	0.048
		1...	0.011	0.023	19.823	0.071
		1...	-0.20...	-0.14...	29.983	0.005
		1...	0.047	0.093	30.532	0.006
		1...	-0.02...	-0.08...	30.671	0.010
		1...	0.045	0.051	31.172	0.013
		1...	0.137	0.084	35.832	0.005
		1...	-0.04...	-0.05...	36.371	0.006
		1...	-0.03...	0.042	36.628	0.009
		2...	-0.02...	-0.05...	36.841	0.012
		2...	0.029	0.058	37.057	0.017
		2...	-0.12...	-0.15...	40.899	0.008
		2...	0.101	0.108	43.502	0.006
		2...	0.035	0.028	43.813	0.008
		2...	-0.04...	-0.06...	44.417	0.010
		2...	-0.01...	-0.03...	44.512	0.013
		2...	-0.11...	-0.11...	47.840	0.008
		2...	0.057	0.031	48.698	0.009
		2...	0.104	0.074	51.548	0.006
		3...	-0.01...	-0.00...	51.572	0.008
		3...	-0.03...	-0.00...	51.968	0.011
		3...	0.027	0.057	52.165	0.014
		3...	-0.04...	-0.04...	52.739	0.016
		3...	-0.07...	-0.07...	54.441	0.014
		3...	0.094	0.037	56.836	0.011
		3...	0.002	0.057	56.838	0.015

ANNEXE C Test Q de Ljung-Box du modèle SARIMA (22,0,4) (12,2,0,2)

Tableau1. 26 Correlogramme du modèle 1 SARIMA source : Eviews

Date: 11/14/21 Time: 12:23

Sample: 2002M01 2020M12

Included observations: 228

Q-statistic probabilities adjusted for 11 ARMA terms

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.081...	0.081...	1.5510...	
		2 -0.09...	-0.09...	3.5276...	
		3 -0.01...	0.002...	3.5741...	
		4 -0.02...	-0.03...	3.7422...	
		5 -0.04...	-0.03...	4.1686...	
		6 0.000...	0.002...	4.1687...	
		7 -0.01...	-0.02...	4.2172...	
		8 0.055...	0.058...	4.9480...	
		9 0.065...	0.050...	5.9750...	
		1... 0.024...	0.024...	6.1188...	
		1... -0.03...	-0.03...	6.4679...	
		1... 0.060...	0.074...	7.3454...	0.006...
		1... -0.06...	-0.07...	8.3860...	0.015...
		1... 0.068...	0.103...	9.5262...	0.023...
		1... 0.070...	0.043...	10.737...	0.029...
		1... 0.066...	0.075...	11.830...	0.037...
		1... 0.043...	0.041...	12.292...	0.055...
		1... -0.08...	-0.09...	14.068...	0.049...
		1... -0.04...	-0.00...	14.483...	0.070...
		2... -0.05...	-0.07...	15.234...	0.084...
		2... 0.077...	0.104...	16.751...	0.080...
		2... 0.004...	-0.03...	16.757...	0.115...
		2... -0.04...	-0.03...	17.342...	0.137...
		2... 0.000...	-0.02...	17.342...	0.184...
		2... 0.034...	0.025...	17.651...	0.223...
		2... -0.08...	-0.10...	19.327...	0.199...
		2... -0.06...	-0.03...	20.519...	0.197...
		2... 0.028...	0.033...	20.739...	0.238...
		2... 0.031...	-0.00...	21.001...	0.279...
		3... 0.021...	0.028...	21.129...	0.329...
		3... 0.001...	-0.04...	21.130...	0.389...
		3... -0.01...	0.029...	21.165...	0.448...
		3... 0.041...	0.018...	21.632...	0.482...
		3... -0.03...	0.007...	21.882...	0.527...
		3... -0.01...	0.017...	21.948...	0.582...
		3... 0.014...	0.013...	22.006...	0.635...

ANNEXE D Test Q de Ljung-Box du modèle SARIMA (11,0,27) (12, 0,0,0)

Tableau1.27 Correlogramme du modèle 2 SARIMA source : Eviews

Date: 11/14/21 Time: 12:25

Sample: 2002M01 2022M12

Included observations: 228

Q-statistic probabilities adjusted for 8 ARMA terms

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.096...	0.096...	2.1634...	
		2 -0.00...	-0.00...	2.1634...	
		3 0.017...	0.018...	2.2375...	
		4 0.017...	0.014...	2.3131...	
		5 -0.03...	-0.03...	2.5258...	
		6 0.020...	0.026...	2.6230...	
		7 -0.03...	-0.03...	2.8820...	
		8 0.019...	0.028...	2.9751...	
		9 0.073...	0.069...	4.2595...	0.039...
		1 0.035...	0.021...	4.5611...	0.102...
		1 0.001...	-0.00...	4.5614...	0.206...
		1 0.026...	0.021...	4.7319...	0.315...
		1 0.012...	0.007...	4.7722...	0.444...
		1 0.098...	0.099...	7.1438...	0.307...
		1 0.015...	-0.00...	7.2001...	0.408...
		1 0.058...	0.063...	8.0524...	0.428...
		1 0.050...	0.037...	8.6785...	0.467...
		1 -0.08...	-0.10...	10.501...	0.397...
		1 -0.00...	0.019...	10.506...	0.485...
		2 -0.03...	-0.05...	10.832...	0.543...
		2 0.073...	0.094...	12.191...	0.512...
		2 0.015...	-0.00...	12.251...	0.586...
		2 0.102...	0.089...	14.964...	0.454...
		2 0.031...	0.011...	15.214...	0.508...
		2 -0.03...	-0.06...	15.480...	0.560...
		2 -0.00...	0.004...	15.480...	0.628...
		2 0.042...	0.041...	15.950...	0.660...
		2 0.045...	0.049...	16.489...	0.685...
		2 0.096...	0.095...	18.956...	0.587...
		3 -0.01...	-0.05...	19.042...	0.642...
		3 -0.02...	-0.03...	19.230...	0.687...
		3 0.027...	0.030...	19.438...	0.728...
		3 -0.04...	-0.08...	20.079...	0.742...
		3 -0.06...	-0.01...	21.304...	0.726...
		3 0.115...	0.121...	24.959...	0.576...
		3 0.033...	0.005...	25.264...	0.613...

ANNEXE E Test Q de Ljung-Box du modèle SARIMA (47,0,4) (12,2,0,2)

Tableau1. 28 Correlogramme du modèle 3 SARIMA source : Eviews

Date: 11/14/21 Time: 12:26

Sample: 2002M01 2020M12

Included observations: 228

Q-statistic probabilities adjusted for 13 ARMA terms

Autocorrelation	Partial Correlation	AC	PAC	Q-Stat	Prob
		1 0.069...	0.069...	1.1295...	
		2 -0.04...	-0.05...	1.6212...	
		3 -0.01...	-0.00...	1.6623...	
		4 -0.06...	-0.06...	2.5274...	
		5 -0.09...	-0.08...	4.5247...	
		6 -0.02...	-0.01...	4.6296...	
		7 0.012...	0.005...	4.6677...	
		8 0.078...	0.071...	6.1408...	
		9 0.093...	0.076...	8.2477...	
		1... 0.031...	0.018...	8.4796...	
		1... -0.07...	-0.07...	9.7502...	
		1... 0.052...	0.075...	10.411...	
		1... -0.02...	-0.02...	10.608...	
		1... 0.076...	0.108...	12.031...	0.000...
		1... -0.02...	-0.04...	12.181...	0.002...
		1... 0.057...	0.064...	13.004...	0.004...
		1... 0.060...	0.043...	13.924...	0.007...
		1... -0.09...	-0.10...	16.201...	0.006...
		1... 0.046...	0.089...	16.746...	0.010...
		2... -0.04...	-0.06...	17.276...	0.015...
		2... 0.041...	0.076...	17.722...	0.023...
		2... 0.032...	-0.00...	17.987...	0.035...
		2... -0.03...	-0.03...	18.234...	0.051...
		2... -0.00...	-0.00...	18.234...	0.076...
		2... 0.050...	0.048...	18.888...	0.091...
		2... -0.09...	-0.11...	21.018...	0.072...
		2... -0.02...	0.031...	21.137...	0.098...
		2... 0.082...	0.059...	22.917...	0.085...
		2... 0.048...	0.023...	23.543...	0.099...
		3... -0.00...	0.009...	23.549...	0.132...
		3... 0.041...	-0.00...	24.012...	0.154...
		3... -0.03...	0.017...	24.325...	0.183...
		3... -0.00...	-0.02...	24.325...	0.228...
		3... -0.00...	0.029...	24.343...	0.276...
		3... 0.041...	0.054...	24.803...	0.306...
		3... 0.012...	-0.00...	24.847...	0.358...

NB : Le modèle parcimonieux est Sarima(11,0,4)(12,2,0,2) $r^2=0.23$

ANNEXE F Performance du modèle Sarima

Tableau1. 29 Performance du modèle SARIMA source : Eviews

Forecast: YTSAF_SARIMA8	
Actual: YTSA	
Forecast sample: 2006M01 2020M12	
Included observations: 180	
Root Mean Squared Error	0.819791
Mean Absolute Error	0.653753
Mean Abs. Percent Error	174.8758
Theil Inequality Coefficient	0.607835
Bias Proportion	0.001704
Variance Proportion	0.426056
Covariance Proportion	0.572241

Titre: Prédiction des IPC par les réseaux de neurones artificiels: le cas algérien

Résumé:

Dans une tentative de modélisation des IPC en Algérie, notre travail consiste à étudier, du point de vue fiabilité, les modèles neuronaux (ANN) relativement aux modèles traditionnels (SARIMA) puis aux récents modèles de logique floue (FL). Les critères retenus pour la performance des prévisions sont le RMSE, le MAE et R^2 . Les résultats révèlent que les modèles de prévisions neuronaux (ANN) sont les plus compétitifs et paradoxalement, parmi ces derniers, les modèles statiques statistiques élémentaires (GRNN, RBFN) sont les plus concurrentiels.

Mots clés : RNA, SARIMA, IPC, FFNN, GRNN, RBFN, TDNN, NNARMAX

Title : CPI forecast by artificial neural networks: the case of Algeria

Abstract ;

In attempt to model the CPI in Algeria, our work consists of studying, from reliability point of view, neural networks (ANN) relative to traditional models (SARIMA) then to recent fuzzy logic models (FL). The criteria used for forecast performance are RMSE, MSE and R^2 . The results reveal that the neural forecast networks (ANN) are the most competitive and paradoxally among the latter, elementary statistical static models (GRNN, RBFN) are the most efficient

Key words: ANN, SARIMA, CPI, FFNN, GRNN, RBFN, TDNN, NNARMAX