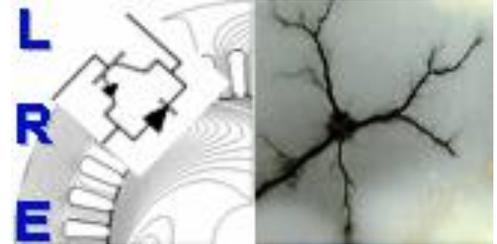


REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département d'ELECTROTECHNIQUE
Laboratoire de Recherche en Electrotechnique

Mémoire de fin d'études pour l'obtention du diplôme d'
Ingénieur d'Etat en Electrotechnique

Intitulé

**Prédiction du comportement d'un isolateur à l'aide des Réseaux de
Neurones Artificiels**

Présenté par
Ayyoub GHERASSI et Abdelhak HASSANI

Sous la direction de **Dr. A. K. Chaou**
Soutenu publiquement le 3 juillet 2019

Membres du Jury

Président : A. Boubakeur, Professeur à l'ENP
Rapporteur : A. K. Chaou, Maitre de Conférence B à l'ENP
Examineur : M. Tegar, Professeur à l'ENP
Invités :

ENP 2019

Laboratoire de Recherche en Electrotechnique (LRE) - Ecole Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique



Département d'ELECTROTECHNIQUE
Laboratoire de Recherche en Electrotechnique

Mémoire de fin d'études pour l'obtention du diplôme d'
Ingénieur d'Etat en Electrotechnique

Intitulé

**Prédiction du comportement d'un isolateur à l'aide des Réseaux de
Neurones Artificiels**

Présenté par
Ayyoub GHERASSI et Abdelhak HASSANI

Sous la direction de **Dr. A. K. Chaou**
Soutenu publiquement le 3 juillet 2019

Membres du Jury

Président : A. Boubakeur, Professeur à l'ENP
Rapporteur : A. K. Chaou, Maitre de Conférence B à l'ENP
Examineur : M. Teguar, Professeur à l'ENP
Invités :

ENP 2019

Laboratoire de Recherche en Electrotechnique (LRE) - Ecole Nationale Polytechnique (ENP)
10, Avenue des Frères Oudek, Hassen Badi, BP. 182, 16200 El Harrach, Alger, Algérie

Dédicace

Je dédie ce projet :

A ma chère mère,

A mon cher père,

Qui n'ont jamais cessé, de formuler des prières à mon égard, de me soutenir

Et de m'épauler pour que je puisse atteindre mes objectifs.

A mes frères, Abdelmadjid et Youcef

A ma chère sœur, Amina,

Pour ses soutiens moraux et leurs conseils précieux tout au long de mes études.

A mon cher grand-père,

Qui je souhaite une bonne santé.

A mon binôme, Abdelhak

Pour son entent et sa sympathie.

A mes chères ami (e)s,

Pour leurs aides et supports dans les moments difficiles.

A tous les Gherassi et Fas,

A tous mes autres ami(e)s,

A tous ceux que j'aime et ceux qui m'aiment.

Ayyoub.

Dédicace

Je dédie ce travail, qui n'aura jamais pu voir le jour sans les soutiens indéfectibles et sans limite de mes chers parents qui ne cessent de me donner avec amour le nécessaire pour que je puisse arriver à ce que je suis aujourd'hui. Que dieux vous protège et que la réussite soit toujours à ma portée pour que je puisse vous combler de bonheur.

Je dédie aussi ce travail à :

*Mon binôme Ayyoub et à mon chère ami Saad Eddine,
À mes sœurs, mes frères, tous mes Amis (es) qui m'Aiment
et qui m'Apprécient À toutes celles et tous ceux qui me
sont chers.*

Abdelhak

Remerciements

En premier lieu, nous remercions Dieu tout puissant de nous avoir donné le courage, la volonté et la patience afin de mener ce modeste travail à terme.

*En second lieu, Nous tenons à remercier notre directeur de mémoire, **A. K. Chaou**, Docteur à l'Ecole National Polytechnique d'Alger, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à la réalisation de ce mémoire.*

*Nos remerciements vont également aux **Pr. A. Mekhaldi** et **Pr. M. Tegar**, pour leurs aides et leurs qualités humaines.*

*Nous tenons également à remercier le président **Pr. A. BOUBAKEUR** et les membres du jury pour nous avoir fait l'honneur d'évaluer notre travail.*

Nous tenons également à remercier tous les enseignants ayant contribué à notre formation et ce depuis le tronc commun jusqu'à la dernière année de graduation.

في هذا العمل ، نقترح نموذجًا لشبكة عصبية اصطناعية (RNA) من اجل التنبؤ بقيمة الناقلية للعازل الدائري الملوث دو التوتر المتناوب (AC) 50 هرتز. حيث يتم استخدام قاعدة بيانات مستخرجة من تجارب مختبر التوتر العالي للمدرسة الوطنية المتعددة التقنيات لتدريب عدة أنواع الذكاء الاصطناعي. في الواقع ، سنقوم بتجربة شعاع التردد (قيمة التوافقي الخامس و الثالث من LC وفرق الزاوية بين التوتر المطبق و LC) وشعاع زمني (القيمة العظمى Imax, انحراف معياري STD, القيمة المتوسطة Imoy) كمتغيرات لمختلف أنواع الذكاء الاصطناعي . اولاً ، نقدم الأعمال السابقة وبعد ذلك سنشرح نظرية الذكاء الاصطناعي بمختلف مركباته بصفة دقيقة ، ولقد لاحظنا ان الذكاء الاصطناعي (RNA) فعال كثيراً لحل هذا النوع من المشاكل. ثم ، نقوم باستخراج خصائص الزمنية والترددية المذكورين اعلاه لتيار التسرب باستخدام برنامج MATLAB. و بعد ذلك أجرينا العديد من الاختبارات للتأكد من صحة اختيارنا للمتغيرات وكذلك لإيجاد أفضل RNA الذي يتنبأ قيمة الناقلية بدقة . أبرزت نتائج هذا العمل امكانيات RNAs في التنبؤ بدرجة تلوث العوازل ، خاصة باستعمال خصائص الترددية لتيار التسرب (LC) .

الكلمات المفتاحية : تيار التسرب (LC) ، الشبكة العصبية الاصطناعية (RNA) ، عازل التوتر العالي، تقييم التلوث ، احاطة العوازل الملوثة.

Abstract:

In this work, we will propose an Artificial Neural Network (RNA) model for the prediction of the conductivity value of a 50 Hz AC voltage isolator model. We are going to propose a frequency vector (3rd and 5th harmonic amplitude of the LC and the phase difference between the applied voltage and LC) and temporal vector (**Imax, Imoy and STD**) of LC such as inputs of RNA models. First, we present the previous works and then we will explain the RNA theory, as well as the different RNA types frequently used to solve this type of problems. Then, we extract the temporal and frequency characteristics of the leakage current using MATLAB. The main results of this work have highlighted the potential of the RNA in the prediction of the degree of pollution of insulators, based on the frequency characteristics of LC.

Keywords: Leakage Current, Artificial Neural Network, High Voltage Insulator, Pollution Evaluation, Flashover of polluted insulators.

Résumé :

Dans le présent, nous œuvrons à trouver modèle optimal de Réseau de Neurone Artificiel (RNA) pour la prédiction de la valeur de la conductivité d'un modèle plan d'isolateur sous tension alternative 50 Hz. Pour cela, une base de données est utilisée pour l'apprentissage de plusieurs types de RNA. En effet, nous allons essayer de savoir si les vecteur d'entrée fréquentiel (**amplitude du 3^{eme}, 5^{eme} harmonique du LC et le déphasage entre le fondamentale de la tension appliquée et LC**) et le vecteur temporel (**Imax, Imoy et la déviation standard STD**) peuvent être utilisés comme entrées de notre RNA , En premier lieu, nous présentons les travaux antérieurs et puis nous allons expliciter la théorie des RNA, ainsi que les types de RNA fréquemment utilisées pour résoudre ce type de problèmes. Puis, nous extrayons les caractéristiques temporelles et fréquentielles du courant de fuite en utilisant logiciel MATLAB. Les principaux résultats de ce travail ont mis en avant le potentiel des RNA dans la prédiction de degré de pollution des isolateurs, surtout en utilisant les caractéristiques fréquentielles du signal du courant de fuite.

Mots-Clés : Courant de fuite, Réseau de Neurone Artificiel, Isolateur Haute Tension, Evaluation de la pollution, Contournement des isolateurs pollués.

SOMMAIRE

LISTE DES TABLEAUX

LISTE DES FIGURES

INTRODUCTION GENERALE22

CHAPITRE I

TRAVAUX ANTERIEURS

Introduction	25
I.1 Évaluation du contournement des isolateurs en porcelaine avec décharge ultraviolet à l'aide d'un système d'apprentissage approfondi	25
I.1.1 Paramètre d'entrées d'un RNC	25
I.1.2 Structure d'un RNC	26
I.1.3 Résultats et discussion	27
I.2 Diagnostic en temps réel des isolateurs de lignes aériennes du réseau de distribution de moyenne tension.....	28
I.2.1 Montage pratique proposé.....	28
I.2.2 Méthodes de prédiction proposées	29
I.2.3 Structure de réseau de neurones artificiels.....	31
I.2.4 Test de montage proposé.....	31
I.3 Régularisation bayésienne du réseau de neurones pour prédire le comportement du courant de fuite dans un environnement de brouillard salin.....	32
I.3.1 Réseaux de neurones artificiels utilisés.....	32
I.3.1.1 Modèle RNA non linéaire autorégressif (NAR).....	32
I.3.1.2 RNA basé sur le modèle input-output (I-O).....	33
I.3.1.3 Modèle RNA à entrées exogènes autorégressif (NARX).....	34
I.3.2 Comparaison entre les différents modelés de RNA	36
I.4 Nouvelle méthode d'estimation de degré de pollution d'un isolateur basée sur l'ultrason et réseaux de neurones artificiels	36
I.4.1 Algorithme SSCEV	38
I.4.2 Modèle de réseau de neurones artificiel proposé	39

CHAPITRE II

THEORIE DES RESEAUX DE NEURONES ARTIFICIELS

Introduction	43
II.1 Neurone formel.....	43
II.1.1 Entrées et les poids des connexions	43
II.1.2 Additionneur	44
II.1.3 Fonction d'activation	44
II.1.3.1 Fonction d'activation log-sigmoïde	45
II.1.3.2 Fonction d'activation tan-sigmoïde.....	45
II.1.3.3 Fonction d'activation linéaire.	46
II.2 Les différentes architectures des réseaux de neurones	46
II.2.1 Réseaux de Neurones Non-Bouclés.....	46
II.2.1.1 Perceptron multicouche.....	47
1. Nombre de couche cachée	47
2. Nombre de neurone par couche cachée	48
II.2.1.2 Réseaux de Neurones à Fonctions à Base Radiale.....	49
1. Vecteur d'entrée.....	50
2. Neurones RBF.....	50
3. Nœuds de sortie	51
4. Fonction d'activation des neurones RBF	51
II.2.2 Réseaux de Neurones Bouclés.....	53
II.2.2.1 Réseaux Elman.....	53
II.2.2.2 Réseaux Jordan.....	54
II.3 Apprentissage des Réseaux de Neurones	54
II.3.1 Apprentissage supervisé :	54
II.3.1.1 Algorithme de perceptron	55
II.3.1.2 Algorithme Back-propagation.....	55
II.3.1.3 Algorithme Gradient Descent	56
II.3.1.4 Algorithme Gradient Descent avec un terme de moment	60
II.3.2 Apprentissage non supervisé	61
II.3.3 Différence entre l'apprentissage supervisé et non supervisé	61

II.4	Performance de réseau de neurone artificiel.....	62
II.4.1	Somme des carrés des erreurs (SSE)	62
II.4.2	Erreur quadratique moyenne (MSE).....	62
II.4.3	Ecart quadratique moyen (RMSE).....	63
II.5	Principaux éléments à retenir dans des réseaux de neurones	63
II.6	Réseau neuronal convolutif	63
II.6.1	Différence entre un réseau de neurones et un réseau de neurones convolutif 64	
II.6.2	Différents types de couches d'un RNC	64
II.6.2.1	Couche de convolution.....	64
II.6.2.2	Couche ReLU (Rectified linear unit)	67
II.6.2.3	Couche de pooling.....	68
II.6.2.4	Couche entièrement connectée FC.....	69
II.7	Conclusion.....	70

CHAPITRE III

EXTRACTION DES CARACTERISTIQUES DU COURANT DE FUIE

Introduction	72
III.	72
III.1 Méthode d'acquisitions du courant de fuite	72
III.1.1 Dispositif expérimental	72
III.1.2 Description du modèle d'isolateur étudié.....	73
III.1.3 Essais expérimentaux	75
III.1.3.1 Préparation des solutions polluantes.....	75
III.1.3.2 Méthode de pulvérisation.....	75
III.1.4 Exemple du courant de fuite enregistré.....	76
III.2 Caractéristiques temporelles du courant de fuite.....	76
III.2.1 Valeur maximale	77
III.2.1.1 Mesure de l'amplitude maximale	77
III.2.1.2 Etude de l'amplitude maximale en fonction de la conductivité.....	77

1.	Conductivité 0.01, 0.19, 0.71 et 1.2 mS/cm	77
2.	Conductivité 10.1 mS/cm	79
III.2.2	Valeur moyenne du courant de fuite (RMS)	79
III.2.2.1	Mesure de la valeur moyenne	79
III.2.2.2	Variation de la valeur moyenne en fonction de la conductivité	79
III.2.3	Valeur de la déviation standard (STD).....	80
III.2.3.1	Mesure de la déviation standard (STD)	80
III.2.3.2	Variation de la STD en fonction de la conductivité :	80
III.3	Caractéristiques fréquentielles du courant de fuite.....	81
III.3.1	Sérier de fourrier.....	81
III.3.1.1	Définition	81
III.3.1.2	Transformation de Fourier rapide (FFT)	82
III.3.2	L'amplitude du 3ème et 5ème harmonique du courant de fuite.....	82
III.3.2.1	Analyse spectrale du courant de fuite en fonction de la conductivité	83
1.	Conductivité $\sigma = 10 \mu\text{S/cm}$	83
2.	Conductivité $\sigma = 190, \sigma = 710 \mu\text{S/cm}$	84
3.	Conductivité $\sigma = 1,2 \text{ mS/cm}$	86
4.	Conductivité $\sigma = 10,1 \text{ mS/cm}$	87
III.3.3	Déphasage entre le courant de fuite et la tension appliquée.....	88
III.3.3.1	Mesure du déphasage.....	88
III.3.3.2	Etude du déphasage courant-tension en fonction de la conductivité	88
1.	Conductivité 0.01, 0.19, 0.71 et 1.2 mS/cm	88
2.	Conductivité $\sigma = 10,1 \text{ mS/cm}$	91
III.4	Conclusion	91

CHAPITRE IV

UTILISATION DES RNA POUR PREDIR LA VALEUR DE LA CONDUCTIVITE DE LA POLLUTION A PARTIR DU COURANT DE FUITE

Introduction	93
IV.1 Collection et organisation des données	93
IV.1.1 Partition de la base des données	93
IV.2 Choix des types des réseaux de neurones	95
IV.2.1 Feed-forward back propagation	95
IV.2.2 ELMAN backpropagation	95
IV.3 Choix des paramètres des deux types de RNA	95
IV.3.1 Entrées (Inputs)	95
IV.3.2 Sorties (Outputs)	96
IV.3.3 Fonctions d'entraînement (training fonction)	96
IV.3.3.1 Fonction d'entraînement traingdx (Gradient descent with momentum and adaptive learning rate backpropagation)	96
IV.3.3.2 Fonction d'entraînement Trainbr (Bayesian régularization backpropagation)	97
IV.3.4 Outil d'évaluation de la performance d'un modèle RNA	97
IV.3.4.1 Par une fonction d'erreur	97
IV.3.4.2 Régression linéaire	98
IV.3.4.3 Erreur Relative	99
IV.3.5 Choix de nombre des couches cachées et des neurones de chaque couche 99	
IV.3.6 Initialisation des poids et des biais	99
IV.3.7 Choix de fonction d'activation	99
IV.4 Modèles proposés	108
IV.5 Les performances des configurations proposées	108
IV.5.1 Vecteur fréquentielle	108
IV.5.1.1 Pour une couche cachée	108
1. Erreur d'entraînement	109
2. Erreur de validation	109

3.	Erreur de tests	110
4.	Erreur relative	110
IV.5.1.2	Pour deux couches cachées.....	111
1.	Erreur d'entraînement	111
2.	Erreur de validation	112
3.	Erreur de tests	113
4.	Erreur relative	114
IV.5.2	Vecteur temporel	116
IV.5.2.1	Une seule couche cachée	116
1.	Erreur d'entraînement	116
2.	Erreur de validation	116
3.	Erreur de tests	117
4.	Erreur relative	117
IV.5.2.2	Pour deux couches cachées.....	118
1.	Erreur d'entraînement	118
2.	Erreur de validation	119
3.	Erreur de tests	120
4.	Erreur relative	121
IV.6	Les modèles choisis	122
IV.6.1	Pour le vecteur fréquentiel	122
IV.6.2	Pour le vecteur temporel	123
IV.7	Résultats.....	124
IV.7.1	Résultats du modèle fréquentiel	124
IV.7.1.1	Performance du modèle	124
1.	La fonction d'erreur MSE.....	124
2.	Régression linéaire.....	125
IV.7.1.2	Résultats de prédiction du modèle fréquentiel	127
IV.7.2	Résultats du modèle temporel	128
IV.7.2.1	Performance du modèle	128
1.	La fonction d'erreur MSE.....	128
2.	Régression linéaire.....	129
IV.7.2.2	Résultats de prédiction du modèle temporel	131

IV.8 Comparaison entre les erreurs relatives de l'ensemble de test données par le vecteur fréquentiel et temporel	132
IV.9 Conclusion	133
Conclusion Générale.....	135
Références bibliographies.....	138

LISTE DES TABLEAUX

Chapitre I

Tableau I.1. Les cas possibles de sortie de montage proposé	32
Tableau I.2. Exemple de données d'apprentissage	35
Tableau I.3. Classement des isolateurs pollués en fonction d'ESDD	39
Tableau I.4. Taux de succès de modèle de RNA	40
Tableau I.5. L'influence de l'humidité sur le classement de modèle de RNA entraîné avec une seule valeur de RH	40

Chapitre II

Tableau II.1. Détermination du nombre de couches cachées	48
Tableau II.2. Comparaison entre l'apprentissage supervisé et non supervisé	62

Chapitre III

Tableau III.1. Dimensions du modèle plan étudié	74
Tableau III.2. Dimensions de l'isolateur réel 1512 L	75

Chapitre IV

Tableau IV.1. Exemple d'une matrice d'entrée de type fréquentielle de classe 10 mS/cm ...	96
Tableau IV.2. Les différentes valeurs de conductivités utilisées	96
Tableau IV.3. Configurations proposées pour choisir les fonctions d'activation	100
Tableau IV.4. Les différentes configurations de réseau de neurones utilisées	108
Tableau IV.5. Paramètres finaux du modèle RNA.....	122
Tableau IV.6. Paramètres finaux du modèle RNA	123
Tableau IV.7. Erreur relative en (%) de prédiction obtenue par le modèle (RNA) choisi pour le vecteur fréquentiel.....	127
Tableau IV.8. Erreur relative en (%) de prédiction obtenue par le modèle (RNA) choisi pour le vecteur temporel	131

LISTE DES FIGURES

Chapitre I

Figure I.1. Schéma de câblage de la plateforme expérimentale	25
Figure I.2. Un échantillon typique d'entraînement.	26
Figure I.3. La structure de RNC.....	26
Figure I.4. Relation entre le nombre d'entraînement et le taux de précision	27
Figure I.5. Montage pratique proposé coté émetteur.....	29
Figure I.6. (a) Schéma explicatif de l'implémentation de RNA1, (b) Modèle de RNA1 proposé.....	30
Figure I.7. (a) Schéma explicatif de l'implémentation de RNA2, (b) Modèle de RNA2 proposé.....	31
Figure I.8. Diagramme de NAR RNA	33
Figure I.9. Diagramme d'input-output RNA	33
Figure I.10. Diagramme de NARX ANN	34
Figure I.11. Exemple d'un sur-apprentissage	36
Figure I.12. Signal ultrasonore (1 s)	37
Figure I.13. FFT obtenus à partir de signal ultrasonore.....	37
Figure I.14. Deux exemples de SSCEV résultant, avec des taux de compression différents. 38	
Figure I.15. Courbes de SSCEV des différents niveaux de pollution	39

Chapitre II

Figure II.1. Un neurone artificiel avec une fonction d'activation g	44
Figure II.2. Graphe de la fonction log-sigmoïde.....	45
Figure II.3. Graphe de la fonction tan-Sigmoïde	45
Figure II.4. Graphe de la fonction linéaire.....	46
Figure II.5. Séparation linéaire des fonctions AND, OR, XOR	46
Figure II.6. Architecture d'un réseau de neurone non-bouclé	47
Figure II.7. Différence entre les réseaux PMC et RBF	49
Figure II.8. Architecture d'un réseau RBF.....	50

Figure II.10. Activation du neurone RBF pour différentes valeurs de β	52
Figure II.11. Deux exemples de réseaux récurrents.....	53
Figure II.12. Le réseau Elman	54
Figure II.13. La descente de gradient	56
Figure II.14. Algorithme de gradient descent	57
Figure II.15. Apprentissage avec l'utilisation d'un terme du moment	61
Figure II.16. Réseau de neurones avec plusieurs couches convolutives	64
Figure II.17. Convolution de matrice d'image et matrice de filtre	65
Figure II.18. Feature Map	65
Figure II.19. Quelques filtres les plus utilisés.....	66
Figure II.20. Création de plusieurs feature maps construit la première couche de convolution.....	66
Figure II.21. Opération ReLu.	67
Figure II.22. Image d'entrée	67
Figure II.23. Image convoluée : noir (valeurs négatives) et blanc (valeurs positives)	67
Figure II.24. Valeur positive seulement	68
Figure II.25. Traitement de toutes les feature maps par la fonction ReLu	68
Figure II.26. Pooled feature map.....	69
Figure II.27. Pooling plusieurs carte construit la couche de pooling	69
Figure II.28. Redimensionnement de la matrice en vecteur.....	69
Figure II.28. Redimensionnement de toutes les cartes en vecteurs.....	70
Figure II.29. Architecture d'un RNC	70
Chapitre III	
Figure III.1. Dispositif expérimentale.....	73
Figure III.2. Modèle plan étudié : (a) vue latérale, (b) vue de haut.....	74
Figure III.3. Profil de l'isolateur réel (1512 L)	74
Figure III.4. Méthode de pulvérisation pour la pollution uniforme	75

Figure III.5. Signal du courant de fuite pour une conductivité de 10 $\mu\text{S}/\text{cm}$ et une tension appliquée de 12kVeff	76
Figure III.6. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivités 0.01mS/cm.	77
Figure III.7. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivités 0.19mS/cm	78
Figure III.8. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivités 0.71mS/cm	78
Figure III.9. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivités 1.2 mS/cm.....	78
Figure III.10. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour la conductivité 10.1 mS/cm	79
Figure III.11. Valeurs moyenne du courant de fuite en fonction de la conductivité.....	80
Figure III.12. Valeurs de STD en fonction de la conductivité.....	81
Figure III.13. Spectre d'une sinusoïde.....	82
Figure III.14. Signal du courant de fuite et sa représentation fréquentielle (FFT)	83
Figure III.15. Signal du courant de fuite et sa représentation fréquentielle (FFT) pour $\sigma = 190 \mu\text{S}/\text{cm}$	84
Figure III.16. Signal du courant de fuite et sa représentation fréquentielle (FFT) pour $\sigma = 710 \mu\text{S}/\text{cm}$	85
Figure III.17. Signal du courant de fuite et sa représentation fréquentielle (FFT)	86
Figure III.18. Signal du courant de fuite et sa représentation fréquentielle (FFT)	87
Figure III.19. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.01 mS/cm	88
Figure III.20. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.19 mS/cm	89
Figure III.21. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.71 mS/cm.....	89
Figure III.22. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 1.2 mS/cm.....	90

Figure III.23. Variation du déphasage courant-tension en fonction de la tension appliquée..91

Chapitre IV

Figure IV.1 courbe montre le problème de sur-apprentissage	94
Figure IV.2 Courbe d'évolution de la fonction MSE	97
Figure IV.3 les performances d'un réseau d'un réseau de neurones par la méthode de régression linéaire	98
Figure IV.4. Erreur d'entraînement des trois configurations 4, 1 et 7	101
Figure IV.5. Erreur d'entraînement des trois configurations 8 et 9	101
Figure IV.6. Erreur d'entraînement des quatre configurations 3, 6, 2 et 5	102
Figure IV.7. Erreur de validation des trois configuration 4, 1 et 7	103
Figure IV.8. Erreur de validation des trois configuration 6, 8 et 9	103
Figure IV.9. Erreur de validation des trois configuration 3, 2 et 5.....	104
Figure IV.10. Erreur de test des trois configuration 4, 1 et	105
Figure IV.11. Erreur de test des trois configuration 3, 6 et 9.....	105
Figure IV.12. Erreur de test des trois configuration 8, 2 et 5	106
Figure IV.13. L'erreur relative des trois configurations 2, 5 et 8	107
Figure IV.14. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle fréquentiel.....	109
Figure IV.15. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle fréquentiel	109
Figure IV.16. Erreur de test en fonction du nombre de neurone par couche cachée du modèle fréquentiel	110
Figure IV.17. Erreur relative en fonction du nombre de neurone par couche cachée du modèle fréquentiel.....	110
Figure IV.18. Erreur d'entraînement en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 a 6N-C1)	111
Figure IV.19. Erreur d'entraînement en fonction du nombre de neurone par couche cachée modèle fréquentiel (7N-C1 a 16N-C1)	112

Figure IV.20. Erreur de validation en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 a 9N-C1).....	112
Figure IV.21. Erreur de validation en fonction du nombre de neurone par couche cachée modèle fréquentiel (10N-C1 a 16N-C1).....	113
Figure IV.22. Erreur de test en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 a 8N-C1)	113
Figure IV.23. Erreur de test en fonction du nombre de neurone par couche cachée modèle fréquentiel (9N-C1 a 16N-C1)	114
Figure IV.24. Erreur relative en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 a 8N-C1)	114
Figure IV.25. Erreur relative en fonction du nombre de neurone par couche cachée modèle fréquentiel (9N-C1 a 16N-C1)	115
Figure IV.26. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel	116
Figure IV.27. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel	116
Figure IV.28. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel	117
Figure IV.29. Erreur relative en fonction du nombre de neurone par couche cachée du modèle temporel	117
Figure IV.31. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 a 9N-C1)	118
Figure IV.31. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel (10N-C1 a 16N-C1)	119
Figure IV.32. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 a 9N-C1)	119
Figure IV.33. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel (9N-C1 a 16N-C1)	120.
Figure IV.34. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 a 9N-C1).....	120

Figure IV.35. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel (9N-C1 a 16N-C1)	121
Figure IV.36. Erreur relative en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 a 8N-C1)	121
Figure IV.38 Structure du modèle fréquentiel.	123
Figure IV.39. Structure du modèle temporel	123
Figure IV.40. Evaluation de la fonction du cout du modèle fréquentiel en fonction du nombre d'itération	124
Figure IV.41. Performance du modèle fréquentiel par la méthode de régression linéaire...	125
Figure IV.42. Performance du modèle fréquentiel par la méthode de régression linéaire du sous-ensemble de test	126
Figure IV.43. Représentation de l'erreur relative du modèle fréquentiel.....	128
Figure IV.44. Évaluation de la fonction cout du modèle temporel en fonction du nombre d'itérations.	128
Figure IV.45. Performance du modèle temporel par la méthode de régression linéaire	129
Figure IV.46. Performance du modèle temporel par la méthode de régression linéaire du sous-ensemble de test	130
Figure IV.47. Représentation de l'erreur relative du modèle temporel	132
Figure IV.48. Erreurs relatives de modèle fréquentiel et temporel.....	132

INTRODUCTION GENERALE

Les équipements d'isolation, tels que les isolateurs présents dans les réseaux aériens de transport de l'énergie électrique, sont supposés opérer de façon fiable, quelles que soient les conditions environnementales. Leur rôle est d'assurer l'isolement électrique entre les parties portées à la haute tension entre elles (câbles) ou avec les parties mises à la terre (pylône).

Depuis plusieurs années, l'un des principaux axes de recherche et de développement dans le domaine de la transmission et de la distribution électrique est l'étude de la diminution du nombre et de la durée des interruptions de l'alimentation en énergie électrique. Une quantité importante d'isolateurs utilisés dans les réseaux aériens de transport d'énergie électrique qui peuvent souvent être amenés à opérer sous différents contextes environnementaux. Parfois, ces milieux sévèrement pollués, provoquent des dégradations, plus ou moins importantes, de la rigidité diélectrique des isolateurs. De ce fait, ces dernières détériorations causent dans souvent des cas, l'interruption complète du fonctionnement des lignes électriques [1].

Parmi les conditions météorologiques auxquelles sont soumis ces équipements d'isolation, ce sont les accumulations dues à la pollution hivernale (glace, pluie ou brouillard) qui sont les plus dangereuses [1-2]. En effet, ces accumulations réduisent considérablement la résistivité superficielle de l'isolateur affectent aussi considérablement leur performance diélectrique. Une fois leur rigidité diélectrique affaiblie, ces équipements d'isolation subissent des contournements électriques (communément appelé phénomène de claquage électrique) pouvant entraîner des interruptions partielles, voir totales, de l'énergie électrique [1-2].

Un contournement électrique se traduit par un court-circuit entre la partie portée à la haute tension et celle mise à la terre. Ce phénomène consiste en un arc électrique (ou arc partiel) qui est créé généralement à la surface de la couche de pollution. Puis, il tend à prendre plus d'espace, pour recouvrir la totalité de l'équipement d'isolation. En effet, l'apparition de ces arcs électriques partiels le long des intervalles d'air (c'est-à-dire les parties non recouvertes de pollution) est principalement causée par la modification de la distribution du champ électrique le long de l'équipement d'isolation recouvert d'un dépôt de pollution. Plus précisément, l'accroissement de la chute de tension le long des intervalles d'air, qui se traduit par une augmentation localisée de l'intensité du champ électrique, est à l'origine de ces arcs électriques. Cet accroissement de la chute de tension est amplifié par la présence d'une couche de pollution. Si l'intensité d'un champ électrique est suffisante, les arcs électriques se propagent le long de la surface de pollution, formant un arc blanc. Quand ce dernier atteint une certaine longueur, l'équipement d'isolation subit soudainement le contournement électrique [2].

Les coupures engendrées par les contournements électriques des isolateurs pollués sont souvent difficiles à identifier. De plus, certains de ces contournements se produisent dans des lieux très éloignés et inhabités. Par conséquent, ceci rend leur observation très aléatoire et difficile, constituant un risque pour les observateurs. Aussi, les interruptions engendrées par ces contournements se produisent dans la majorité des cas à des périodes où le besoin en énergie électrique est vital.

Au cours des dernières années, des méthodes d'intelligence artificielle telles que les réseaux de neurones artificiels (RNA) [3-5], la logique floue (FL) [4-6], la méthode des machines à vecteurs supports à moindres carrés (LS-SVM) [7-8] ainsi que d'autres méthodes ont été utilisées dans les cas d'isolateurs de haute tension pour l'estimation de la sévérité de pollution et la prédiction des conditions de contournement (tension appliquée et courant de fuite) [9]. Une telle démarche nous donne des informations qui peuvent être de forts utiles afin

de développer un système de surveillance en temps réel des postes dits stratégiques. De tels systèmes ont pour vocation de prédire l'imminence d'un contournement électrique au niveau des isolateurs. Ainsi, il permettrait aux équipes d'entretien d'avoir assez de temps pour agir en conséquence et donc d'assurer la fiabilité du réseau électrique.

Le but de ce travail consiste à proposer une technique de prédiction de la conductivité électrique de la couche polluante qui se dépose sur l'isolateur, en utilisant l'une des techniques de l'intelligence artificielle qui est les Réseaux de Neurones Artificiels (RNA). Cette démarche utilise le courant de fuite (LC) qui est un paramètre facilement mesurable et permettant de recueillir un certain nombre d'informations à la fois sur l'état de surface de l'isolateur ainsi que sur l'activité électrique se produisant à sa surface.

D'abord, le premier chapitre est consacré à synthétiser les travaux antérieurs. Il récapitule les études les plus récentes dans le domaine des isolateurs de haute tension, qui ont utilisées soit des méthodes mathématiques ou des méthodes qui se basent sur l'intelligence artificielle. Une courte description de chacune des méthodes de prédiction utilisées est fournie pour illustrer comment ces modèles de prédiction sont mis en œuvre.

Par la suite, le second chapitre est réservé à une description théorique des réseaux de neurones artificiels, leurs modes de fonctionnement, les différentes architectures RNA les plus adaptées, types de réseau, le choix et le traitement des entrées pertinentes, ainsi que nombre de couches cachées, nombre de neurones, types de fonction de transfert...etc. On présente aussi quelques processus d'apprentissage ainsi que la justification du recours au RNA.

Puis, le troisième chapitre est destiné pour l'extraction des paramètres du courant de fuite, qui sont les caractéristiques temporelles (I_{\max} , RMS, STD) qui sera le premier vecteur d'entrée du RNA, les caractéristiques fréquentielles (amplitude du 3^{ème} et 5^{ème} harmonique du LC et le déphasage entre la tension appliquée et LC) qui sera le deuxième vecteur. Nous avons expliqué en détail le choix de ces deux vecteurs par une analyse temporelle et fréquentielle du courant de fuite.

Durant le quatrième chapitre, nous avons détaillé et expliciter les modèles du RNA que nous avons appliqué à base des différents tests effectués. Ainsi, nous avons pu comparer minutieusement les performances de chaque modèle du RNA, dépendamment de leur paramètres (fonction d'apprentissage, nombres de couches cachées, nombres du neurones, etc...)

Enfin, notre manuscrit se solde par une conclusion générale, résumant nos principaux résultats, ainsi que nos commentaires et préconisations, se rapportant au choix du RNA optimal pour l'étude des isolateurs pollués.

CHAPITRE I
TRAVAUX
ANTERIEURS

Introduction

L'objectif de ce chapitre est d'effectuer une revue de littérature des principaux modèles des réseaux de neurones artificielles développés au cours de la dernière décennie utilisés dans le domaine de l'étude des isolateurs de Haute Tension. Ces recherches sont, pour la plupart, basées sur l'étude des signaux unidimensionnels à savoir le signal du courant de fuite ou de la tension appliquée, ainsi que les signaux bidimensionnels (image), ou bien le traitement et analyse du signal ultrasonore qui contiennent des informations précieuses et efficaces sur l'état de surface des isolateurs.

I.1 Évaluation du contournement des isolateurs en porcelaine avec décharge ultraviolet à l'aide d'un système d'apprentissage profond

S. Pei et al [10] ont mis en œuvre l'algorithme d'un réseau neuronal à convolution (RNC) pour l'évaluation du contournement dans les isolateurs en porcelaine. Durant leur étude, les images des spectres ultraviolets sont utilisées comme entrées du RNC, alors que l'isolateur de test en porcelaine a été soumis à des essais de contournement. Ainsi, les auteurs ont enregistré les spectres ultraviolets de différents états de décharge : sans décharge, faible décharge coronaire et une forte décharge d'étincelle ont été capturés par l'imageur UV FILIN.

I.1.1 Paramètre d'entrées d'un RNC

Actuellement, le traitement numérique de l'image est devenu la principale méthode utilisée pour la quantification du spectre UV. Les RNC sont à ce jour les modèles les plus performants pour le classement des images. A l'entrée du RNC, les images UV de l'isolateur sont bidimensionnelles, elles sont ensuite utilisées comme caractéristiques de convolution des isolateurs dégradés et extraites par propagation vers l'avant dans le RNC.

Pour obtenir les spectres UV des isolateurs en porcelaine dans différentes étapes de contournement, des essais ont été effectués, en outre, un isolateur en porcelaine double parapluie (Type- XWP2-70) a été sélectionné. Le dispositif expérimental est présenté à la figure 2.

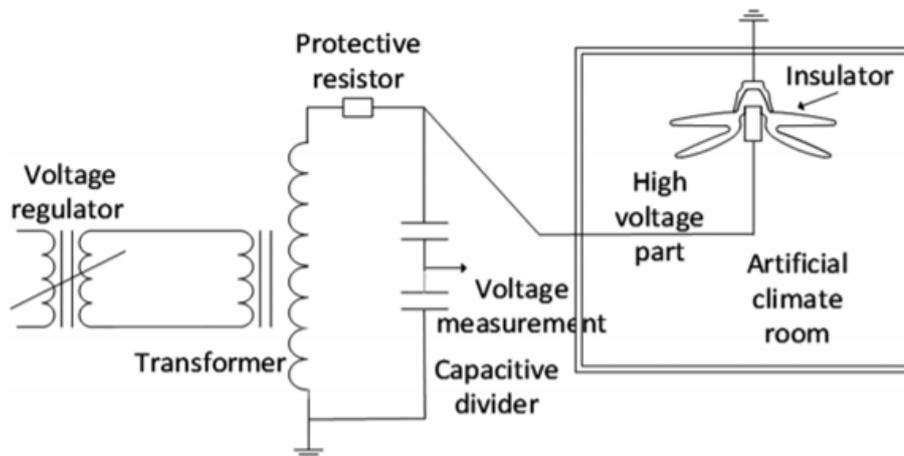


Figure I.1. Schéma de câblage de la plateforme expérimentale

Au total, 5326 images de spectres UV ont été obtenues de toutes les expériences UV dans un test de contournement d'isolateur en porcelaine : elles ont été divisées, au hasard, en 4793 cadres formant l'ensemble d'entraînement, et 533 cadres formant l'ensemble d'essai.

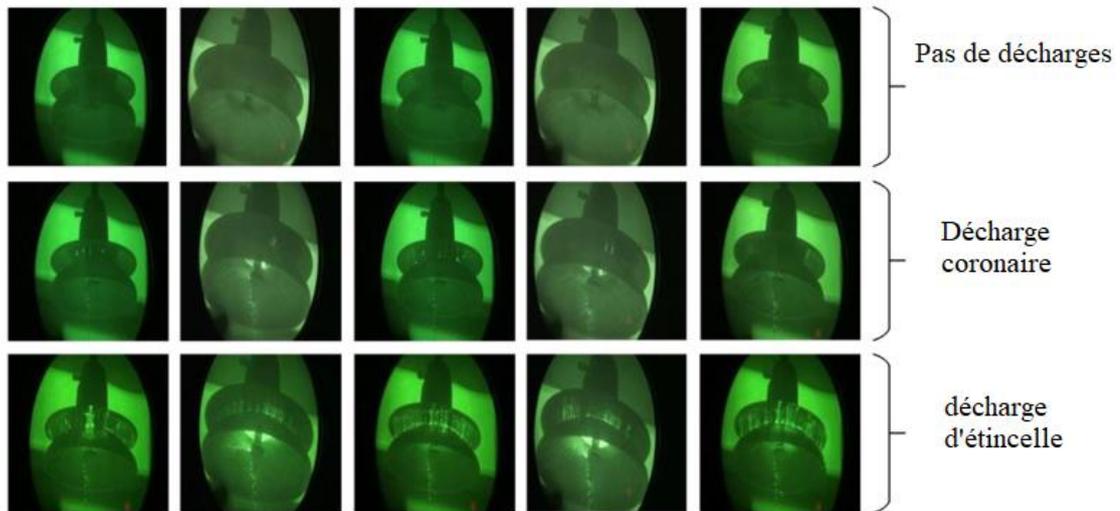


Figure I.2. Échantillon typique d'entraînement

I.1.2 Structure d'un RNC

Les RNC, pour leur structure de réseau neuronal multicouche, sont généralement constitués d'une couche de convolution, d'une couche de pool, d'une couche de connexion complète, d'une fonction d'activation. La structure de RNC est présentée dans figure I.3 [11,12].

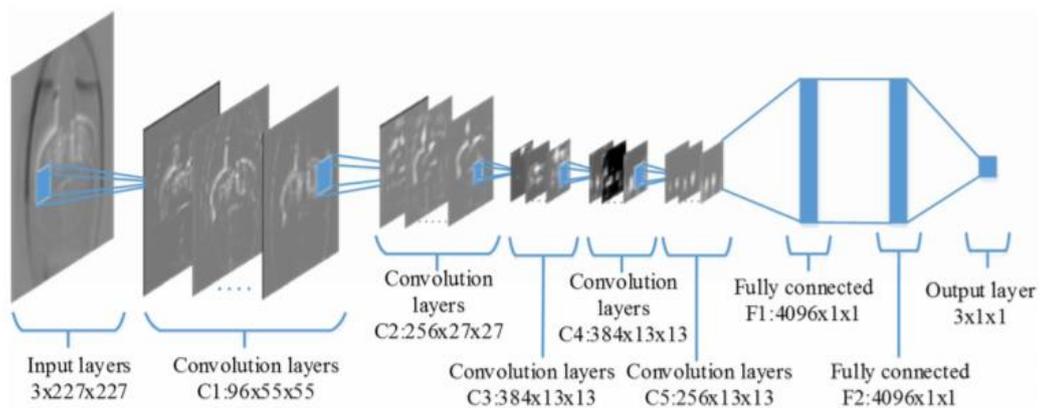


Figure I.3. Structure de RNC

La première partie d'un RNC est la partie convolutive. Elle fonctionne comme un extracteur de caractéristiques des images. Une image est passée à travers une succession de filtres, ou noyaux de convolution, créant de nouvelles images appelées cartes de convolutions. Certains filtres intermédiaires réduisent la résolution de l'image par une opération de maximum local. Au final, les cartes de convolutions sont mises à plat et concaténées en un vecteur de caractéristiques, appelé code RNC [13].

Ce code RNC en sortie de la partie convolutive est ensuite envoyé en entrée d'une deuxième partie, constituée de couches entièrement connectées (fully connected) qui représente un perceptron multicouche. Le rôle de cette partie est de combiner les caractéristiques du code RNC pour classer l'image.

I.1.3 Résultats et discussion

Pour obtenir un entraînement optimal, l'objectif de l'optimisation du modèle d'entraînement est atteint en fixant le paramètre du taux d'entraînement de base. En effet, Le taux d'apprentissage est un paramètre qui contrôle à quel point ajustant les poids de notre réseau, Plus la valeur est basse, plus que le temp de convergence vers la cible est grand. Au cours du processus d'entraînement, les taux d'entraînement de base utilisés étaient de 0,01, 0,001, 0,005, 0,0001 et 0,0005.

La figure I.4 montre, pour différents taux d'entraînement, l'augmentation du nombre d'entraînement entraîne l'augmentation de la précision. Intuitivement, on peut voir que lorsque le taux d'entraînement est de 0,001, le modèle d'entraînement peut atteindre une convergence rapide et atteindre une précision élevée de 96,8 %.

À mesure que le taux d'entraînement augmente, comme 0,005 et 0,01, le modèle ne peut pas atteindre une valeur optimale en raison du taux d'entraînement élevé : la précision des deux taux d'entraînement est finalement restée à 53,6 %. Comme le taux d'entraînement est réduit à 0,0005 et à 0,0001, les taux de précision finale étaient de 93,2 et 87,6 % ; comme le taux d'entraînement était faible, l'étape d'entraînement était courte, le modèle était facilement limité à une valeur optimale locale, et l'optimum global n'a pas été atteint.

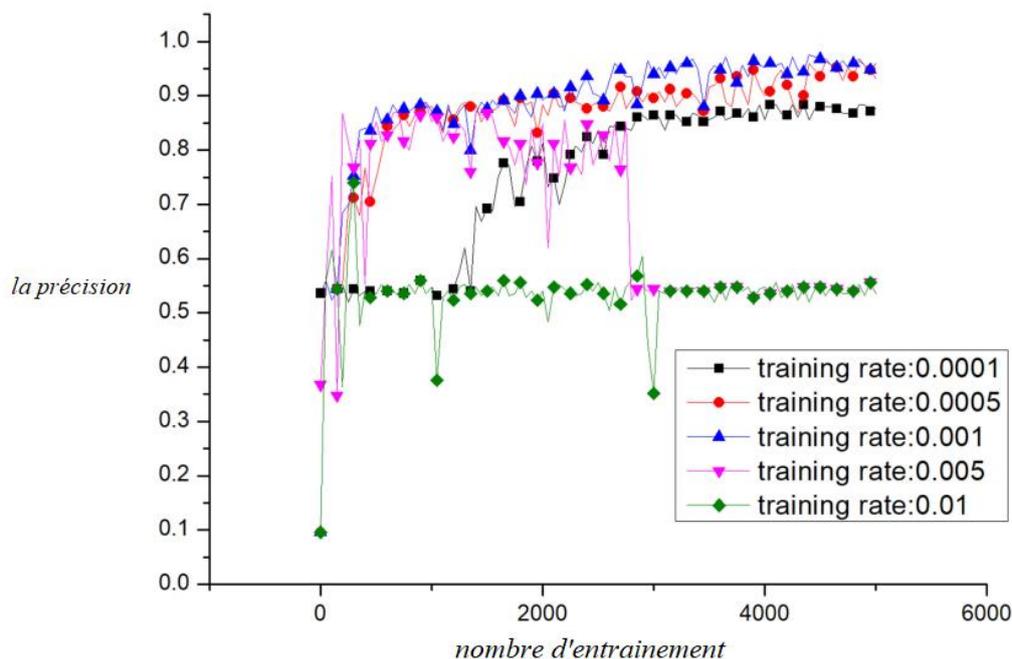


Figure I.4. Relation entre le nombre d'entraînement et le taux de précision

Pour la première fois en 2012, les RNC atteignent des taux d'erreur de classification d'image aussi bas que 15,4 %, puis un taux d'erreur sous-optimal de 26,2 %. La performance

rend le RNC exact dans les deux-dimensionnels enjeux de classification de l'information d'image [14].

Comparativement à la théorie traditionnelle des méthodes d'apprentissage machine, le RNC transforme l'ensemble du lien entre les neurones traditionnels en un partage de poids entre les couches et les neurones, ce qui réduit considérablement l'entraînement requise pour l'ensemble des paramètres de poids CNN [15], permettant ainsi un entraînement et des tests de modèle plus rapides. Les résultats montrent que les CNN sont très efficaces pour la vision par ordinateur et la reconnaissance des images [16].

Dans ce travail les auteurs ont concentré sur l'application d'un CNN dans la classification et évaluation du motif de décharge Ultra-Violet d'un isolant, basé sur le motif de décharge Ultra-Violet unique d'un isolateur en porcelaine, la bibliothèque de divers types d'état de décharge prise par l'imageur UV a été établie par une série de tests de contournement de la fréquence d'alimentation des isolateurs en porcelaine. La méthode de convolution profonde par réseau neuronal a été utilisée pour étudier la porosité de la classification intelligente et l'évaluation de l'imagerie ultraviolette à haute précision des isolateurs. Les résultats montrent que le CNN offre les avantages d'une grande précision, d'une convergence rapide et d'une forte capacité de généralisation pour l'entraînement et la classification des données saisie en deux dimensions. Il peut répondre aux exigences de la détection d'ingénierie pratique, et est donc applicable.

I.2 Diagnostic en temps réel des isolateurs de lignes aériennes du réseau de distribution de moyenne tension

Osama E. Gouda et al [17] ont traité la prédiction du degré de pollution des isolateurs des lignes aériennes du réseau de distribution de la moyenne tension dans le désert d'égypt. Une nouvelle méthode a été proposée qui est basée essentiellement sur deux structures de réseaux de neurone, ils sont utilisés pour classifier l'état de surface d'un isolateur afin d'éviter les claquages et prendre des actions correctives sur les isolateurs (nettoyage, changement, réparation, etc...).

I.2.1 Montage pratique proposé

Le montage diagnostic en temps réel des isolateurs de lignes aériennes proposé par les auteurs est représenté sur la figure I.5

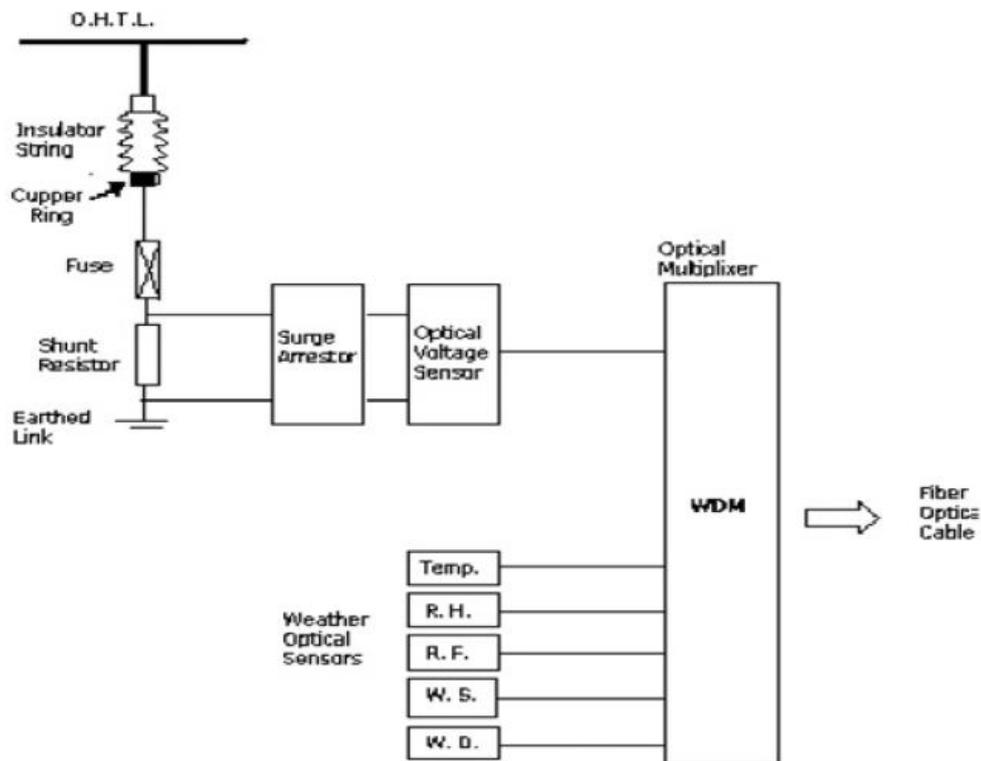


Figure I.5. Montage pratique proposé coté émetteur

Le montage proposé est composé de deux équipements essentiels qui aident à transporter les valeurs instantanées des paramètres qu'on va citer dans ce que suit.

Il faut mesurer les signaux de courant de fuite provenant de la chaîne d'isolateurs, et aussi les conditions atmosphériques périodiquement. Pour qu'on puisse les analyser, interpréter et prendre la décision sur l'état de la surface de l'isolateur surveillé par le montage de figure I.5.

Les conditions atmosphériques utilisées dans le modèle proposé sont :

- Température
- Précipitations de pluie (R.F)
- Humidité relative (R.H)
- Vitesse du vent (W.S)
- Direction de vent (W.D)

I.2.2 Méthodes de prédiction proposées

Les auteurs ont construit deux modèles de réseaux de neurones artificiels travaillant en parallèle. Comme indiqué sur la figure I.6, les signaux de courant de fuite après avoir normalisé, ils rentrent dans le premier réseau de neurone artificiel pour les classer selon sa catégorie de forme, le signal de courant de fuite est divisé en 21 entrées (c'est-à-dire chaque cycle échantillonné sur 21 échantillons) qui vont être les entrées de RNA1, et d'après les essais expérimentaux les auteurs ont décidé que la sortie de ce modèle peut identifier 3 catégories de forme de courant de fuite :

1. Sinusoïdal (petit courant sans décharges)
2. Fortement non linéaires (formation de bande sèche ou intermédiaire décharges)
3. Avec des impulsions (décharges sévères)

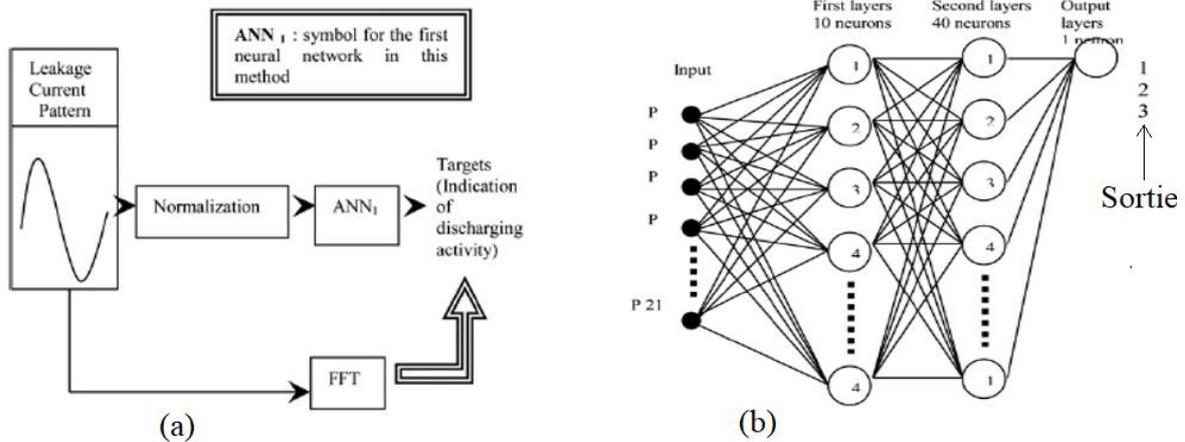


Figure I.6. (a) Schéma explicatif de l'implémentation de RNA1, (b) Modèle de RNA1 proposé

Chaque signal contient des harmoniques spécifiques, donc le processus de classification de (RNA1) atteint ses objectifs à partir d'une analyse spectrale réalisée à l'aide de la transformée de Fourier rapide, le RNA1 est représenté dans la Figure I.6.

Pour le deuxième réseau de neurones artificiels (RNA2) représenté sur la figure I.7, les auteurs considèrent les valeurs moyennes de 5 dernières heures successives (c'est-à-dire on prend la valeur moyenne de chaque heure de chaque paramètre) de 5 paramètres atmosphérique qu'on a choisis avant comme entrées de ce modèle, donc le nombre d'entrée du modèle est 25, en entraînant ce modèle avec ces entrées on obtient en sortie une classification des conditions atmosphériques :

- Sorite=1 ces conditions sont responsables du contournement de l'isolant dans la région.
- Sorite=0 ces conditions ne sont responsables du contournement de l'isolant dans la région.

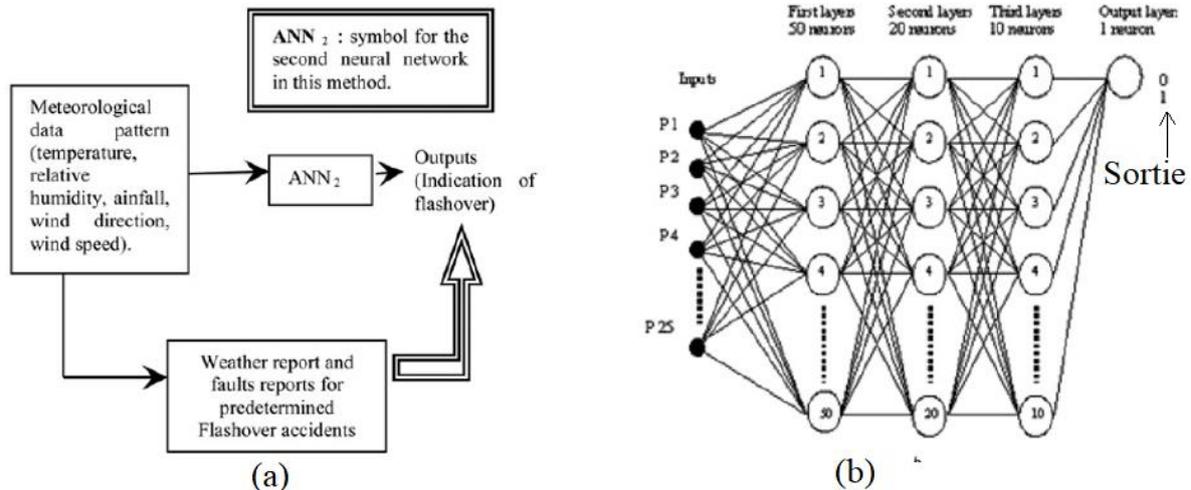


Figure I.7. (a) Schéma explicatif de l'implémentation de RNA2, (b) Modèle de RNA2 proposé

Une alarme indiquant l'approche d'événement de contournement est donnée lorsque les deux réseaux sont activés, surtout lorsque (RNA1) indique la deuxième ou la troisième cible d'activation. Sinon, aucun danger n'est conclu puisque les décharges ne sont pas notifiées ou les paramètres atmosphériques ne motivent pas à la continuité de la décharge.

I.2.3 Structure de réseau de neurones artificiels

Il existe plusieurs méthodes d'apprentissage des RNA, la méthode feed-forward back-propagation est utilisée pour l'apprentissage de deux réseaux de neurones proposés dans Figure I.6 et Figure I.7, cette méthode nous a permis d'obtenir les paramètres optimaux de ces deux modèles avec un erreur carrée égale à 0,246. Aussi le nombre de couches cachées, et le nombres de neurones de chaque couche est indiqué sur les figures I.6 (b) et I.7 (b).

I.2.4 Test de montage proposé

Après les résultats obtenus à partir des simulations dans des conditions différentes les auteurs ont décidé la classification suivante [18] :

1. Signal sinusoïdal petit non linéaire déformé dans lequel le pourcentage des harmoniques impairs contenu dans ce signal est <20%.
2. Signal déformé très non linéaire dans lequel le pourcentage des harmoniques impairs contenu dans ce signal est compris entre 20% et 30%.
3. Décharges sévères avec courant de fuite transitoire dans lequel le pourcentage des harmoniques impairs contenu dans ce signal est >30%. Et le pourcentage des harmoniques impair est > 5%.
4. Décharges sévères avec courant de fuite non transitoire dans lequel le pourcentage des harmoniques impairs contenu dans ce signal est >30%. Et le pourcentage des harmoniques impair est <5%.

Tableau I.1. Les cas possibles de sortie de montage proposé

Sortie ANN1	Sortie ANN2	Signification	Alarme
1	0	Sinusoïdal /petit temps normal non linéaire	Non
2	0	Temps normal hautement non linéaire	Non
3	0	Rejets sévères temps normal	Non
1	1	Sinusoïdal /petit temps mauvais non linéaire	Standby
2	1	Temps mauvais hautement non linéaire	Oui

Par conséquent, en connaissant les harmoniques contenus dans le signal de courant de fuite et sa forme on peut savoir l'état de la surface d'un isolateur, en plus les informations sur les conditions atmosphériques vont être interpréter par le RNA2, et il va indiquer si ces conditions vont causer le contournement de l'isolateur ou non.

Les auteurs ont présenté un nouveau modèle pour vérifier l'état des isolateurs pollués des lignes de transport de moyenne tension et pour prévoir les événements de contournement plus tôt avant qu'un défaut ne se produise et ne soit détecté par des capteurs protégés. Ce modèle fonctionne en temps réel et il donne des résultats sur les modèles de courant de fuite et les enregistrements atmosphériques. Les réseaux de neurones artificiels utilisés dans ce modèle sont construits, formés et testés pour classer les modèles de courants de fuite et les conditions atmosphériques suivant les cas proposés avant. Le montage affiche une erreur quadratique moyenne de 0,246 au cours des tests.

I.3 Régularisation bayésienne du réseau de neurones pour prédire le comportement du courant de fuite dans un environnement de brouillard salin

AI Khafaf et al [19] ont utilisé le réseau de neurones bayésien régularisé pour prédire les composantes fondamentale et troisième harmonique du courant de fuite dans des conditions de brouillard salin. Trois différents modèles de réseau de neurone sont proposés et chacun est Conçu pour prédire la série temporelle du fondamental et du troisième harmonique du courant de fuite.

I.3.1 Réseaux de neurones artificiels utilisés

Trois différents modèles RNA ont été utilisés pour prédire la valeur suivante des composantes fondamentale et troisième harmonique du LC.

I.3.1.1 Modèle RNA non linéaire autorégressif (NAR)

Le premier modèle de réseau implémenté c'est le réseau de neurones non linéaires autorégressifs (NAR), qui prédit la valeur suivante d'une série temporelle à partir de la valeur précédente de cette série temporelle.

Le réseau de neurone autorégressif non linéaire est approprié pour l'estimation des valeurs suivantes de la variable d'entrée. Il permet de prédire les valeurs suivantes d'une série chronologique (temporelle), Grâce à l'utilisation d'une boucle servant à exploiter les valeurs prédites du LC comme entrée du RNA [20, 21]. Autrement dit, le réseau est créé et formé en boucle ouverte, en utilisant les valeurs cibles actuelles comme retour d'information, assurant

ainsi une plus grande précision de l'Apprentissage. Après l'apprentissage, le réseau est converti en boucle fermée et les valeurs prédites sont utilisées pour fournir de nouvelles entrées de retour au réseau.

Cette boucle s'exprime analytiquement ainsi :

$$Y(t) = g(Y(t-1), \dots, Y(t-d)) \quad (I.1)$$

$Y(t)$ est la valeur prédite de la série temporelle, $Y(t-1), \dots, Y(t-d)$ sont les valeurs précédentes de la série temporelle et d est le nombre de retards dont dépend la prédiction suivante. Le modèle de réseau est illustré à la figure I.8.

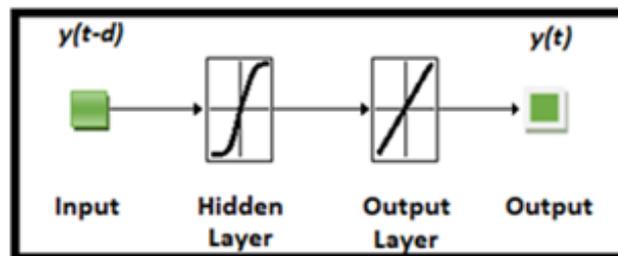


Figure I.8. Diagramme de NAR RNA

I.3.1.2 RNA basé sur le modèle input-output (I-O)

Le deuxième modèle RNA utilisé est le modèle entrées-sorties (I-O), dans lequel la valeur suivante d'une série temporelle est prédite uniquement à partir de la valeur précédente d'une autre série temporelle corrélée.

Il est exprimé comme suit :

$$Y(t) = g(X(t-1), \dots, X(t-d)) \quad (I.2)$$

$Y(t)$ est la série temporelle du fondamental (ou troisième harmonique) de LC, et $X(t-1), \dots, X(t-d)$ sont les valeurs précédentes effectives de la série temporelle du troisième harmonique (ou fondamental) de LC et d est le délai qui représente combien de points d'échantillons antérieurs ont été utilisés dans la prédiction. Le modèle I-O est illustré à la figure I.9.

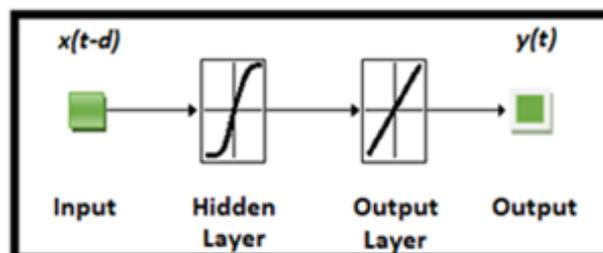


Figure I.9. Diagramme d'input-output RNA

I.3.1.3 Modèle RNA à entrées exogènes autorégressif (NARX)

Le troisième modèle du RNA utilisé est le réseau de neurone à entrées exogènes (NARX) autorégressif non linéaire où la valeur suivante d'une série temporelle est prédite à partir des valeurs précédentes de la même série temporelle ainsi que des valeurs précédentes d'une autre série temporelle corrélée pouvant être convertie. Le modèle exogène autorégressif (NARX) est un modèle autorégressif non linéaire qui comporte des entrées exogènes. Cela signifie que le modèle relie la valeur actuelle d'une série temporelle à la fois aux :

- Valeurs précédentes de la même série
- Valeurs actuelles et précédentes de la série exogène, c'est-à-dire de la série déterminée de manière externe qui influence la série d'intérêt.

Ce modèle est exprimé comme :

$$Y(t) = g(Y(t-1), \dots, Y(t-d), X(t-1), \dots, X(t-d)) \quad (I.3)$$

Où $Y(t)$ est la série temporelle fondamentale prédite de LC et $Y(t-1) \dots, Y(t-d)$ et $X(t-1) \dots, X(t-d)$ sont les valeurs actuelles et précédentes des composantes fondamentale et troisième harmonique de la série temporelle de LC, respectivement. La figure 12 présente le schéma fonctionnel de NARX ANN.

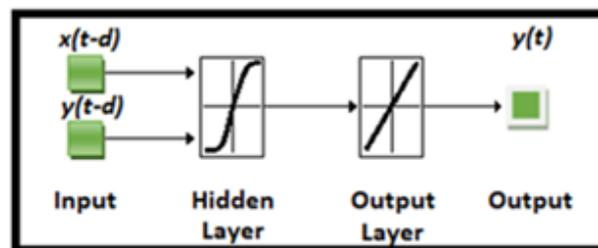


Figure I.10. Diagramme de NARX ANN

Les trois modèles RNA proposés comprennent un réseau à deux couches avec une fonction de transfert sigmoïde dans la couche cachée et une fonction linéaire dans la couche en sortie. Les réseaux nécessitent un entraînement avant de pouvoir être utilisés pour prédire les valeurs suivantes de la série chronologique (temporelle).

Les données d'Apprentissage pour prédire le fondamental et le troisième harmonique sont : la tension en (kV), la conductivité en (S/cm), la longueur en (cm) et la densité équivalente de dépôts de sel (ESSD) en (mg/cm).

Le tableau I.2 présente des exemples de données utilisées pour l'apprentissage du RNA ainsi que des conditions expérimentales.

Tableau I.2. Exemple de données d'apprentissage

Rod Number	Voltage (kV)	Conductivity (S/cm)	Length (cm)	ESDD (mg/cm ²)
1	0.6	15	6	0.0466
2	0.6	15	6	0.0322
4	0.6	15	6	0.0268
8	1.2	10	6	0.0792
13	1.2	15	6	0.15036
19	1.8	15	6	0.2639
20	1.8	15	6	0.225
26	1.8	15	6	0.307
42	1.8	15	6	0.147
48	2	20	10	0.03335
55	4	15	10	0.016
62	5	20	10	0.258
64	5	15	10	0.113
79	5	20	10	0.203

Tous les modèles du réseau de neurone permettant de prédire le fondamentale et le troisième harmonique sont conçus à l'aide de l'algorithme de régularisation bayésien, de deux couches cachées et de 5 à 7 délais d pour différents modèles de réseau.

Le RNA conçu prend 5 à 7 valeurs précédentes de LC (en fonction du modèle RNA utilisé) en tant qu'intrant permettant de prédire la prochaine valeur future du LC. Le processus est répété jusqu'à ce que la prédiction soit arrêtée (précise). La valeur prédite dépend principalement des observations précédentes ainsi que des tendances précédemment observées à partir de l'ensemble de données d'apprentissage. Une fois la prédiction arrêtée, les valeurs prédites sont comparées aux valeurs actuelles mesurées pour évaluer l'erreur de la prédiction et la performance du le RNA.

L'un des problèmes de l'apprentissage d'un RNA en utilisant une approche standard est le surapprentissage (Figure I.11), dans lequel la fonction d'erreur de performance du réseau est faible pendant la phase d'apprentissage, c'est à dire le réseau ne prévoit pas de manière fiable les observations suivantes, en outre la performance augmente considérablement lorsque le réseau de neurone reçoit de nouvelles données.

La figure I.11 nous montre un surapprentissage dans un apprentissage supervisé. En rouge, l'erreur sur l'ensemble de validation. En bleu, l'erreur d'apprentissage. Si l'erreur de validation augmente alors que l'erreur d'apprentissage continue à diminuer alors il y a un risque de surapprentissage.

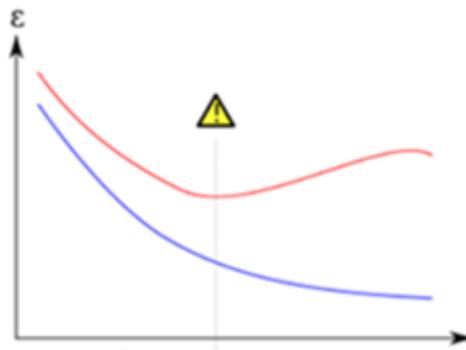


Figure I.11. Exemple d'un surapprentissage.

En effet, la régularisation bayésienne revient à ajouter des informations durant l'apprentissage en pénalisant les valeurs extrêmes des paramètres, car ces valeurs correspondent souvent à un surapprentissage [22].

I.3.2 Comparaison entre les différents modèles de RNA

Chaque fois que l'I-O RNA se comportait bien, une performance similaire a été constatée sur le NARX RNA. Toutefois, le NARX a obtenu de meilleurs résultats que le I-O RNA, car il dépend à la fois des composants fondamentaux et du troisième harmonique de LC, tandis que le I-O RNA utilisera l'un des composants pour prédire l'autre.

En moyenne, le NAR RNA a enregistré la meilleure erreur de prédiction sur presque tous les échantillons de bâtonnets, ce qui indique que c'est un modèle efficace pour prédire les valeurs suivantes de la série temporelle des composantes de LC.

Lorsqu'il existe une forte corrélation entre le fondamentale et le troisième harmonique de LC, les performances de NARX RNA sont meilleures que celles de NAR RNA et inversement.

La régularisation bayésienne du RNA prédit la série temporelle du fondamentale et du troisième harmonique de LC à partir des valeurs antérieures de cette série, ce qui rend le système proposé supérieur aux autres systèmes de prédiction en termes d'efficacité de prédiction du LC. De plus, le système prédit la valeur instantanée du LC, ce qui laisse plus de temps à l'ingénieur des services publics pour planifier une action.

Dans l'ensemble, les trois différents modèles de RNA à régularisation bayésienne proposés peuvent être appliqués dans le cadre d'un modèle d'étude des conditions pour indiquer chaque fois qu'une action est requise pour empêcher l'accumulation de contaminants à la surface de l'isolateur pour éviter les contournements.

I.4 Nouvelle méthode d'estimation de degré de pollution d'un isolateur basée sur l'ultrason et réseaux de neurones artificiels

Tarso Ferreira et al [23] ont proposé une méthode récente pour évaluer le degré de contamination d'un isolateur pollué, parmi les différentes méthodes existantes traitant ce phénomène, elle est basée sur l'utilisation des émissions acoustiques (AE) générées par des isolateurs en céramique, étant donné que, l'isolateur émet des ondes acoustiques, durant les conditions de fonctionnement réelles, se caractérisant par la quantité d'énergie portée par chaque harmonique. Cette méthode de détection acoustique est préconisée son en raison de

leurs caractéristiques, telles que le faible Coût et ainsi que l'immunité au bruit électromagnétique (EM) [24].

Les auteurs ont utilisé des capteurs pour enregistrer les ondes ultrasons, après ils ont utilisé transformé de fourrier (FFT) pour passer du domaine temporelle vers le domaine fréquentiel le résultat est indiqué dans la figure I.12, les harmoniques contenus dans le signal ultrason sont consistante, par conséquent il est difficile de construire un modèle de réseaux de neurones artificiels RNA avec un tel nombre d'entrées, il va réduire l'efficacité de cette méthode , c'est pour cette raison les auteurs , ils ont fait appel à l'algorithme « subband spectral centroid energy vectors » (SSCEV) proposé par Paliwal en 1998 [25] .

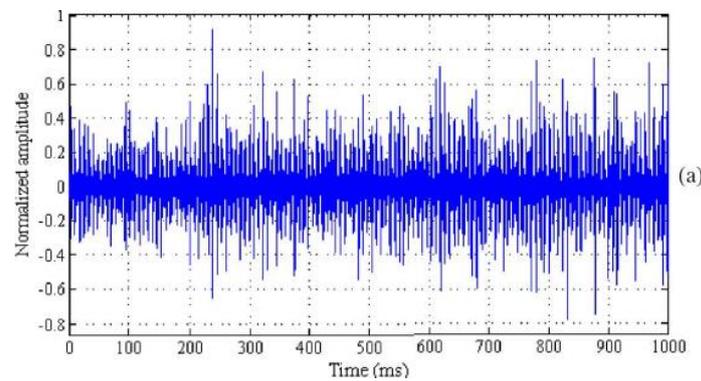


Figure I.12. Signal ultrasonore (1 s)

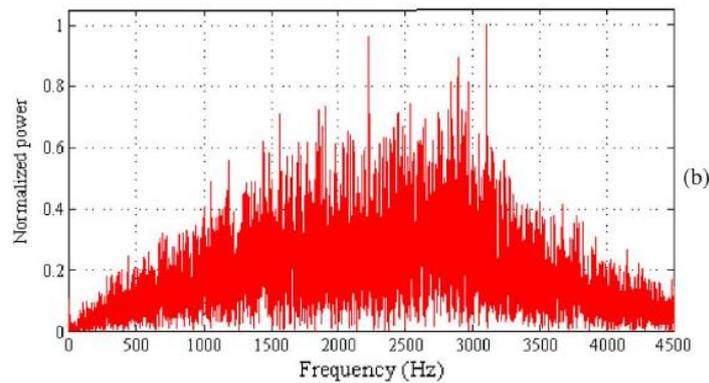


Figure I.13. FFT obtenu à partir de signal ultrasonore

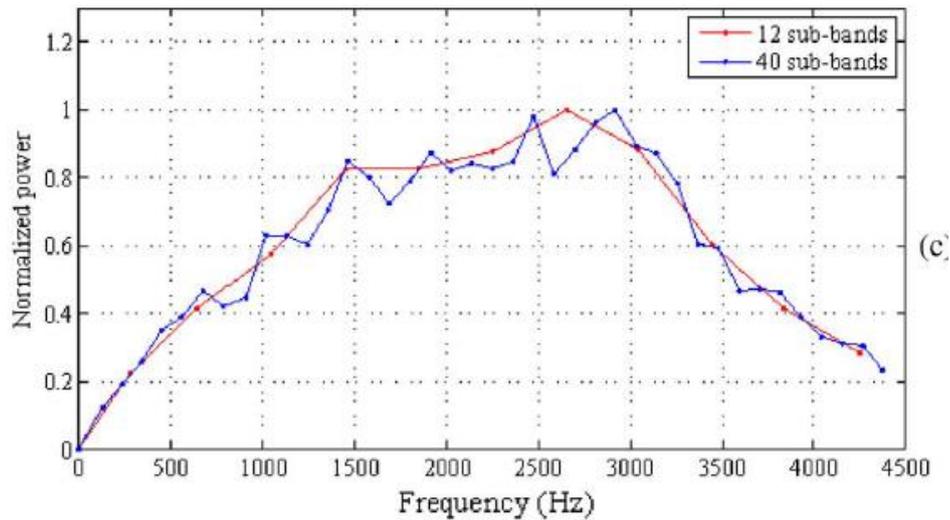


Figure I.14. Deux exemples de SSCEV résultant, avec des taux de compression différents.

I.4.1 Algorithme SSCEV

Etant donné que les ondes sonores sont portées par des harmoniques, cet algorithme propose de diviser les harmoniques à des intervalles qui ont une centre et un largeur bien définie, ensuite la somme des énergies est calculée pour les harmoniques contenus dans chaque intervalle et on l'associe au centre de cet intervalle [26], donc à l'aide de SSCEV on obtient une représentation simple, facile à interpréter, comme montré sur la figure I.12, les auteurs ont utilisé deux différentes largeurs de l'intervalle. Il faut bien choisir la largeur pour avoir des détails suffisants afin de pouvoir différencier les différents degré de pollution, par contre la représentation après l'application de FFT est difficile à interpréter comme il est montré dans la figure I.13.

Les étapes pour calculer les SSCEV :

1. Calculer le FFT de chaque signal d'USAE enregistré.
2. Ensuite, on subdivise le Spectrum obtenu en plusieurs intervalles en utilisant l'application de 'rectangular superposed bandpass filters'.
3. Après, On calcule le centroïde $C_H(\text{Hz})$ pour chaque sous-bande est calculée en utilisant

$$C_H = \frac{\sum_{k=0}^{N-1} H_m(k) \times P(k)}{\sum_{k=0}^{N-1} P(k)} \quad (I.4)$$

Où $P(k)$ est le spectre estimé, $H_m(k)$ le vecteur des fréquences et N est le nombre d'échantillons limités par le filtre.

4. L'énergie associée à chaque centroïde est calculée par

$$E_c = \sum_{k=C_H-\delta \cdot N}^{C_H+\delta \cdot N} P(k) \quad (I.5)$$

Où δ est un paramètre qui définit la largeur de la plage au voisinage Ch .
L'énergie de l'élément SSCEV est ensuite calculée dans la plage définie par δ .

En suivant les étapes pour calculer les SSCEV, on obtient un nombre réduit des entrées de réseaux de neurones artificiels, en plus cette réduction n'influe pas sur l'évaluation de degré de pollution des isolateurs.

I.4.2 Modèle de réseau de neurones artificiel proposé

Les auteurs ont considéré les énergies des groupes des harmoniques obtenus par l'algorithme SSCEV comme entrées du réseau de neurones artificielles, ils ont choisi un modèle de réseaux de neurones artificiels de type perceptron à trois couches [27] (entrées, une couche cachée, une sortie, entraîné par l'algorithme de propagation résiliente [28]).

Ils ont utilisé trois approches différentes pour entraîner, tester et évaluer le modèle de RNA, et ils ont pris comme sortie Le niveau de contamination d'un isolateur représentant par le dépôt de sel équivalent (ESDD) avec les classes assignées par les auteurs, comme représenté sur le tableau I.3.

Tableau I.3. Classement des isolateurs pollués en fonction d'ESDD

Classe	Classification	ESDD (mg/cm ²)	Conductivité (mS/cm)
1	Atmosphère propre	0-0,032	10
2	Contamination faible	0,032-0,038	14
3	Contamination légère	0,038-0,056	20
4	Contamination importante	0,056-0,123	40
5	Contamination très importante	0,123	80

En pratique chaque niveau de pollution indiqué dans le tableau I.3 est caractérisé par l'énergie portée par les harmoniques composés du signal ultrasonore émit, on peut voir clairement le différent degré de pollution sur les figures I.15.

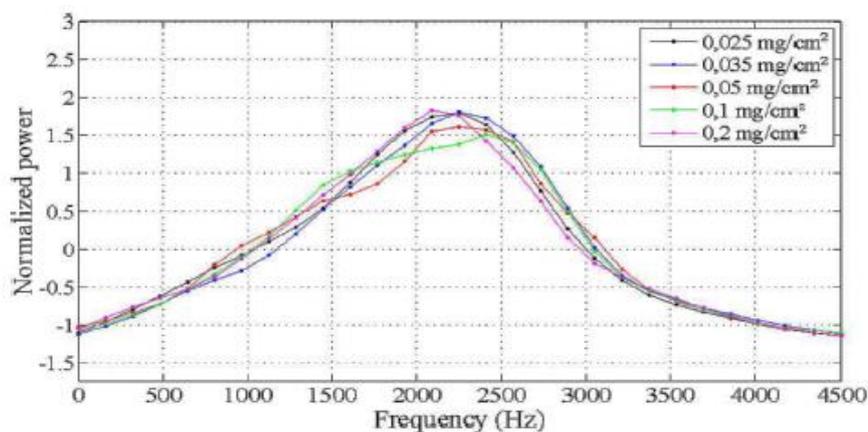


Figure I.15. Courbes de SSCEV des différents niveaux de pollution

Cette méthode récente d'évaluation de degré de pollution des isolateurs en céramique, est basée essentiellement sur l'analyse des émissions acoustiques. Pour augmenter la performance de cette méthode, l'algorithme de SSCEV a été utilisé. En plus, le modèle de réseaux de neurones artificiels (RNA) était un instrument de soutien de la prise de décision, comme il est montré dans le Tableau I.4.

Tableau I.4. Taux de succès de modèle de RNA

RH (%)	Succès (%)
60	96,38
65	96,88
70	98,33
75	98,88
80	99,96
85	98,88
90	99,17

Les résultats obtenus ont montré que le RNA est capable de reconnaître des modèles émissions acoustiques associés à des niveaux de pollution déterminés.

En d'autres termes, la technique est fiable tant que la base de données disponible prend en compte toutes les conditions d'humidité et de pollution.

La méthode proposée a donné des résultats satisfaisants, mais des études supplémentaires sont encore nécessaires pour déterminer les niveaux d'humidité relative RH qui sont plus représentatifs de la formation du RNA.

Tableau I.5. Influence de l'humidité sur le classement de modèle de RNA entraîné avec une seule valeur de RH

Testing		Training (RH = 75%)																		
RH (%)	ESDD (mg/cm ²)	0.025	0.035	0.05	0.1	0.2	0.025	0.035	0.05	0.1	0.2	0.025	0.035	0.05	0.1	0.2				
60	0.025	0.00	92.58	7.42	0.00	0.00	75	0.025	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00				
	0.035	0.00	59.03	0.00	40.97	0.00		0.035	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00				
	0.05	0.00	40.65	12.90	46.45	0.00		0.05	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00				
	0.1	0.00	36.77	3.23	60.00	0.00		0.1	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00				
	0.2	0.00	95.48	3.87	0.65	0.00		0.2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00				
65	0.025	0.00	64.19	1.94	33.87	0.00	80	0.025	38.71	38.71	18.06	0.00	4.52	85	0.025	98.39	1.61	0.00	0.00	0.00
	0.035	0.00	92.58	0.00	7.42	0.00		0.035	63.55	0.65	35.81	0.00	0.00		0.035	100.00	0.00	0.00		
	0.05	0.00	33.87	13.55	52.58	0.00		0.05	30.65	0.00	41.29	28.06	0.00		0.05	100.00	0.00	0.00		
	0.1	0.00	56.77	16.77	26.45	0.00		0.1	99.03	0.00	0.00	0.97	0.00		0.1	72.58	0.00	27.42	0.00	
	0.2	0.00	97.42	0.00	2.58	0.00		0.2	0.00	3.55	12.58	0.00	83.87		0.2	0.00	4.84	2.90	0.00	92.26
70	0.025	65.16	9.35	3.23	22.26	0.00	90	0.025	47.42	22.26	30.32	0.00	0.00	95	0.025	98.39	1.61	0.00	0.00	0.00
	0.035	0.00	91.61	0.00	8.39	0.00		0.035	16.13	64.19	0.00	19.68	0.00		0.035	100.00	0.00	0.00		
	0.05	0.00	16.77	0.32	82.90	0.00		0.05	82.90	0.00	0.00	17.10	0.00		0.05	100.00	0.00	0.00		
	0.1	0.00	56.13	37.10	6.77	0.00		0.1	100.00	0.00	0.00	0.00	0.00		0.1	100.00	0.00	0.00		
	0.2	0.00	95.16	4.84	0.00	0.00		0.2	0.00	0.00	23.87	0.00	76.13		0.2	0.00	0.00	23.87	0.00	

L'analyse des résultats du Tableau I.4. Montre que la performance de le RNA est insuffisante car la plupart des modèles ont été affectés à une mauvaise classe de pollution. Cela montre l'influence de RH dans l'USEA.

Les auteurs ont constaté l'existence de caractéristiques spectrales particulières dans les niveaux de pollution qui pourraient être préservés même avec la variation d'humidité.

L'influence de RH Peut-être minimisée si la base de données utilisée pour former le RNA est suffisamment diversifiée (on prend plusieurs valeurs de l'humidité pour entraîner le modèle de RNA). Mais, les grandes bases de données ont tendance à ralentir le processus d'entraînement.

Un certain nombre de modèles de prédiction et de classification de degré de pollution d'un isolateur pollué ont été examinés dans ce chapitre. Une courte description de chacune des méthodes de prédiction utilisées a été fournie.

Le choix d'un modèle de prédiction ou de classification n'est pas une tâche facile. Toutes les méthodes de prédiction ou de classification décrites dans la littérature ci-dessus ont des dispositifs distinctifs. Certains d'entre elles s'exécutent mieux que d'autres. Les modèles de série chronologique sont basés sur la théorie mathématique classique. Ainsi, le comportement de ces modèles, est bien compris. Les modèles artificiels du réseau neurologique fonctionnent généralement mieux pour des isolateurs qui montrent un comportement non linéaire, et aussi ils sont capables de traiter des relations fortement non linéaires.

Cependant, le succès d'employer les RNA dépend d'un certain nombre de questions de conception telles que le choix des entrées, les sorties, le nombre de couches cachées, le nombre de neurones utilisés dans chaque couche et les algorithmes d'apprentissage utilisés.

Plusieurs accents ont été mis sur le modèle de réseau neurologique parce qu'il a l'avantage unique qu'aucun rapport clair entre les variables d'entrée et la sortie ne doit être défini avant que le modèle soit employé dans le processus de prédiction. Une fois que l'entrée et la sortie sont choisies et introduits dans un RNA, le rapport entre elles est identifié par un processus d'auto apprentissage. L'exactitude de la prédiction peut être améliorée en ajustant l'architecture du RNA. En raison de ce dispositif unique, le temps et l'effort qui sont normalement exigés pour établir un modèle mathématique approprié dans une méthodologie conventionnelle de prédiction peut être réduit et limité.

CHAPITRE II
THEORIE DES
RESEAUX DE
NEURONES
ARTIFICIELS

Introduction

Le réseau de neurones artificiels (RNA) représente un modèle de calcul simplifié des réseaux de neurones biologiques, dans le but de reproduire les avantages du cerveau humain [29]. Les neurologues W. McCulloch et W. Pitts publièrent dès la fin des années 1950 les premiers travaux sur les réseaux de neurones, avec un article fondateur : What the frog's eye tells the frog's brain (Ce que l'œil d'une grenouille dit à son cerveau) [32]. Ils constituèrent ensuite un modèle simplifié de neurone biologique communément appelé neurone formel. Ils montrèrent que des réseaux de neurones formels simples peuvent théoriquement réaliser des fonctions logiques, arithmétiques et symboliques complexes [9].

Au cours des deux dernières décennies, nous avons pu constater un développement fulgurant des réseaux de neurones. Cet intérêt a démarré avec l'application réussie de cette technique puissante pour des problématiques très différentes, et dans des domaines aussi divers tels que le traitement efficace des données, la stratégie de jeu, le contrôle du véhicule, la chimie quantique, la robotique, le diagnostic médical, la reconnaissance de la parole et de l'image, la prise de décision dans les domaines commercial et financier, pour en nommer d'autres. L'implémentation des méthodes d'intelligence artificielle tel que RNA dans le domaine de la Haut Tension pour le but de protection des équipements HT a connu aussi un recours croissant avec des résultats impressionnants.

Dans un RNA, les informations sont traitées et transférées d'entrée en sortie via un ou plusieurs nœuds intermédiaires, appelés neurones artificiels. De plus, les informations sont transmises d'un neurone à un autre par une liaison de neurone ; et pour chaque lien d'entrée, il existe un poids associé qui représente la force de ce lien qui, à son tour, amplifie le signal d'entrée en conséquence.

II.1 Neurone formel

Un neurone formel (artificiel) est une unité de traitement qui reçoit des données en entrée, sous la forme d'un vecteur, et produit une sortie réelle. Cette sortie est une fonction des entrées et des poids des connexions

Un neurone est constitué de :

- Signaux d'entrée
- Poids des connexions
- Un additionneur permettant d'ajouter la somme pondérée des entrées
- Fonction d'activation
- Etat de sortie

II.1.1 Entrées et les poids des connexions

On note x_i ($1 \leq i < k$) les k informations parvenant au neurone. De plus, chacune sera plus ou moins valorisée vis à vis du neurone par le biais d'un poids. Un poids est simplement un coefficient w_i lié à l'information x_i . La i -ème information qui parviendra au neurone sera donc en fait $w_i * x_i$. Il y a toutefois un "poids" supplémentaire, qui va représenter ce que l'on appelle le coefficient *de* biais. Nous le noterons w_0 et le supposons lié à une information $x_0 = -1$. Nous verrons plus tard son utilité, dans la section **Fonction d'activation** [30].

II.1.2 Additionneur

Le neurone artificiel (qui est une modélisation des neurones du cerveau) va effectuer une somme pondérée de ses entrées plutôt que de considérer séparément chacune des informations. On définit une nouvelle donnée, in , par [31] :

$$in = \sum_{i=0}^k w_i \times x_i = \left(\sum_{i=0}^k w_i \times x_i \right) - w_0 \quad (I.1)$$

Cette donnée est passée à la fonction d'activation, qui fait l'objet de la prochaine section. C'est d'ailleurs pour ça que l'on peut parfois appeler un neurone une **unité de traitement**.

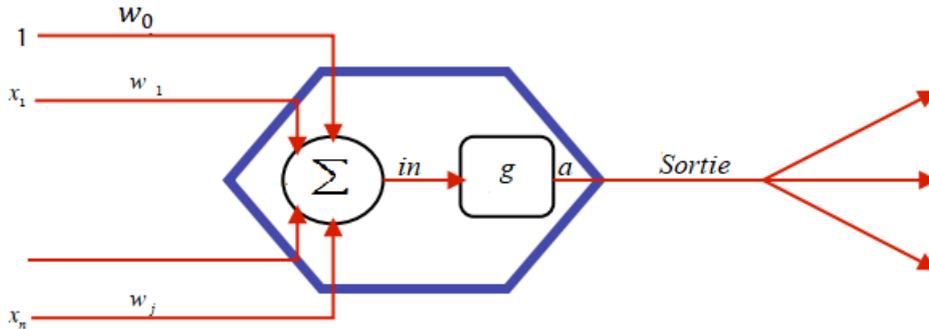


Figure II.1. Un neurone artificiel avec une fonction d'activation g .

II.1.3 Fonction d'activation

La fonction d'activation, ou fonction de transfert, est une fonction qui doit renvoyer un réel proche de 1 quand les "bonnes" informations d'entrée sont données et un réel proche de 0 quand elles sont "mauvaises". On utilise généralement des fonctions à valeurs continues comprises dans l'intervalle réel $[0,1]$ ou $[-1,1]$. Quand le réel est proche de 1, on dit que l'unité (le neurone) est **active** alors que quand le réel est proche de 0, on dit que l'unité est **inactive**. Le réel en question est appelé la **sortie** du neurone et sera noté a . Si la fonction d'activation est linéaire, le réseau de neurones se réduirait à une simple fonction linéaire.

En notant g la fonction d'activation, on obtient donc la formule donnant la sortie d'un neurone : [30]

$$a = g(in) = g\left(\sum_{i=0}^k w_i \times x_i\right) \quad (II.2)$$

Remarquez que le coefficient de biais est inclus dans la somme, d'où la formule plus explicite [30] :

$$a = g(in) = g\left(\left(\sum_{i=0}^k w_i \times x_i\right) - w_0\right) \quad (II.3)$$

Les différents types de neurones se distinguent par la nature g de leur fonction d'activation. Les principaux types sont :

Les 3 fonctions de transfert les plus utilisées

- La fonction d'activation log-sigmoïde
- La fonction d'activation tan-sigmoïde
- La fonction d'activation linéaire.

Les neurones à sortie sigmoïde sont souvent utilisés pour les problèmes de non-linéaire (prédiction ou classification), alors que les neurones à sortie linéaire sont utilisés pour les problèmes d'ajustement de fonctions.

II.1.3.1 Fonction d'activation log-sigmoïde

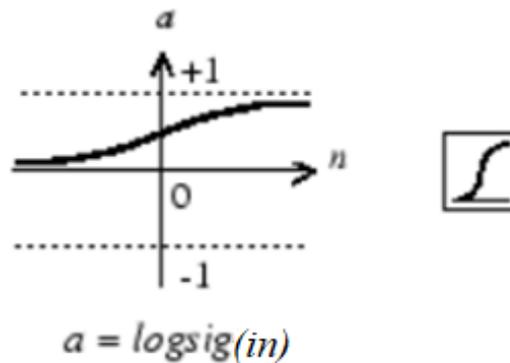


Figure II.2. Courbe de la fonction log-sigmoïde.

La fonction logsig génère des sorties entre 0 et 1 lorsque l'entrée nette du neurone passe de l'infini négatif à l'infini positif.

II.1.3.2 Fonction d'activation tan-sigmoïde

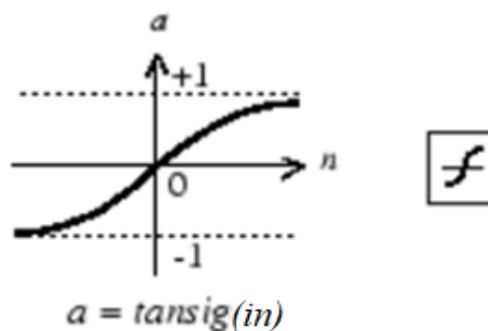


Figure II.3. Courbe de la fonction tan-Sigmoïde

II.1.3.3 Fonction d'activation linéaire.

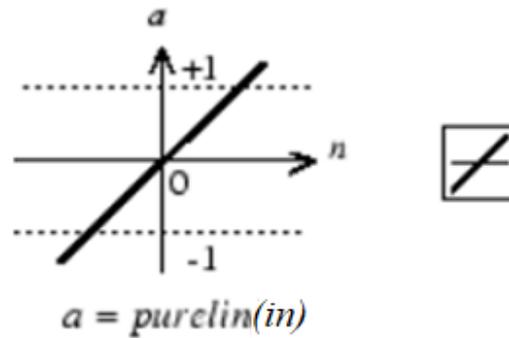


Figure II.4. Courbe de la fonction linéaire

Les trois fonctions de transfert décrites ci-dessus sont les fonctions de transfert les plus couramment utilisées pour les réseaux des neurones, mais d'autres fonctions de transfert différentiables peuvent être créées et utilisées.

En effet, dans les premiers temps, un réseau de neurones ne contenait qu'une couche d'entrée, un neurone et une couche de sortie. Toutefois, la sortie de ce réseau est linéaire, plane ou hyperplan et ne peut pas résoudre les problèmes de types XOR [31]. Comme nous le voyons dans la figure II.5 ci-dessous, aucune ligne droite ne permet de classer correctement les deux classes pour le problème du type XOR. Ces problèmes avaient empêché les scientifiques d'utiliser le réseau de neurones pendant un certain temps jusqu'à la découverte du réseau de neurones multicouches.

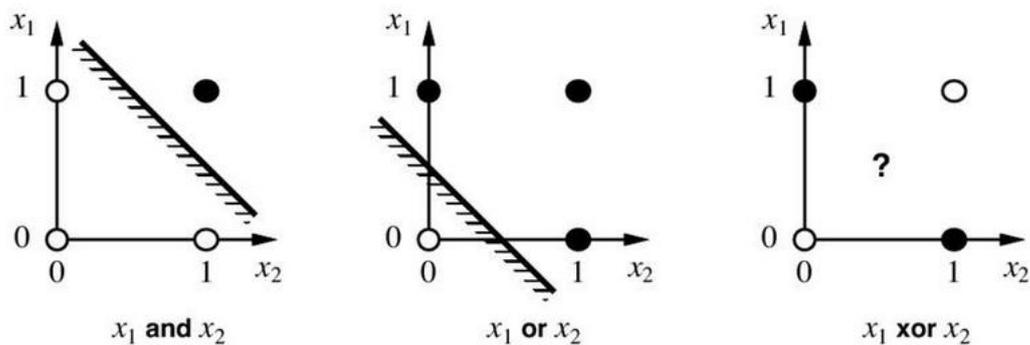


Figure II.5. Séparation linéaire des fonctions AND, OR, XOR [34]

II.2 Les différentes architectures des réseaux de neurones

Il existe deux architectures principales de réseau de neurones : Réseaux de Neurones Non-Bouclés (ou Réseaux Feedforward) et réseaux de Neurones Bouclés (ou Réseaux récurrent) :

II.2.1 Réseaux de Neurones Non-Bouclés

Dans les réseaux de neurones Non-Bouclés (FFNN), le signal circule dans un seul sens de la couche d'entrée à la couche de sortie et la sortie d'un neurone n'affecte pas les autres neurones de la même couche, FFNN est principalement utilisé pour les problèmes de classification et de régression.

FFNN possède une couche d'entrée, une couche de sortie et une ou plusieurs couches cachées (figure II.6). Les couches cachées sont les principales parties de traitement du réseau neuronal et affectent directement les résultats du réseau neuronal. Chaque couche cachée reçoit des informations de la couche d'entrée (pour la première couche cachée) ou de la couche cachée précédente (pour les autres couches cachées), traite et transmet ces informations à la couche suivante. En pratique, un réseau de neurones ne comporte probablement pas plus de deux couches cachées. Il existe de nombreuses matrices de pondération entre la couche d'entrée et la première couche cachée, entre les couches cachées et entre la dernière couche cachée et la couche de sortie.

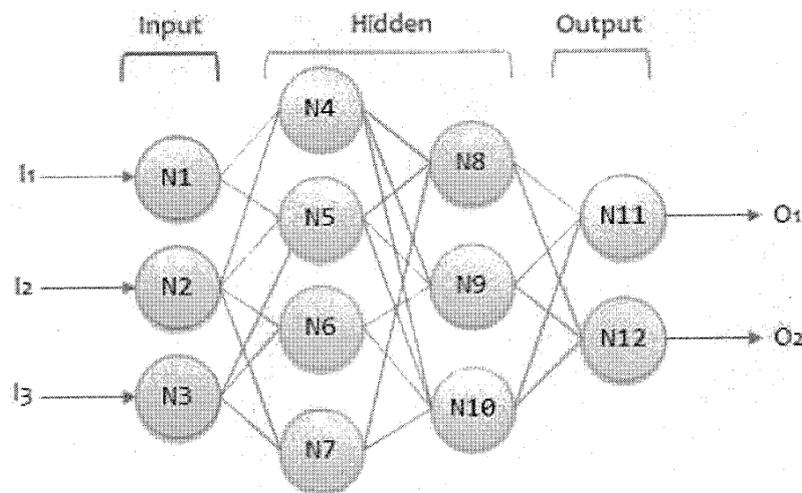


Figure II.6. Architecture d'un réseau de neurone non-bouclé [32]

Le nombre de neurones influe fortement sur le résultat du réseau, mais il n'existe actuellement aucune méthode permettant de déterminer le nombre idéal de neurones dans chaque couche cachée. Le concepteur de réseau de neurones doit essayer un nombre différent de neurones dans des couches cachées pour produire le nombre optimal pour le problème actuel.

Les types de réseaux de neurones non bouclés les plus utilisés sont :

II.2.1.1 Perceptron multicouche

Le réseau de neurones multicouches (PMC) déploie entre la couche d'entrée et la couche de sortie une ou plusieurs couches cachées dont les neurones ne sont pas connectés directement à la sortie. Ces couches cachées agissent comme des hyperplans supplémentaires pour aider le réseau à augmenter la capacité de séparation.

1. Nombre de couche cachée

Deux paramètres très importants à choisir quand on définit les couches cachées : combien de couches cachées il faut réellement avoir dans le réseau de neurones et combien de neurones seront dans chacune de ces couches. Nous allons d'abord examiner comment déterminer le nombre de couches cachées à utiliser avec le réseau de neurones.

Les problèmes nécessitant deux couches cachées sont rarement rencontrés. Cependant, les réseaux de neurones à deux couches cachées peuvent représenter des fonctions de formes quelconques. Il n'y a actuellement aucune raison théorique d'utiliser des réseaux de neurones comportant plus de deux couches cachées. En fait, pour de nombreux problèmes pratiques, il

n'ya aucune raison d'utiliser plus d'une couche cachée. Le tableau ci-dessous résume les capacités des architectures de réseau neuronal avec diverses couches cachées.

Tableau II.1. Détermination du nombre de couches cachées [36]

Nombre de couches cachées	Résultat
0	Uniquement capable de représenter des fonctions ou des décisions séparables linéaires.
1	Peut approximer n'importe quelle fonction contenant un mapping continu d'un espace fini à un autre.
2	Peut représenter une limite de décision arbitraire avec une précision arbitraire avec des fonctions d'activation rationnelles et peut se rapprocher de tout lisse cartographie avec précision.

Décider le nombre de couches de neurones cachées n'est qu'une petite partie du problème. Il faut également déterminer le nombre de neurones dans chacune de ces couches cachées. Ce processus est couvert dans la section suivante.

2. Nombre de neurone par couche cachée

Décider le nombre de neurones dans les couches cachées est une partie très importante de la décision de l'architecture globale de réseau de neurones. Bien que ces couches n'interagissent pas directement avec l'environnement externe, elles ont une influence considérable sur le résultat final. Le nombre de couches cachées et le nombre de neurones dans chacune de ces couches cachées doivent être soigneusement pris en compte. Si on utilise trop peu de neurones dans les couches cachées, on obtient ce qu'on appelle un sous-ajustement. Il y aura un sous-ajustement lorsqu'il y a trop peu de neurones dans les couches cachées pour détecter correctement les signaux dans un ensemble de données compliqué. Utiliser trop de neurones dans les couches cachées peut entraîner plusieurs problèmes. Premièrement, un trop grand nombre de neurones dans les couches cachées peut entraîner un sur-ajustement. Un sur-ajustement survient lorsque le réseau de neurones a une capacité de traitement de l'information si importante que la quantité limitée d'informations contenues dans l'ensemble d'apprentissage ne suffit pas pour entraîner tous les neurones des couches cachées. Un deuxième problème peut survenir même lorsque les données d'apprentissage sont suffisantes, un nombre excessivement élevé de neurones dans les couches cachées peut augmenter le temps nécessaire de l'apprentissage du réseau. Le temps d'entraînement peut augmenter à tel point qu'il est impossible d'entraîner correctement le réseau de neurones. De toute évidence, il faut trouver un compromis entre trop et trop peu de neurones dans les couches cachées [36].

Il existe de nombreuses méthodes empiriques permettant de déterminer le nombre correct de neurones à utiliser dans les couches cachées, telles que les suivantes :

- Le nombre de neurones cachés doit être compris entre la taille de la couche d'entrée et la taille de la couche de sortie.
- Le nombre de neurones cachés doit être $2/3$ de la taille de la couche d'entrée, plus ceux de la couche de sortie.

- Le nombre de neurones cachés doit être inférieur à deux fois la taille de la couche d'entrée.

Ces trois règles constituent un point de départ à considérer. En fin de compte, le choix d'une architecture pour le réseau de neurones résultera en essais et erreurs. Afin d'assurer la capacité du réseau à pouvoir être généraliser, le nombre de nœuds doit être maintenu aussi bas que possible. Si on prend un grand nombre de nœuds, le réseau devient une banque de mémoire capable d'exiger la perfection d'ensemble d'apprentissage, i.e. : le réseau ne fonctionne pas correctement sur des échantillons ne faisant pas partie de l'ensemble d'apprentissage [36].

Les réseaux multicouches utilisent diverses techniques d'apprentissage, les plus courantes étant **Backpropagation**. Ici, les valeurs de sortie sont comparées à la réponse correcte pour calculer la valeur d'une fonction d'erreur prédéfinie. Par diverses techniques, l'erreur est ensuite renvoyée à travers le réseau. À l'aide de cette information, l'algorithme ajuste les poids de chaque connexion afin de réduire légèrement la valeur de la fonction d'erreur. Après avoir répété ce processus pendant un nombre suffisamment grand de cycles d'apprentissage, le réseau converge généralement vers un état dans lequel l'erreur de calcul est faible. Dans ce cas, on dirait que le réseau a appris une certaine fonction cible. Pour ajuster correctement les poids, on applique une méthode générale d'optimisation non linéaire appelée **gradient descente**.

II.2.1.2 Réseaux de Neurones à Fonctions à Base Radiale

Les Réseaux de Neurones Automatisés est connu sous le nom de Fonctions à Base Radiale (RBF). Les réseaux RBF constituent probablement le type de réseaux de neurones le plus utilisé après les PMC, ils peuvent être utilisés pour la prédiction ou la classification (linéaire ou non linéaire).

Le schéma ci-dessous représente la différence entre des réseaux de neurones PMC et RBF dans un espace à deux dimensions pour les données d'entrée. Une manière de séparer les clusters d'entrées consiste à tracer les plans adaptés pour séparer les différentes classes entre elles, cette méthode est utilisée par les réseaux PMC. Une autre approche consiste à ajuster chaque classe de données d'entrée dans des cercles à l'aide de fonctions Gaussiennes de base, cette méthode est utilisée par les réseaux RBF.

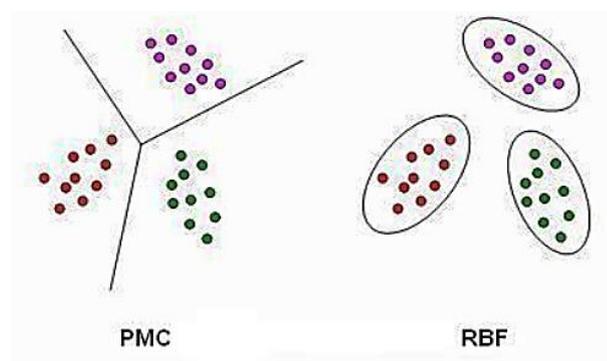


Figure II.7. Différence entre les réseaux PMC et RBF

L'approche RBF est plus intuitive que la MLP. Une RBF effectue une classification en mesurant la similarité de l'intrant avec les exemples du jeu d'entraînement. Chaque neurone RBF stocke un « prototype », qui n'est que l'un des exemples tirés de l'entraînement. Lorsque nous voulons classer une nouvelle entrée, chaque neurone calcule la distance euclidienne

entre l'entrée et son prototype. En gros, si l'entrée ressemble plus aux prototypes de classe A que les prototypes de classe B, elle est classée dans la classe A.

Cependant, depuis 1988, le séminal de Broomhead et Lowe [33]. Les réseaux à fonction de base radiale (réseaux RBF) sont traditionnellement associés à des réseaux monocouches avec des fonctions radiales dans chaque neurone de la couche cachée, comme illustré à la figure II.8.

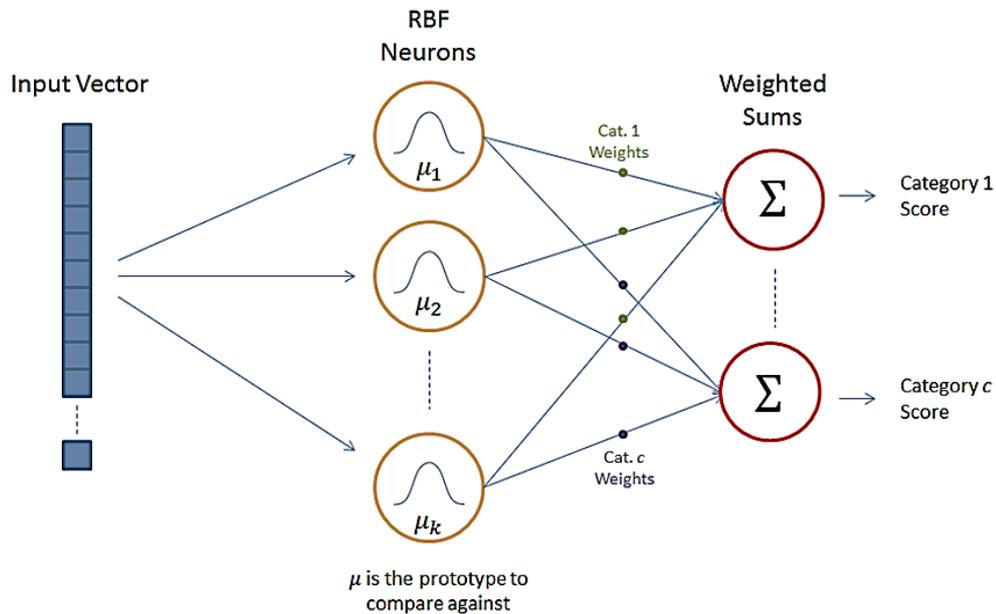


Figure II.8. Architecture d'un réseau RBF [38]

L'illustration ci-dessus montre l'architecture typique d'un réseau RBF. Il se compose d'un vecteur d'entrée, d'une couche de neurones RBF et d'une couche de sortie avec un nœud par catégorie ou classe de données. Un réseau RBF est non linéaire si les fonctions de base peuvent être déplacées ou modifiées, ou s'il existe plusieurs couches cachées.

1. Vecteur d'entrée

Le vecteur d'entrée est le vecteur à n dimensions qu'il faut classer. Le vecteur d'entrée entier est montré à chacun des neurones RBF.

2. Neurones RBF

Chaque neurone RBF stocke un vecteur "prototype" qui n'est qu'un des vecteurs de l'ensemble d'apprentissage. Chaque neurone RBF compare le vecteur d'entrée à son prototype et fournit une valeur comprise entre 0 et 1, qui est une mesure de la similarité. Si l'entrée est égale au prototype, la sortie de ce neurone RBF sera égale à 1. À mesure que la distance entre l'entrée et le prototype augmente, la réponse décroît de façon exponentielle vers 0. La forme de la réponse du neurone RBF est une courbe en cloche, comme illustré dans le diagramme d'architecture de réseau RBF en dessus.

La valeur de réponse du neurone est également appelée sa valeur « d'activation ». Le prototype de vecteur est également souvent appelé le « centre » du neurone, car il s'agit de la valeur au centre de la courbe en cloche.

3. Nœuds de sortie

La sortie du réseau consiste en un ensemble de nœuds, un par catégorie que nous essayons de classer ou de prédire. Chaque nœud de sortie calcule une sorte de score pour la catégorie associée. Généralement, une décision de classification est prise en affectant l'entrée à la catégorie avec le score le plus élevé.

Le score est calculé en prenant une somme pondérée des valeurs d'activation de chaque neurone RBF. Par somme pondérée, nous entendons qu'un nœud de sortie associe une valeur de poids à chacun des neurones RBF et multiplie l'activation du neurone par ce poids avant de l'ajouter à la réponse totale.

Étant donné que chaque nœud de sortie calcule le score pour une catégorie différente, chaque nœud de sortie a son propre ensemble de pondérations. Le nœud de sortie donnera généralement un poids positif aux neurones RBF appartenant à sa catégorie et un poids négatif aux autres.

4. Fonction d'activation des neurones RBF

Chaque neurone RBF calcule une mesure de la similarité entre l'entrée et son vecteur prototype (tiré du jeu d'entraînement). Les vecteurs d'entrée qui sont plus similaires au prototype renvoient un résultat plus proche de 1. Il existe différents choix possibles de fonctions de similarité, mais le plus populaire est basé sur le gaussien. Voici ci-dessous l'équation pour un gaussien avec une entrée unidimensionnelle [38].

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (II.4)$$

x est l'entrée, μ la moyenne et σ l'écart type. Ceci produit la courbe en cloche familière montrée ci-dessous, qui est centrée sur la moyenne, μ (dans le graphique ci-dessous, la moyenne est 5 et σ est 1).

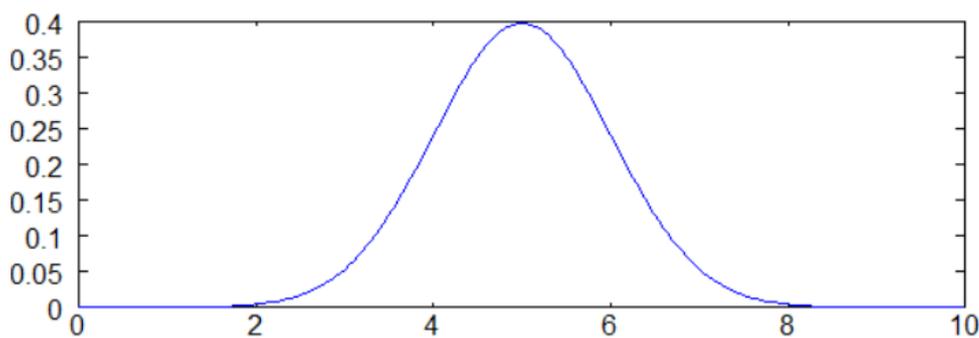


Figure II.9. Courbe de la fonction gaussienne

La fonction d'activation des neurones RBF est légèrement différente et s'écrit généralement comme suit [38] :

$$\varphi(x) = e^{-\beta\|x-\mu\|^2} \quad (II.5)$$

Dans la distribution gaussienne, μ fait référence à la moyenne de la distribution. Ici, c'est le vecteur prototype qui est au centre de la courbe en cloche.

Pour la fonction d'activation, ϕ , la valeur de l'écart type, σ , n'est pas intéressant, donc procédant à quelques modifications simplificatrices.

Le premier changement c'est de supprimer le coefficient externe, $1 / (\sigma * \sqrt{2 * \pi})$. Ce terme contrôle normalement la hauteur de la gaussienne. Ici, cependant, il est redondant avec les poids appliqués par les nœuds de sortie. Pendant l'entraînement, les nœuds de sortie apprendront le coefficient ou le «poids» correct à appliquer à la réponse du neurone.

Le deuxième changement c'est de remplacer le coefficient interne, $1 / (2 * \sigma^2)$, par un seul paramètre «beta». Ce coefficient bêta contrôle la largeur de la courbe en cloche. Encore une fois, dans ce contexte, nous ne nous soucions pas de la valeur de σ , nous tenons simplement à ce qu'il existe un coefficient qui contrôle la largeur de la courbe en cloche. Simplifiant donc l'équation en remplaçant le terme par une seule variable.

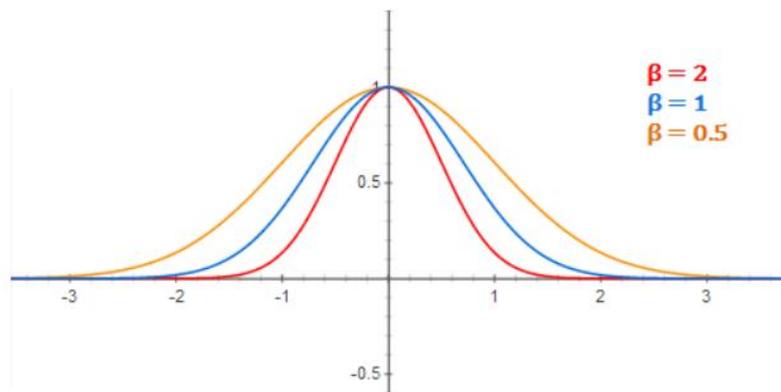


Figure II.10. Activation du neurone RBF pour différentes valeurs de bêta [38]

Il y a également un léger changement de notation lors de l'application de l'équation à des vecteurs à n dimensions. La notation en double barre dans l'équation d'activation indique que nous prenons la distance euclidienne entre x et μ et que nous quadrillons le résultat. Pour le gaussien à une dimension, cela se résume à seulement $(x - \mu)^2$.

Il est important de noter que la métrique sous-jacente utilisée ici pour évaluer la similarité entre un vecteur d'entrée et un prototype est la distance euclidienne entre les deux vecteurs.

De plus, chaque neurone RBF produira sa plus large réponse lorsque l'entrée sera égale au vecteur prototype. Cela permet de prendre ça comme une mesure de similarité et d'ajouter les résultats de tous les neurones RBF. Lorsque le vecteur prototype est quitté, la réponse diminue de manière exponentielle. Rappelant de l'illustration de l'architecture RBF que le nœud de sortie de chaque catégorie prend la somme pondérée de chaque neurone RBF du réseau - en d'autres termes, chaque neurone du réseau aura une certaine influence sur la décision de classification. La décroissance exponentielle de la fonction d'activation signifie toutefois que les neurones dont les prototypes sont éloignés du vecteur d'entrée contribueront en réalité très peu au résultat.

II.2.2 Réseaux de Neurones Bouclés

Dans les réseaux de neurones récurrents (RNN), le signal est transféré dans les deux sens d'un neurone à un autre neurone, formant des cycles dirigés. Contrairement aux réseaux FFNN, les neurones actuels reçoivent non seulement les entrées de la couche précédente, mais aussi celles des couches suivantes ou même d'autres neurones de la même couche. Il permet au réseau au neurones de stocker les états fréquemment modifiés (comme la mémoire) jusqu'à ce que le réseau atteigne son point d'équilibre. Étant donné que RNN peut stocker des informations, les sorties des échantillons d'entraînement précédents sur tout le réseau sont également affectées. Avec la capacité de stocker des informations de l'itération précédente, ce type de réseau de neurones est principalement utilisé pour les séries chronologiques et les problèmes séquentiels.

La figure II.11 montre deux exemples de réseaux récurrents. Le premier est un simple multicouche qui utilise un vecteur d'entrée qui contient les copies des activations de la couche de sortie du réseau et le deuxième est un réseau à mémoire se distingue du premier par la présence des unités mémoires [9].

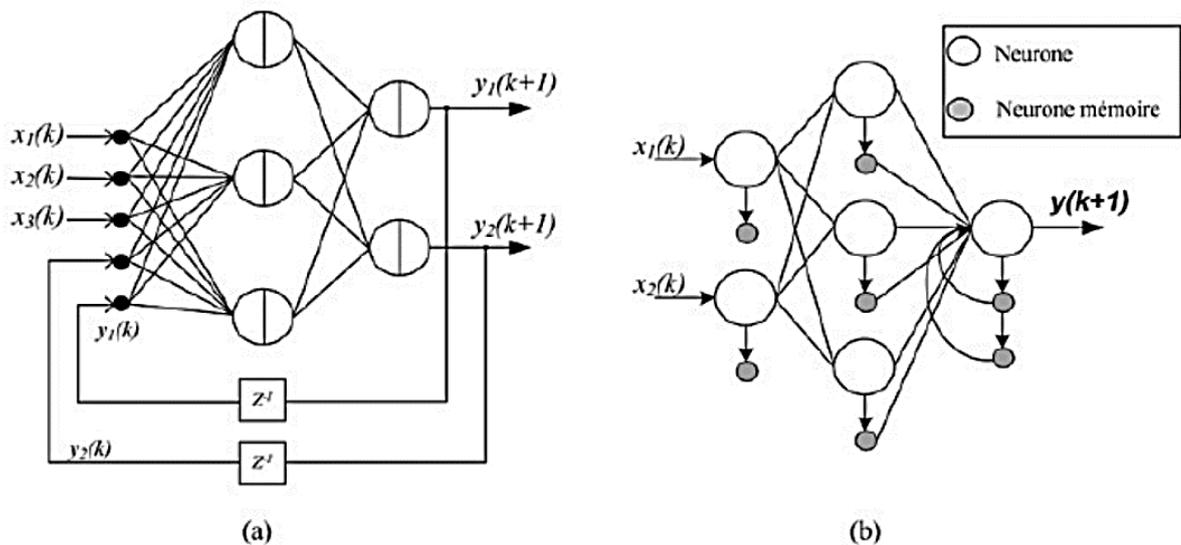


Figure II.11. Deux exemples de réseaux récurrents [9].

Les deux types les plus simples de réseaux de neurones bouclés sont :

II.2.2.1 Réseaux Elman

Un réseau Elman est un réseau de 3 couches (couches x, y, et z) complété d'un ensemble de "unités de contexte" (couche u). La couche du milieu (cachée) est connectée à ces unités de contexte, et a des poids. A chaque étape, l'entrée est retransmise et une règle d'apprentissage est appliquée. Les connexions finales fixées enregistrent une copie des valeurs précédentes des unités cachées dans les unités de contexte (puisqu'elles propagent les connexions avant que la règle d'apprentissage soit appliquée). Donc le réseau peut maintenir une sorte d'état, lui permettant d'exécuter des tâches telles que la prédiction séquentielle qui est au-delà de la puissance d'un perceptron multicouche standard [34].

Le réseau est entraîné avec l'algorithme de **Gradient Descent-backpropagation** avec un taux d'apprentissage adaptatif.

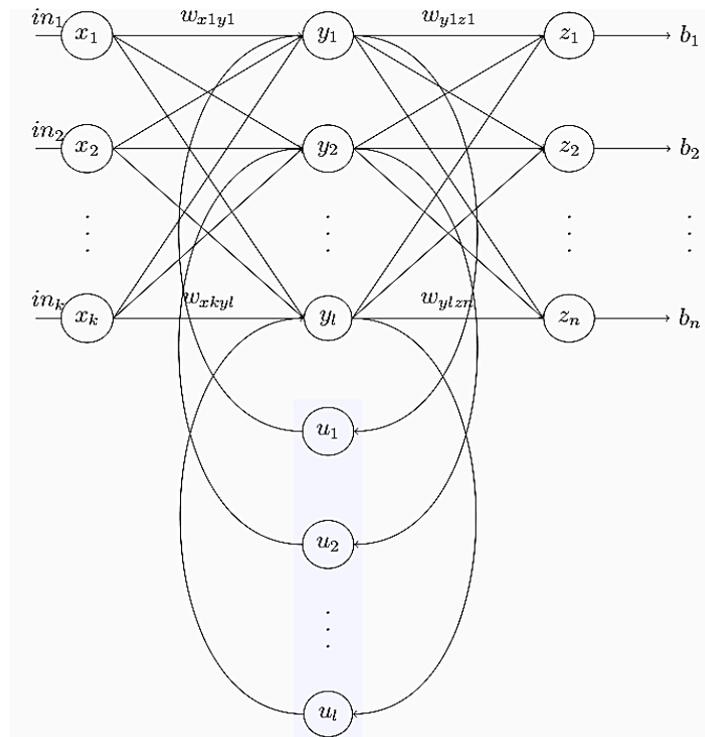


Figure II.12. Réseau Elman [34]

II.2.2.2 Réseaux Jordan

Les réseaux Jordan sont similaires aux réseaux Elman. Les unités de contexte sont alimentées par la couche de sortie à la place de la couche cachée. Les unités de contexte dans un réseau Jordan mentionnent aussi la couche d'état. Elles ont une connexion récurrente à elles-mêmes.

II.3 Apprentissage des Réseaux de Neurones

L'apprentissage dans le contexte des réseaux de neurones, est le processus de modification des poids de connexions (y compris les biais) ou plus rarement du nombre de couches et de neurones, afin d'adapter le traitement effectué par le réseau à une tâche particulière [9].

II.3.1 Apprentissage supervisé :

Dans ce cas, un superviseur (ou expert humain) fournit une valeur ou un vecteur de sortie (appelé cible ou sortie désirée), que le réseau de neurones doit associer à un vecteur d'entrée. L'apprentissage consiste dans ce cas à ajuster les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

Dans le cas de l'apprentissage supervisé, la robustesse de l'algorithme dépendra de la précision de son entraînement. Un algorithme apprenant des contenus supervisés produit une carte interne qui permet sa réutilisation pour classifier et prédire de nouvelles quantités de données

Il existe plusieurs algorithmes d'apprentissage supervisé, ci-dessous on peut citer quelques algorithmes d'apprentissages usuels dans un réseau de neurone :

II.3.1.1 Algorithme de perceptron

Considérons un neurone avec une entrée $X = (x_1, x_2 \dots x_m)$, poids d'entrée $w = (w_1, w_2 \dots w_m)$ et fonction d'activation g pouvant être linéaire ou non linéaire. Il reçoit les entrées X et renvoie la sortie $y(X) = g(w^T X)$, où T représente la transposition d'une matrice.

Initialisez toutes les pondérations sur des valeurs aléatoires dans $[-1, 1]$, y compris une pondération pour le nœud de polarisation, et ajoutez l'élément $x_0 = 1$ à toutes les entrées. Répétez le processus suivant jusqu'à ce qu'il n'y ait plus d'amélioration ou après un certain nombre de générations. Soit y_k le vecteur de sortie du réseau à l'itération k et g est la fonction d'activation, η le taux d'apprentissage et t la cible (résultat attendu). Pour chaque vecteur d'entrée X avec la matrice de pondération actuelle w , [32] :

$$y_k = g(w^T x) = \begin{cases} 1 & \text{Si } w^T x > 0 \\ 0 & \text{Si } w^T x \leq 0 \end{cases} \quad (II.6)$$

$$w_{k+1} = w_k + \eta(t - y_k) * x \quad (II.7)$$

II.3.1.2 Algorithme Back-propagation

Soit t la cible, y la sortie du réseau de neurones et E l'erreur du réseau au niveau de la couche de sortie. Quelle que soit la méthode utilisée, nous voulons minimiser l'erreur E . Dans l'entraînement au perceptron, $E = t - y$, ce qui est bien, car il n'existe qu'un ensemble de poids et une sortie.

Avec un réseau neuronal multicouche, il existe une ou plusieurs couches cachées et plusieurs sorties. Il est donc plus difficile de déterminer quel ensemble de poids a provoqué l'erreur, les poids d'entrée ou les poids cachés. C'est encore plus compliqué quand il y a quelques couches cachées. S'il y a deux erreurs, l'une est $+1$ et l'autre -1 , E serait égal à 0 et $E = \text{somme}(t_i - y_i)$ (où y_i est la sortie de neurone indice i) n'est pas exacte car le réseau ne reconnaît pas les erreurs. L'erreur quadratique moyenne peut résoudre ce problème car même s'il existe une erreur $+1$ et une erreur -1 , l'erreur totale n'est pas [32].

$$E = \frac{1}{2} \sum_{i=0}^m (t_i - y_i)^2 \quad (II.8)$$

Le nombre $1/2$ n'affecte pas la signification de l'erreur bien qu'il puisse affecter certains nombres lors du calcul, et il est plus facile de calculer la dérivée de la fonction car elle annule l'exposant. Le but de l'entraînement est de minimiser E , et le négatif de la pente de E au point actuel indique la direction la plus rapide que E devrait suivre pour que l'amélioration ou la réduction soit la plus élevée. Une fois que l'entraînement commence, le seul paramètre pouvant être contrôlé est le poids et les biais. Cependant, il existe des poids d'entrée et un ou plusieurs

pois cachés. Pour simplifier, nous supposons qu'il n'y a qu'une couche cachée. Soit v_{ij} un vecteur de poids reliant l'entrée i^{th} et le neurone j^{th} dans la couche cachée et w_{jk} le vecteur de poids reliant le neurone j^{th} à la couche cachée et le neurone k^{th} à la couche de sortie, puis de [32],

$$\frac{\partial E}{\partial w_{jk}} = \sigma_0 a_j = (y_k - t_k) y_k (y_k - t_k) a_j \text{ et } \frac{\partial E}{\partial v_{ij}} = a_j (1 - a_j) \left(\sum_k \sigma_0 w_{jk} \right) x_i \quad (II.9)$$

où δ_0 est l'erreur à la couche de sortie et a_j est la fonction d'activation au neurone j^{th} .
Mettant à jour v_{ij} et w_{jk} en conséquence [32],

$$v_{ij} = v_{ij} - \eta * \frac{\partial E}{\partial v_{ij}} \text{ et } w_{jk} = w_{jk} - \eta * \frac{\partial E}{\partial w_{jk}} \quad (II.10)$$

L'entraînement BP comporte deux phases : en avant et en arrière. Lors de la propagation vers l'avant, l'entrée est transmise au réseau et la sortie de la couche de sortie est utilisée pour calculer l'erreur à l'aide de l'équation d'erreur E (Sum Square Error). En phase arrière, la sortie de la couche de sortie est utilisée pour calculer l'erreur et la rétroaction vers la couche cachée afin de réajuster le poids sous la forme indiquée ci-dessus.

II.3.1.3 Algorithme Gradient Descent

L'algorithme de la plus forte pente est un algorithme d'optimisation à direction de descente. Deux quantités sont à définir : la direction de descente et le pas de descente, qui vise à minimiser une fonction, alors le déplacement sera systématiquement vers le bas (figure II.13) en suivant les lignes de plus grande pente, celle-ci étant mathématiquement calculée par le gradient.

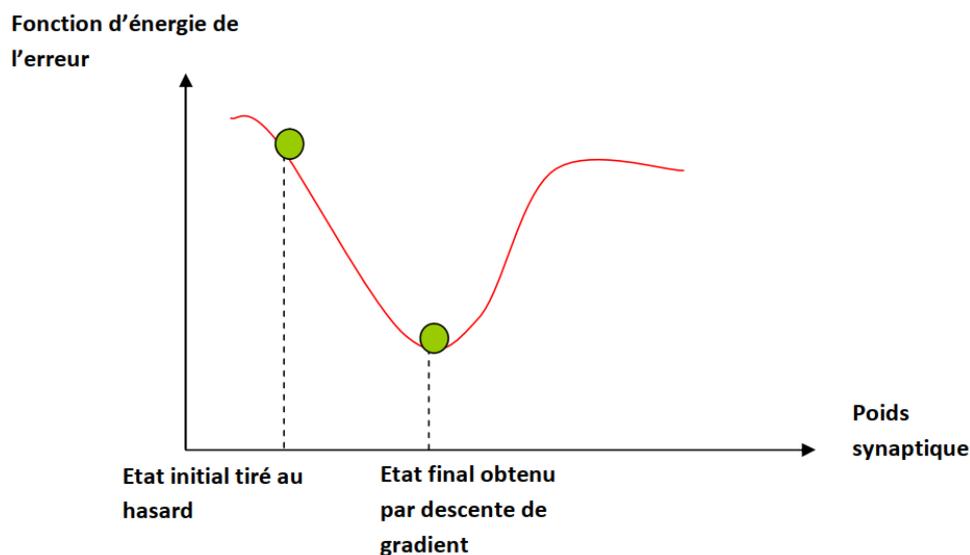


Figure II.13. La descente de gradient [35]

L'algorithme utilise les conventions de notation représentées sur sa description édifée sur la figure II.14.

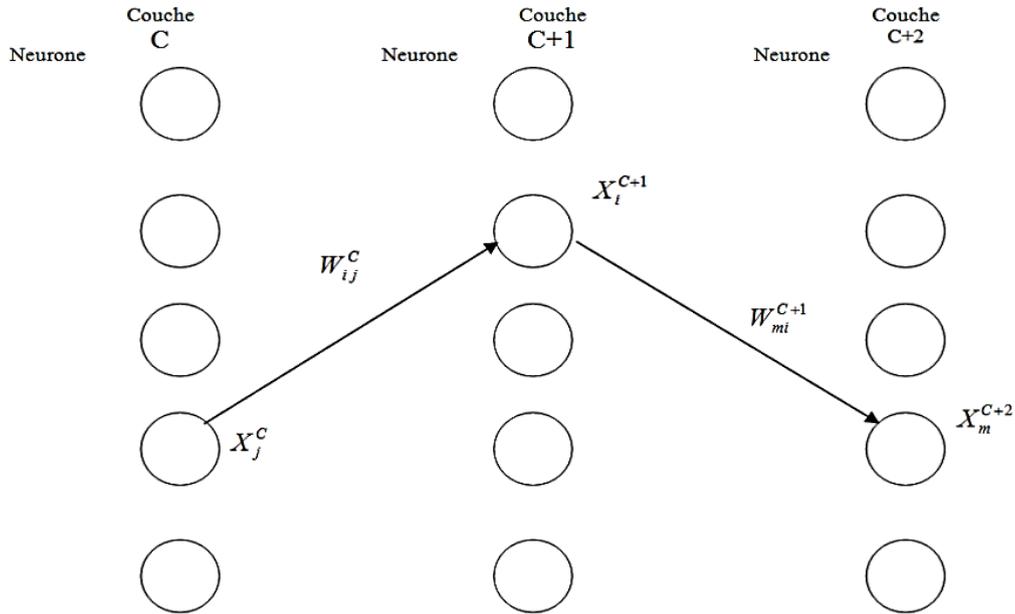


Figure II.14. Algorithme de gradient descent

- Soit n le nombre total des exemples d_i , $i=1 \dots n$ de la base d'apprentissage et N_n le nombre de cellule de la dernière couche.
- La couche de rang C contient N_c cellules repérées X_i^C , $i=1 \dots N_c$.
- Pour un réseau contenant N couches, $C=1$ et $C=N$ correspondent à la couche d'entrée et à la couche de sortie.

Soit X_i^{C+1} la sortie de la cellule numéro i de la couche $C+1$, son contenu vaut :

$$X_i^{C+1} = f \left(\sum_{j=0}^{N_c} w_{ij}^C X_j^C \right) = f(A_i^{C+1}) \quad (II.11)$$

f : Fonction sigmoïde

$$A_i^{C+1} = W_i^{tr} X^C \quad (II.12)$$

- En désignant par X_{ik}^N la sortie de la cellule de rang i de la dernière couche associée à l'exemple X_k présenté sur la couche d'entrée, par d_i^k la sortie souhaitée, ou k varie de 1 à n et N_n est le nombre de cellules de la dernière couche, la fonction quadratique de l'erreur a pour expression :

$$\bullet \quad R = \sum_{i=0}^n \sum_{i=0}^n \|X_{ik}^N - d_i^k\|^2 \quad (II.13)$$

Cette fonction de coût fait appel de façon implicite à tous les éléments du réseau et à les poids synaptiques un élément de la dernière cellule est fourni par :

$$X_i^N = f \left(\sum_{j=0}^{N-1} W_{ij}^{N-1} X_j^{N-1} \right) \quad (II.14)$$

Une règle d'apprentissage possible consisterait à minimiser la fonction R avec la technique des moindres carrés, la règle de gradient descent utilise la règle de Windrow-Hoff en minimisant la fonction de coût évalué lors de la présentation de l'exemple d_k

$$R^* = \sum_{i=0}^N (X_{ik}^N - d_i^k)^2 \quad (II.15)$$

La loi d'adaptation du vecteur poids W s'écrit :

$$W_k = W_{k-1} - \alpha_k \nabla_k(W) \quad (II.16)$$

La loi de pondération des poids sera notée par :

$$W_{ij}^k = W_{ij}^{k-1} - \alpha_k \frac{\partial R^*}{\partial W_{ij}} \quad (II.17)$$

$$\frac{\partial R^*}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) \frac{\partial X_{ik}^N}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) \frac{\partial X_{ik}^N}{\partial A_{ij}^{N-1}} \frac{\partial A_{ik}^{N-1}}{\partial W_{ij}^{N-1}} \quad (II.18)$$

En considérant la dernière couche, on obtient pour la sortie désirée d_i^k

Avec

$$A_{ij}^N = \sum_{j=0}^{j=Nn-1} W_{ij}^{N-1} X_{jk}^{N-1} \text{ et } \frac{\partial A_{ik}^{N-1}}{\partial W_{ij}^{N-1}} = X_{jk}^{N-1} \quad (II.19)$$

Compte tenu de II.14

$$\frac{\partial X_{ik}^N}{\partial A_{ik}^N} = f'(A_{ik}^N) \quad (II.20)$$

D'où

$$\frac{\partial R^*}{\partial W_{ij}^{N-1}} = 2(X_{ik}^N - d_i^k) f'(A_{ik}^N) X_{jk}^{N-1} \quad (II.21)$$

Le calcul des poids de la dernière couche dépend des éléments des couches précédentes.

- Pour la couche C on peut écrire de façon similaire

$$\frac{\partial R^*}{\partial W_{ij}^C} = \frac{\partial R^*}{\partial A_{ik}^{C+1}} \cdot \frac{\partial A_{ik}^{C+1}}{\partial W_{ij}^C} \quad (II.22)$$

Avec

$$\frac{\partial A_{ik}^{C+1}}{\partial W_{ij}^C} = X_{ik}^C \quad (II.23)$$

$$\frac{\partial R^*}{\partial A_{ik}^{C+1}} = \sum_m \frac{\partial R^*}{\partial A_{mk}^{C+2}} \frac{\partial A_{mk}^{C+2}}{\partial A_{ik}^{C+1}} \quad (II.24)$$

- L'indice m correspond au nombre de termes qui contribue à la fonction de coût lorsque l'on passe de la couche c+1 à la couche c+2

$$\frac{\partial A_{mk}^{C+2}}{\partial A_{ik}^{C+1}} = \frac{\partial [\sum_{i=1}^{n_{c+2}} W_{mi}^{C+1} f(A_{ik}^{C+1})]}{\partial A_{ik}^{C+1}} = W_{mi}^{C+1} f'(A_{im}^{C+1}) \quad (II.25)$$

n_{c+2} Représente le nombre de cellules de la couche (c+2) augmenté d'une unité et finalement on obtient :

$$\frac{\partial R^*}{\partial W_{ij}^C} = \sum_{i=0}^n \frac{\partial R^*}{\partial A_{mk}^{C+2}} W_{mi}^{C+1} f'(A_{ik}^{C+1}) X_{jk}^C \quad (II.26)$$

L'algorithme de gradient descent se résume comme suite :

- Pour la dernière couche, les poids sont adaptés suivant la loi

$$W_{ijk}^N = W_{ijk-1}^{N-1} - \alpha_k (X_{ik}^N - d_i^k) f'(A_{ik}^N) X_{jk}^{N-1} \quad (II.27)$$

- Pour les couches intermédiaires on a

$$W_{ijk}^N = W_{ijk}^C - \alpha_k \sum_{m=0}^{m=nc+2} \frac{\partial R^*}{\partial A_{mk}^{C+2}} W_{mik}^{C+1} f'(A_{ik}^{C+1}) X_{jk}^C \quad (II.28)$$

II.3.1.4 Algorithme Gradient Descent avec un terme de moment

La formule d'adaptation des poids synaptiques, fournie par l'algorithme de gradient descent est souvent modifiée par l'ajout d'un terme de moment. Dans ce cas, la valeur d'un poids synaptique n'est plus seulement adaptée proportionnellement à la dérivée de la fonction de coût par rapport à ce poids, mais est également modifiée en fonction de la correction appliquée à l'instant précédent. Sous forme mathématique, la formule d'adaptation des poids synaptiques est :

$$W_{1ij}(\tau + 1) - W_{t,ij}(\tau) = -\eta \frac{\partial E^{(k)}}{\partial W_{t,ij}} + \beta (W_{t,ij}(\tau) - W_{t,ij}(\tau - 1)) \quad (II.29)$$

Cette expression peut être réécrite :

$$W_{l,ij}(\tau + 1) - W_{t,ij}(\tau) = -\eta \sum_{i=0}^n \beta^q \frac{\partial E^{(k-q)}}{\partial W_{t,ij}} \quad (II.30)$$

η : est le taux d'apprentissage, $\eta > 0$.

E : fonction de coût à minimiser.

τ : Vecteur des variances générales des caractéristiques, avec:

$$\tau = [\sigma_1^2 \ \sigma_2^2 \ \dots \ \dots \ \sigma_d^2]^T \quad (II.31)$$

σ_i^2 Est le paramètre de normalisation de ce neurone (W correspond ici aux coordonnées du centre de la Gaussienne).

β : $0 \leq \beta \leq 1$, est un paramètre qui est appelé terme de moment [36].

Grâce à l'utilisation d'un terme de moment, la direction de recherche du minimum a un instant donné, est une somme pondérée des gradients actuel et précédent. La pondération qui intervient, est telle que l'importance relative d'un gradient décroît exponentiellement au fur et à mesure que ce dernier est éloigné dans le temps. Le fait de prendre en compte plusieurs gradients consécutifs aide les poids synaptiques à traverser les sections plates de la surface de la fonction de coût, après qu'ils en aient descendu des sections abruptes (voir figure II.15). Ceci permet en outre de modifier les poids synaptiques, non plus à l'aide d'un même taux d'adaptation pour l'ensemble de ceux-ci, mais selon un taux qui est propre à chaque poids, et qui est dépendant de son histoire particulière.

Les valeurs des paramètres η et β doivent être déterminées empiriquement, de manière à limiter la fréquence d'apparition de deux phénomènes qui sont opposés, mais qui conduisent tous deux à un net ralentissement de l'évolution de l'apprentissage

D'une part, lorsque les valeurs de ces paramètres sont faibles, les poids synaptiques sont peu modifiés quand une région de faible pente de la fonction de coût est rencontrée ; l'ensemble du système n'évolue alors que lentement

D'autre part, lorsque les valeurs de ces paramètres sont élevées, des corrections d'amplitude importante sont appliquées aux poids synaptiques dans les régions de forte pente de la fonction de coût, ce qui peut entraîner une augmentation de la valeur de cette dernière. Il en résulte ainsi également un ralentissement de l'évolution de l'apprentissage

L'apprentissage du perceptron multicouche nécessite donc une phase de méta-optimisation, au cours de laquelle des valeurs des coefficients a et b , conduisant au meilleur compromis entre ces deux situations extrêmes, doivent être recherchées.

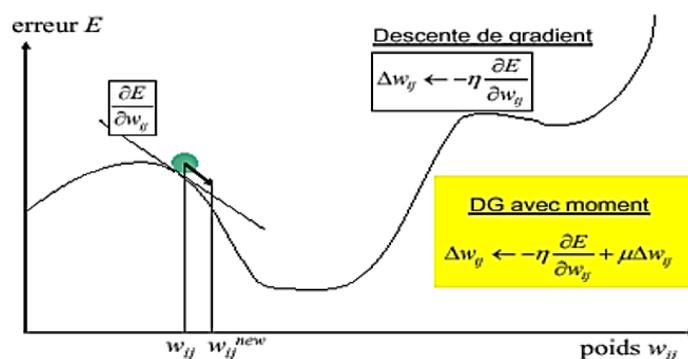


Figure II.15. Apprentissage avec l'utilisation d'un terme du moment [36]

II.3.2 Apprentissage non supervisé

Dans cet apprentissage, les données ne contiennent pas d'informations sur la sortie désirée. Il n'y a pas besoin de l'intervention d'un être humain, car l'algorithme va par lui-même comprendre comment différencier deux objets différents cherchant leurs corrélations. Étant donné qu'un algorithme ne peut pas tout simplement savoir ce qui constitue ces objets, donc la méthode non-supervisé va partitionner et classer les données dans des groupes homogènes.

II.3.3 Différence entre l'apprentissage supervisé et non supervisé

La différence distincte entre l'apprentissage supervisé à l'apprentissage non-supervisé est le fait que l'apprentissage non-supervisé cherche à trouver des partitions de modèles par lui-même, l'extraction des données est donc descriptive. Par contre, L'apprentissage supervisé est utilisé quand l'utilisateur sait étiqueter les informations, l'extraction des données est donc prédictive.

Tableau II.2. Comparaison entre l'apprentissage supervisé et non supervisé [42]

Supervisé	Non supervisé
- Utiliser pour Classier / prédire des données futures	- Utiliser pour comprendre et explorer les données
- Nombre de classes connues	- Nombre de classes inconnues
- Sur entraînement	- Pas de connaissance préalable

II.4 Performance de réseau de neurone artificiel

Pour évaluer la performance d'un réseau de neurones au cours de l'apprentissage, on utilise des fonctions d'erreurs qui nous indiquent dans quelle mesure les prévisions du réseau sont proches des valeurs cible et donc, quel ajustement doit être apporté aux poids par l'algorithme d'apprentissage à chaque itération.

Toutes les fonctions d'erreur utilisées pour l'apprentissage des réseaux de neurones doivent intégrer une certaine mesure des distances entre les valeurs cibles et les prévisions correspondant aux entrées, ci-dessous des approches courantes consistent à évaluer la performance du RNA utilisant des fonctions d'erreurs comme :

II.4.1 Somme des carrés des erreurs (SSE)

Cette fonction mesure la répartition des valeurs obtenues de la prévision par rapport aux valeurs cibles. Son expression est donnée par :

$$SSE = \sum_{i=1}^N (x_i - \hat{x}_i)^2 \quad (\text{II.32})$$

Avec :

N : le nombre de point dans l'échantillon.

$\{x_i\}$ C'est la valeur désirer.

$\{\hat{x}_i\}$ C'est la valeur prédite.

Une valeur proche de zéro indique une faible valeur de l'erreur répartie de manière aléatoire, commise par l'interpolation, ce qui rend le modèle utile pour la prédiction de valeurs autres que celles obtenues.

II.4.2 Erreur quadratique moyenne (MSE)

L'erreur quadratique moyenne est une valeur unique fournissant des informations sur la qualité de l'ajustement des poids, définit par :

$$MSE = \frac{SSE}{N} \quad (\text{II.33})$$

SSE : C'est la somme des carrés des erreurs

N : le nombre de point dans l'échantillon.

Une erreur quadratique moyenne faible indique donc que la séquence d'erreurs est caractérisée, à la fois par une faible variance et une erreur moyenne réduite.

II.4.3 Ecart quadratique moyen (RMSE)

RMSE est une mesure de la précision permettant de comparer les erreurs de prévision de différents modèles pour un jeu de données, elle est toujours non négative, et une valeur de 0 indiquerait un ajustement parfait aux données. Défini par :

$$RMSE = \sqrt{MSE} \quad (II.34)$$

Avec :

MSE : c'est l'Erreur quadratique moyenne.

II.5 Principaux éléments à retenir dans des réseaux de neurones

- Un réseau de neurones est un système composé de neurones, généralement répartis en plusieurs couches connectées entre elles
- Un tel système s'utilise pour résoudre divers problèmes statistiques. Mais nous nous intéressons ici qu'au problème de prédiction ou de classification (très courant) Dans ce cas, le réseau calcule à partir de l'entrée un score (ou probabilité) pour chaque classe. La classe attribuée à l'objet en entrée correspond à celle de score le plus élevé.
- Chaque couche reçoit en entrée des données et les renvoie transformées. Pour cela, elle calcule une combinaison linéaire puis applique éventuellement une fonction non-linéaire, appelée fonction d'activation. Les coefficients de la combinaison linéaire définissent les paramètres (ou poids) de la couche.
- Un réseau de neurones est construit en empilant les couches : la sortie d'une couche correspond à l'entrée de la suivante.
- Cet empilement de couches définit la sortie finale du réseau comme le résultat d'une fonction différentiable de l'entrée
- La dernière couche calcule les probabilités finales en utilisant pour fonction d'activation.
- Une fonction d'erreur est associée à la couche finale pour calculer l'erreur de la prédiction ou de classification. Il s'agit en général de (MSE, SSE et RMSE).
- Les valeurs des poids des couches sont apprises par rétropropagation du gradient : on calcule progressivement (pour chaque couche, en partant de la fin du réseau) les paramètres qui minimisent la fonction de perte régularisée. L'optimisation se fait avec une descente du gradient.

II.6 Réseau neuronal convolutif

Les Réseau neuronal convolutif (RNC) appelé aussi Alexnet a été développé par des chercheurs de Toronto en 2012, mais est largement inspiré du réseau de neurones convolutif LeNet, développé en 1998 par le chercheur français Yann LeCun. RNC est particulièrement utile pour identifier des modèles dans les images afin de reconnaître des objets, des visages et des scènes. Ils apprennent directement à partir de données d'images et utilisent des modèles pour classer ces dernières. Cette idée a été développée par une expérience fascinante menée par Hubel et Wiesel en 1962, dans laquelle ils ont montré que certaines cellules neuronales individuelles du cerveau ne réagissaient (ou n'étaient déclenchées) qu'en présence de bords d'une certaine orientation, par exemple, certains neurones sont activés lorsqu'ils sont exposés à

des bords verticaux et d'autres lorsqu'ils sont montrés horizontaux ou diagonaux. Hubel et Wiesel ont découvert que tous ces neurones étaient organisés dans une architecture en colonnes et qu'ensemble, ils étaient capables de produire une perception visuelle.

II.6.1 Différence entre un réseau de neurones et un réseau de neurones convolutif

Les réseaux de neurones convolutifs désignent une sous-catégorie de réseaux de neurones : ils présentent toutes les caractéristiques listées dans la partie II.5. Cependant, les RNC sont spécialement conçus pour traiter des images en entrée. Leur architecture est alors plus spécifique : elle est composée de deux blocs principaux.

Le premier bloc fait la particularité de ce type de réseaux de neurones, peut avoir plusieurs dizaines, voire plusieurs centaines de couches, qui apprennent chacune à identifier différentes caractéristiques d'une image. Des filtres sont appliqués à chaque image utilisée pour l'apprentissage à différentes résolutions, et la sortie de chaque image convoluée est utilisée comme entrée de la couche suivante. Les premiers filtres peuvent être des caractéristiques très simples, comme par exemple la luminosité ou les bords, puis passer à des caractéristiques plus complexes qui définissent l'objet de façon unique.

Le second bloc n'est pas caractéristique d'un RNC : il s'agit d'une couche appelé la couche « entièrement connectée » (FC), qui est une couche de type perceptron multicouche, utilisé pour la prédiction ou la classification.

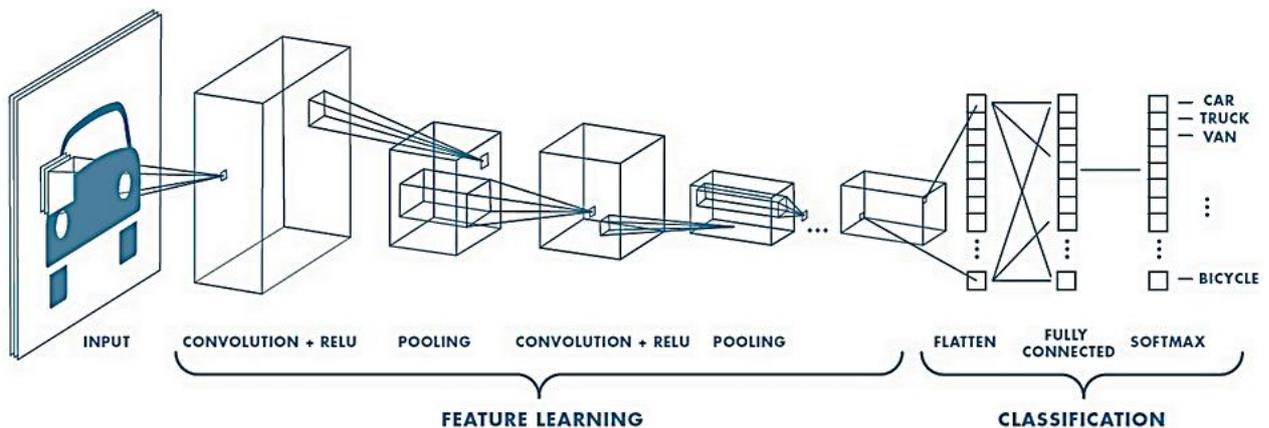


Figure II.16. Réseau de neurones avec plusieurs couches convolutives [43]

II.6.2 Différents types de couches d'un RNC

II.6.2.1 Couche de convolution

La couche de convolution fait passer les images d'entrée par un ensemble de filtres convolutifs (kernels), chacun de ces filtres activant certaines caractéristiques des images.

Considérons une matrice d'entrée dont les pixels d'image sont 0,1, et une matrice de filtrage 3x3, comme indiqué ci-dessous :

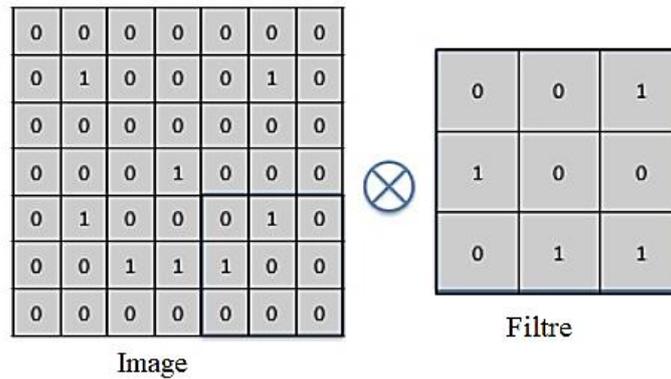


Figure II.17. Convolution de matrice d’image et matrice de filtre [43]

Ensuite, la convolution de la matrice d’images avec la matrice de filtres 3x3, est une sortie appelée « Feature Map », comme indiqué ci-dessous :

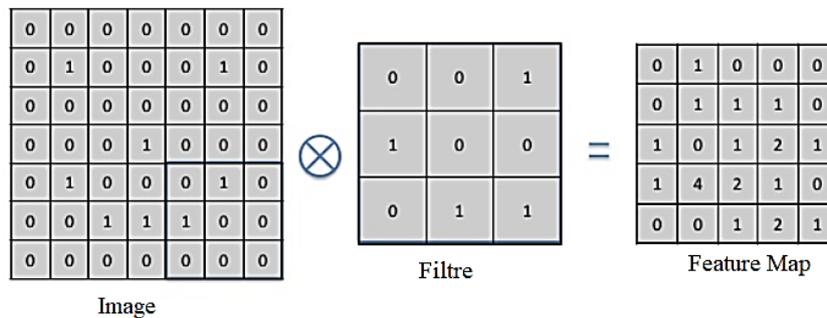


Figure II.18. Feature Map [43]

La convolution d'une image avec différents filtres peut effectuer des opérations telles que la détection des contours, le flou et la netteté en appliquant des filtres. L'exemple ci-dessous montre diverses images de convolution après l'application de différents types de filtres.

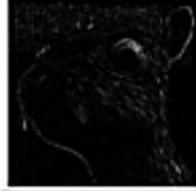
Operation	Filtre	Convolved image
Détection des contours	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
La netteté	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Le flou	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

Figure II.19. Quelques filtres les plus utilisés [44]

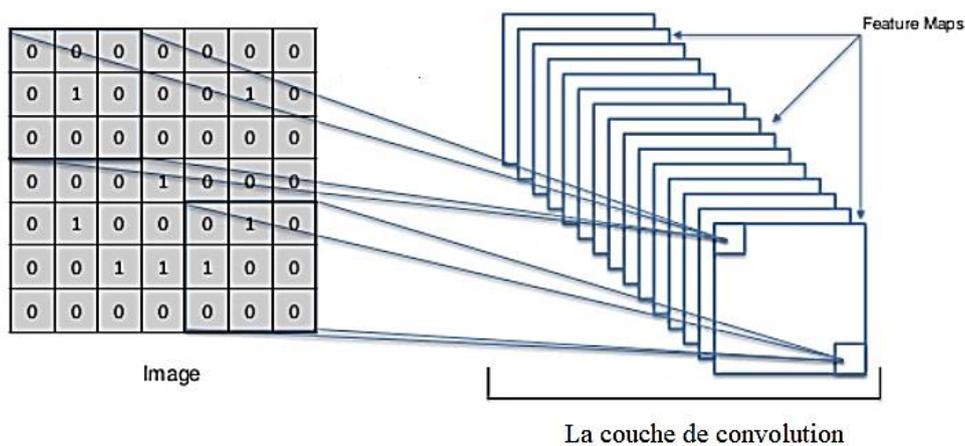


Figure II.20. Création de plusieurs feature maps construit la première couche de convolution [43]

II.6.2.2 Couche ReLU (Rectified linear unit)

Permet d'accélérer et d'optimiser l'apprentissage en mappant les valeurs négatives à zéro et en conservant les valeurs positives. Elle est parfois appelée *couche d'activation*, car seules les caractéristiques activées sont transférées dans la couche suivante. La sortie est :

$$f(x) = \max(0, x) \quad (\text{II.35})$$

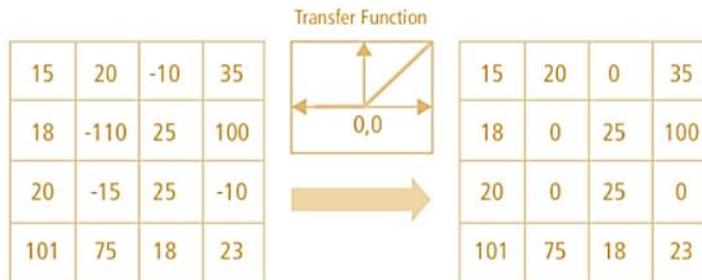


Figure II.21. Opération ReLu [45]

Exemple :



Figure II.22. Image d'entrée [44]



Figure II.23. Image convoluée : noir (valeurs négatives) et blanc (valeurs positives) [44]



Figure II.24. Valeur positive seulement [44]

Cette opération est appliquée sur toutes les feature maps ce qui donne :

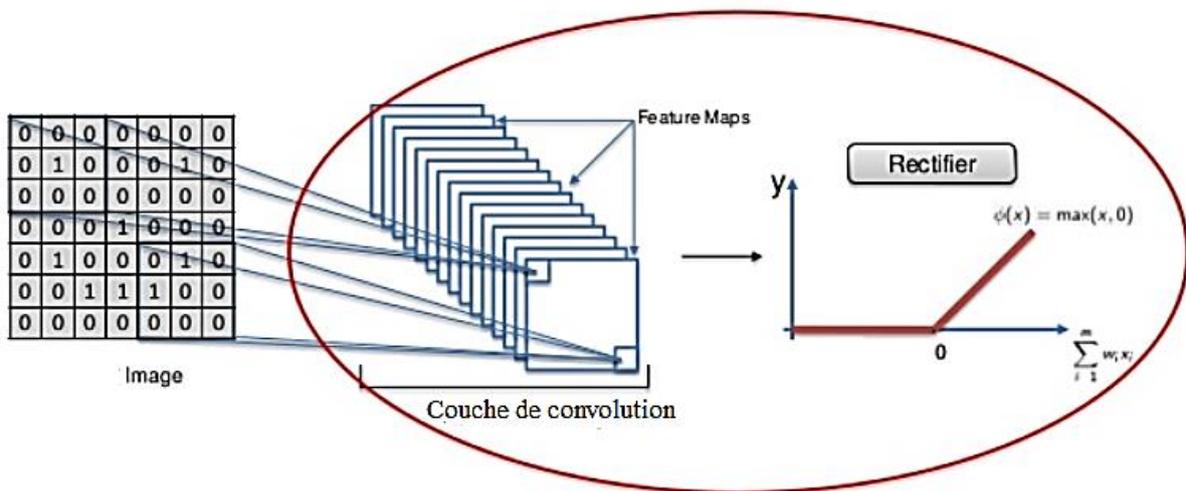


Figure II.25. Traitement de toutes les feature maps par la fonction ReLu [43]

II.6.2.3 Couche de pooling

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs *feature maps*, et applique à chacune d'entre elles l'opération de **pooling**.

L'opération de *pooling* consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes, il existe plusieurs opérations qui sont :

- Max Pooling : prend le plus gros élément de la feature map rectifiée.
- Average Pooling : prend la valeur moyenne de la feature map rectifiée.
- Sum Pooling : prend la somme de tous les éléments de la feature map rectifiée.

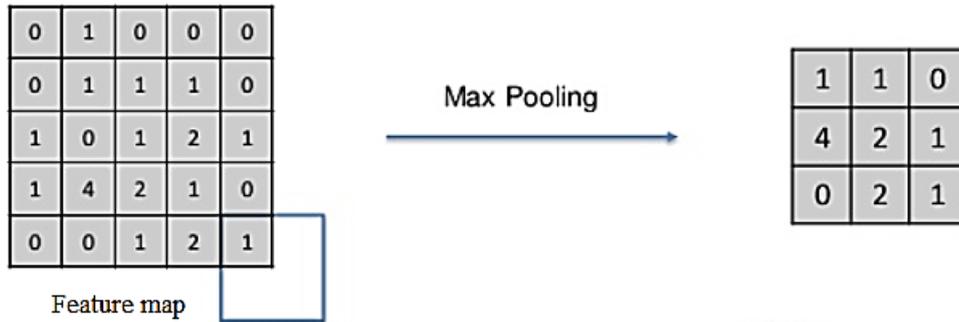


Figure II.26. Pooled feature map [43]

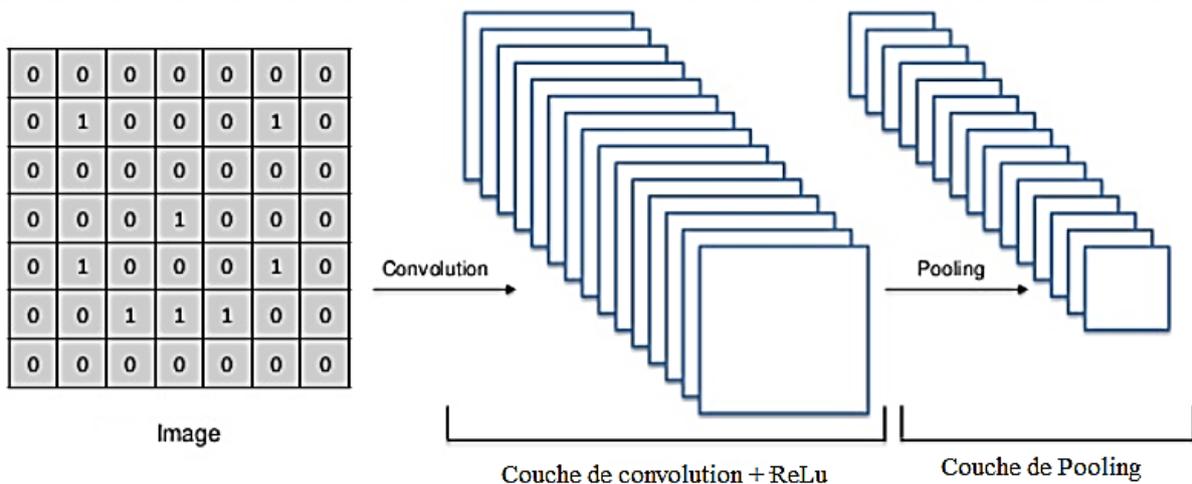


Figure II.27. Pooling plusieurs carte construit la couche de pooling [43]

II.6.2.4 Couche entièrement connectée FC

Dans la couche FC, la matrice appelée pooled feature map sera transformer en vecteur de taille N , où N est le nombre de classes dans le problème de classification d'images, chaque élément du vecteur indique la probabilité pour l'image en entrée d'appartenir à une classe. Le vecteur de dimension N sera l'entrée de la couche entièrement connectée qui représente un perceptron multicouche.

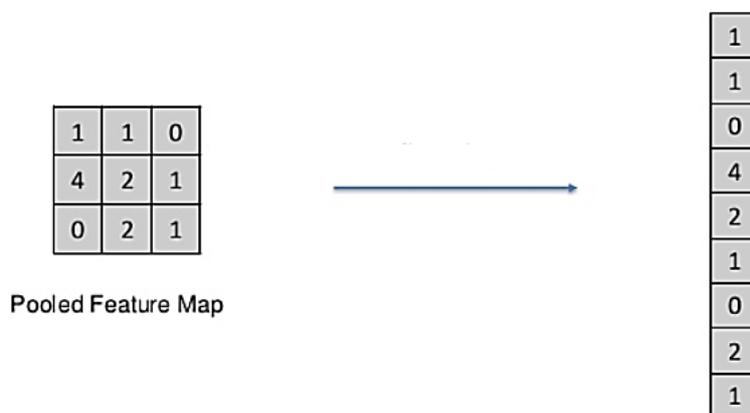


Figure II.28. Redimensionnement de la matrice en vecteur [43]

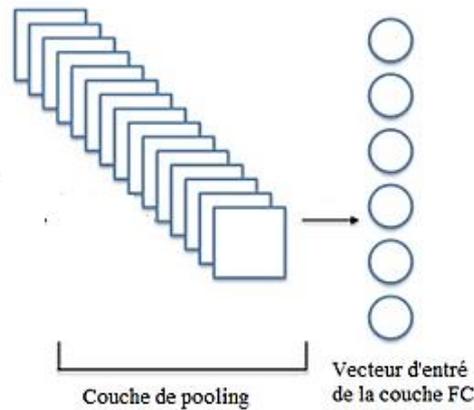


Figure II.28. Redimensionnement de toutes les cartes en vecteurs [43]

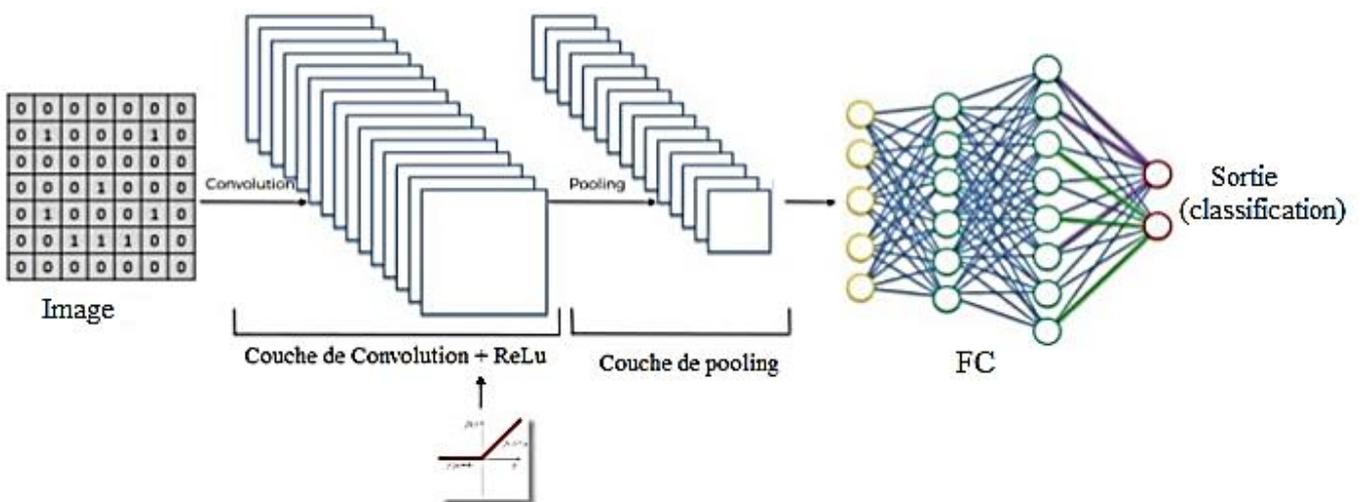


Figure II.29. Architecture d'un RNC [43]

II.7 Conclusion

Le présent chapitre a été consacré à la présentation de la théorie des réseaux de neurones, en partant du neurone biologique, jusqu'à l'étude des différents types de réseaux, le principe d'apprentissage et l'algorithme de rétropropagation du gradient.

Les réseaux de neurones multicouches entraînés par l'algorithme de rétropropagation du gradient sont aujourd'hui les modèles les plus utilisés en diagnostic et en surveillance en raison de leur capacité à résoudre les problèmes de prédiction et de classification.

CHAPITRE III
EXTRACTION DES
CARACTERISTIQUES
DU COURANT DE
FUITE

Introduction

Les caractéristiques extraites à partir des signaux du courant de fuite sont riches en informations. De ce fait, ces dernières peuvent être utilisées pour diagnostiquer et analyser les isolateurs pollués afin de minimiser les pannes causées par la dégradation de leur performance d'isolation [21].

Les articles, cités à travers le premier chapitre, résument l'état de l'art de la littérature donnent des indications pour le choix d'entrée, ou caractéristiques, pertinentes du RNA. Par exemple :

Yan Mi, et al [37] ont utilisé des caractéristiques temporelles, à savoir, l'amplitude, la valeur efficace du courant de fuite, le nombre d'impulsions et le paramètre d'environnement de l'humidité relative dans la prédiction du niveau de pollution de l'isolateur.

C. Volat, et al [38] ont proposé des indicateurs fréquentiels, tels que : le déphasage entre le fondamental du courant de fuite et la tension appliquée, amplitude du troisième et cinquième harmonique du LC pour la prédiction de la présence des décharges électriques dans le courant de fuite.

Dans notre travail, nous avons comparé les caractéristiques temporelles et fréquentielles du courant de fuite comme entrée de RNA. Durant ce chapitre, nous allons, d'abord, expliquer théoriquement la signification des indicateurs, tant que temporels et fréquentiels. Puis, nous allons mettre en avant l'efficacité de ces caractéristiques pour décrire avec pertinence la sévérité de pollution à la surface des isolateurs. Autrement dit, ce chapitre vise à prouver la corrélation entre les entrées (temporelles ou fréquentielles) du RNA avec la sortie de ce dernier (valeur de la conductivité de la solution polluante).

III.1 Méthode d'acquisitions du courant de fuite

Durant notre thèse, nous avons étudié le comportement du modèle plan de l'isolateur 1512 L sous une pollution uniforme. Les essais pratiques ont été effectués au laboratoire de Haute Tension de l'Ecole Nationale Polytechnique (l'ENP) [39].

Ces essais ont été effectués pour la pollution uniforme, ils nous ont permis de décrire l'évolution du courant de fuite à la surface du modèle plan pour différentes conductivités et différents niveaux de tension appliqués.

III.1.1 Dispositif expérimental

La station d'essai du laboratoire contient les équipements suivants :

- Un transformateur d'essai : 500V/300kV, 50 Hz
- Un transformateur de réglage : 220/0 à 500 V, 50 Hz.
- Un diviseur de tension capacitif qui est relié au secondaire du transformateur d'essai pour réduire la tension aux bornes du voltmètre, situé près du pupitre de commande. Il est composé d'une capacité $C1=400 \mu\text{F}$ et d'une capacité variable qui permet de contrôler la tension de sortie.
- Des appareils de mesures et de protection.
- Un oscilloscope numérique « TEKTRONIX » de bande passante 500 MHz.
- Une caméra numérique SONY DCR-SR45 pour suivre l'évolution des décharges parallèles en fonction de la tension appliquée et de l'état de surface du modèle.

- Un ordinateur pour l'acquisition des données de la tension appliquée et du courant de fuite qui traverse la surface du modèle.

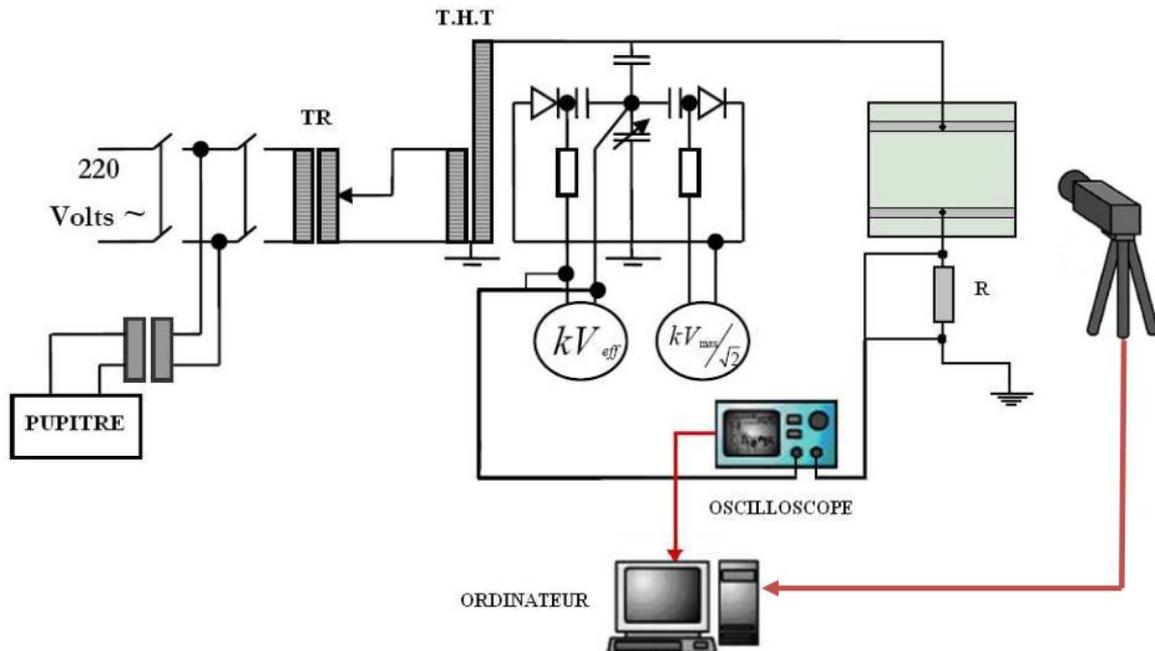


Figure III.1. Dispositif expérimentale [39]

III.1.2 Description du modèle d'isolateur étudié

Les essais ont été effectués sur un modèle plan simulant l'isolateur 1512 L. Deux électrodes sont posées sur deux bandes conductrices en aluminium qui sont découpées (500 x 30 mm²) de manière à ne pas avoir de pointes à leurs extrémités. A l'aide du gazoil, celles-ci sont parfaitement collées contre la paroi du modèle plan pour éviter les couches d'air qui ont tendance à se former sous ces bandes, pouvant engendrer l'apparition de micro-décharges qui risquent de nuire à la qualité des mesures de la tension de contournement et du courant de fuite pendant les expériences. Le modèle est élevé à une hauteur de 175 cm afin de limiter l'effet des capacités parasites qui peuvent altérer la mesure de la tension appliquée et du courant de fuite.

Le nettoyage du plan est tout d'abord fait avec de l'eau du robinet ; puis il est séché avec des serviettes en papier pour éliminer le maximum d'impuretés déposées. Un deuxième nettoyage est effectué à l'aide d'un bout de coton imbibé d'alcool chirurgical à 700 afin d'assurer une propreté quasi-totale du modèle plan. Cette méthode de nettoyage est adoptée pour le cas de la pollution uniforme et la pollution non-uniforme.

L'isolateur réel 1512 L ainsi que son modèle étudié au laboratoire de haute tension sont montrés aux figures III.2 et III.3

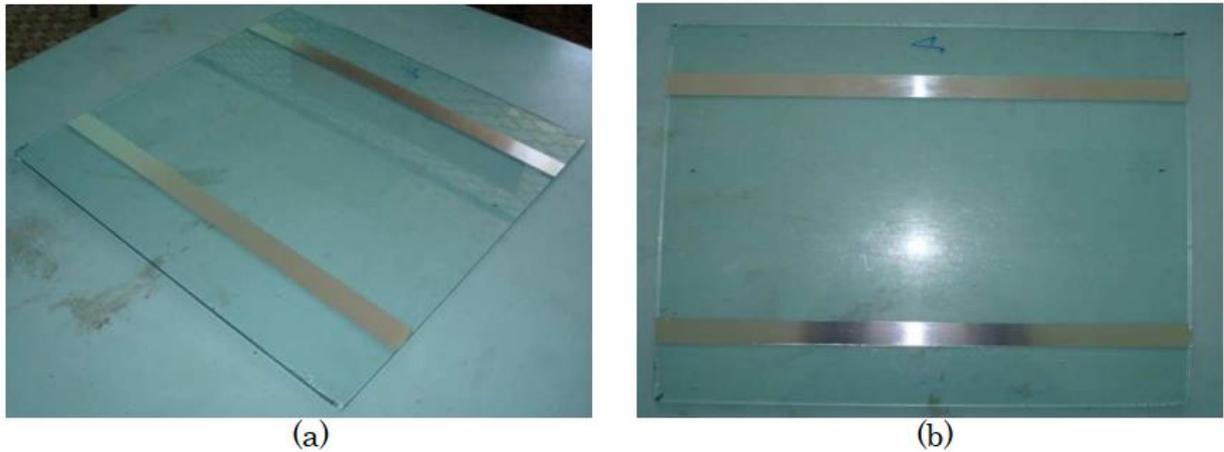


Figure III.2 : Modèle plan étudié : (a) vue latérale, (b) vue de haut [39]

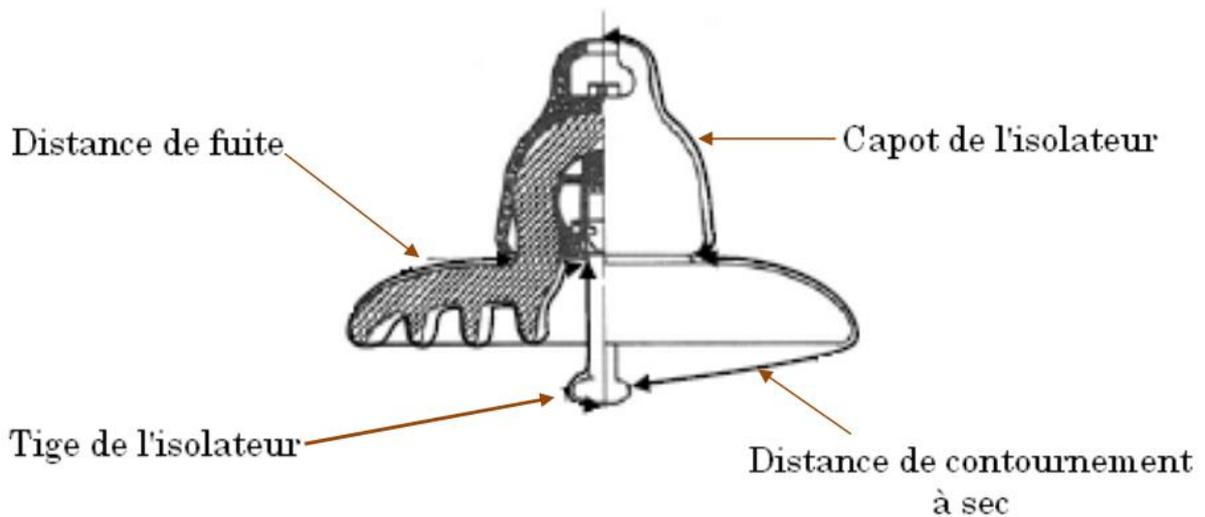


Figure III.3 : Profil de l'isolateur réel (1512 L) [39]

Les principales dimensions du modèle plan étudié et de l'isolateur réel 1512 L sont données dans les tableaux III.1 et III.2 après avoir mesuré les différentes grandeurs à l'aide d'un mètre ruban.

Tableau III.1 : Dimensions du modèle plan étudié

Grandeur	Longueur (mm)
Longueur de la plaque	500
Largeur de la plaque	500
Epaisseur de la plaque	5
Distance inter-électrode	292

Tableau III.2 : Dimensions de l'isolateur réel 1512 L

Grandeur	Longueur (mm)
Capot de l'isolateur	295
Distance de contournement à sec	220
Ligne de fuite	292
Profile de la tige	128

III.1.3 Essais expérimentaux

III.1.3.1 Préparation des solutions polluantes

Les solutions polluantes ont été préparées tantôt au laboratoire du Génie de l'Environnement tantôt au laboratoire du Génie Chimique. Celles-ci sont composées de différentes quantités de NaCl mélangées à de l'eau distillée, afin d'obtenir les conductivités volumiques suivantes : 0.01 mS/cm, 0.19 mS/cm, 0.71 mS/cm, 1.2 mS/cm et 10.1 mS/cm. Le choix de ces conductivités est fait pour avoir une bonne visibilité concernant l'élongation des décharges parallèles et aussi pour une bonne étude du courant de fuite pour le cas de la pollution uniforme.

III.1.3.2 Méthode de pulvérisation

La méthode de pulvérisation a été choisie pour la pollution artificielle de la surface du modèle plan pour le cas uniforme et le cas non-uniforme. La longueur de fuite $L_f = 29.2$ cm du modèle est gardé constante pour la pollution uniforme.

Le plan est humidifié avec une conductivité donnée en gardant une ouverture constante du pulvérisateur. L'humidification de la surface est obtenue après cinq pulvérisations de chaque côté de l'isolateur (figure III.4) à une distance de 0.5 m qui est gardée constante durant toute l'opération.

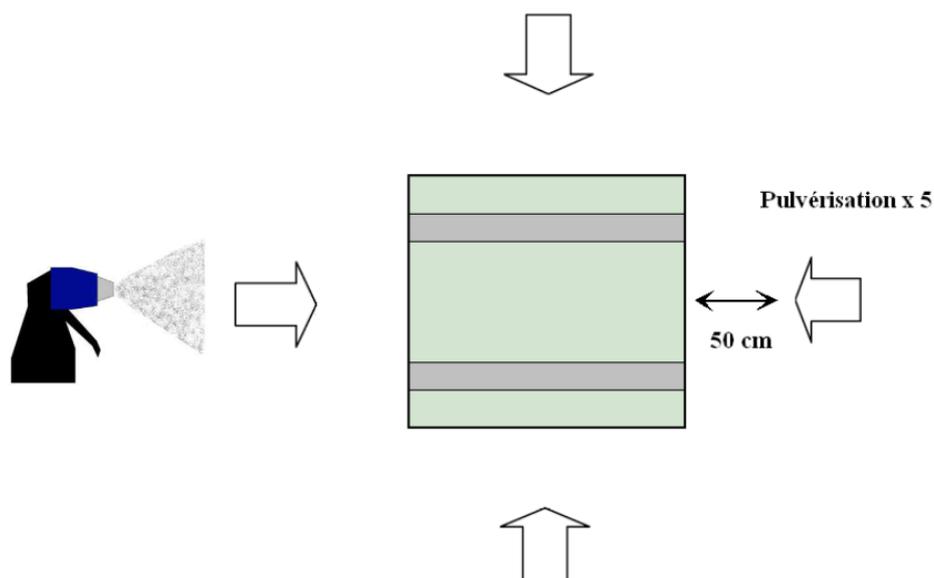


Figure III.4 : Méthode de pulvérisation pour la pollution uniforme [39].

III.1.4 Exemple du courant de fuite enregistré

La mesure du courant de fuite a été faite grâce à l'oscilloscope TEKTRONIX par l'intermédiaire du signal de la tension avec une résistance $R = 1k\Omega$, afin de limiter l'effet des parasites qui altèrent la qualité du signal du courant de fuite.

La résistance est insérée en série avec l'objet d'essai et le retour de terre. Un câble coaxial est utilisé pour récupérer les données du courant de fuite et de la tension appliquée envoyés vers l'oscilloscope. La fréquence d'échantillonnage est de 100 kHz. Ce choix permet d'avoir une vision plus large des ondes de tension et du courant de fuite [39].

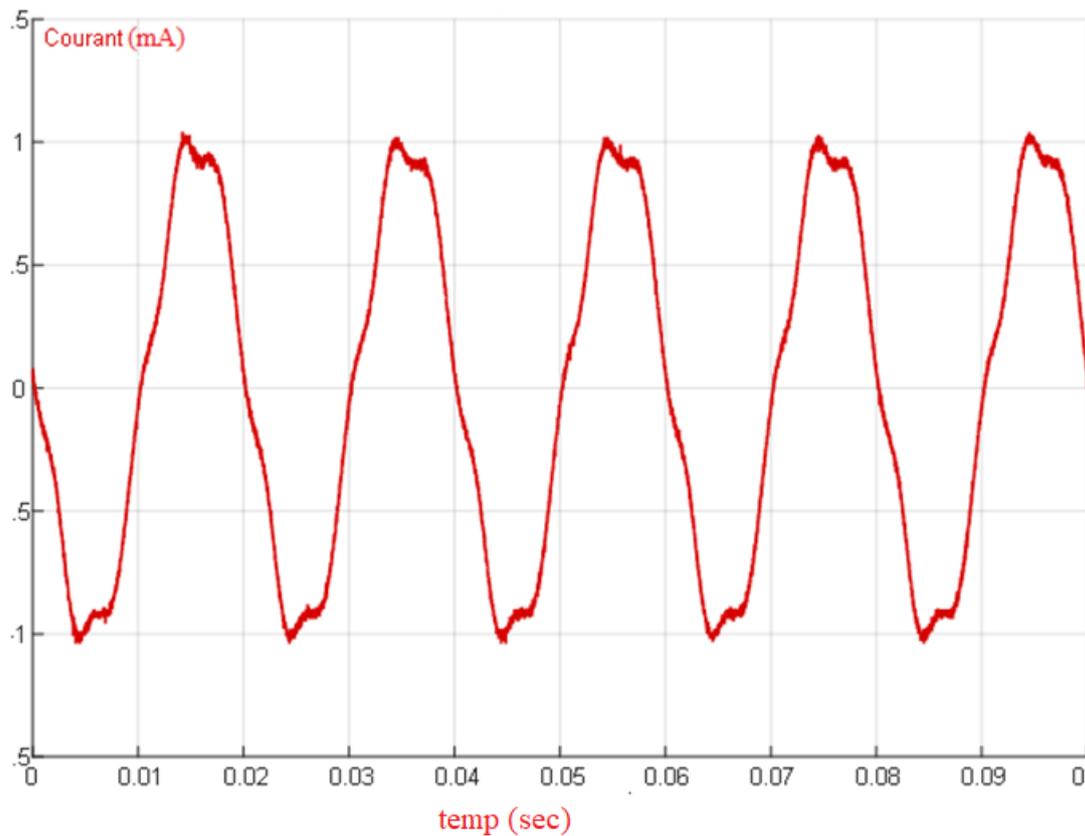


Figure III.5. Signal du courant de fuite pour une conductivité de $10 \mu\text{S}/\text{cm}$ et une tension appliquée de 12kV_{eff}

III.2 Caractéristiques temporelles du courant de fuite

Afin de caractériser l'évolution du courant de fuite, plusieurs tests ont été réalisés avec différentes valeurs de la conductivité et de tensions appliquées pour l'isolateur. Ainsi, ces tests permettront d'identifier l'évolution des paramètres (I_{max} , RMS, STD) en fonction de la conductivité. Une fois que la corrélation entre ces paramètres avec la sévérité de pollution est prouvée, ces derniers sont utilisés comme des entrées de RNA, tel que nous allons présenter dans le Chapitre 4, dans ce qui suit, on va justifier notre choix des caractéristiques temporelles du courant de fuite.

III.2.1 Valeur maximale

III.2.1.1 Mesure de l'amplitude maximale

La mesure de l'amplitude maximale du courant de fuite est déterminée à l'aide de Programme MATLAB® qui permet de calculer la valeur maximale d'un signal d'entrée.

III.2.1.2 Etude de l'amplitude maximale en fonction de la conductivité

1. Conductivité 0.01, 0.19, 0.71 et 1.2 mS/cm

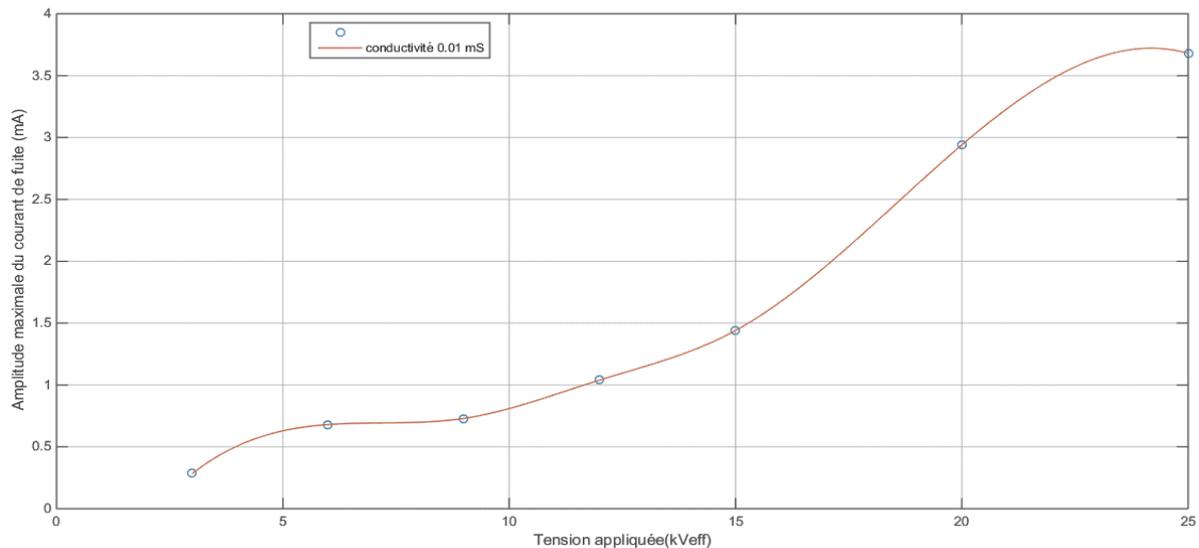


Figure III.6. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivité de 0.01mS/cm.

Pour la conductivité 10 μ S/cm, le courant de fuite augmente de manière progressive car il n'y a pas d'assèchement important.

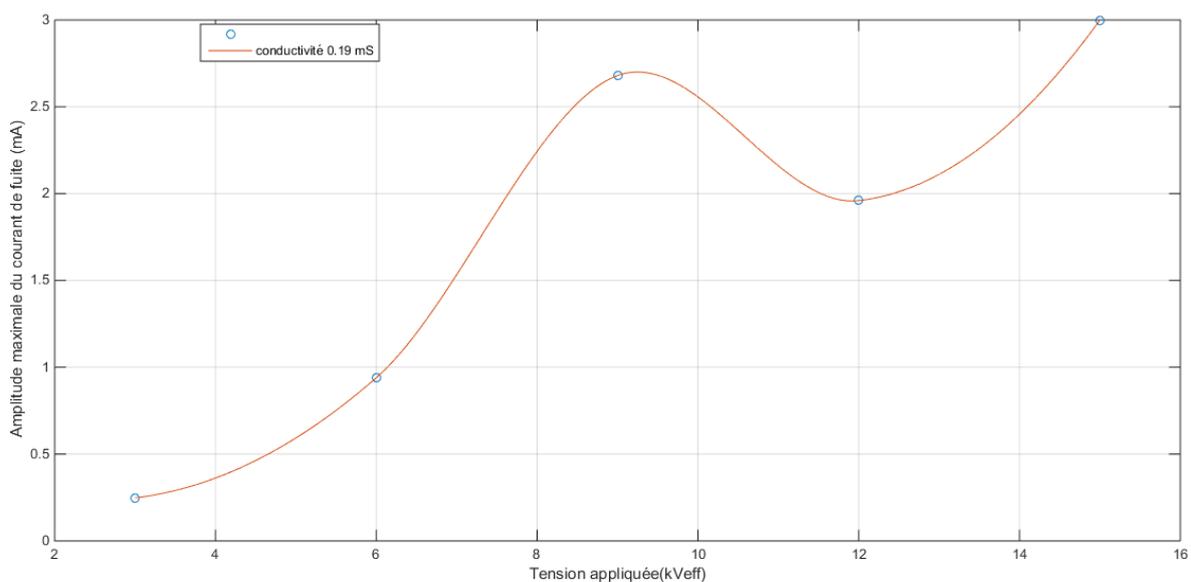


Figure III.7. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivité de 0.19mS/cm.

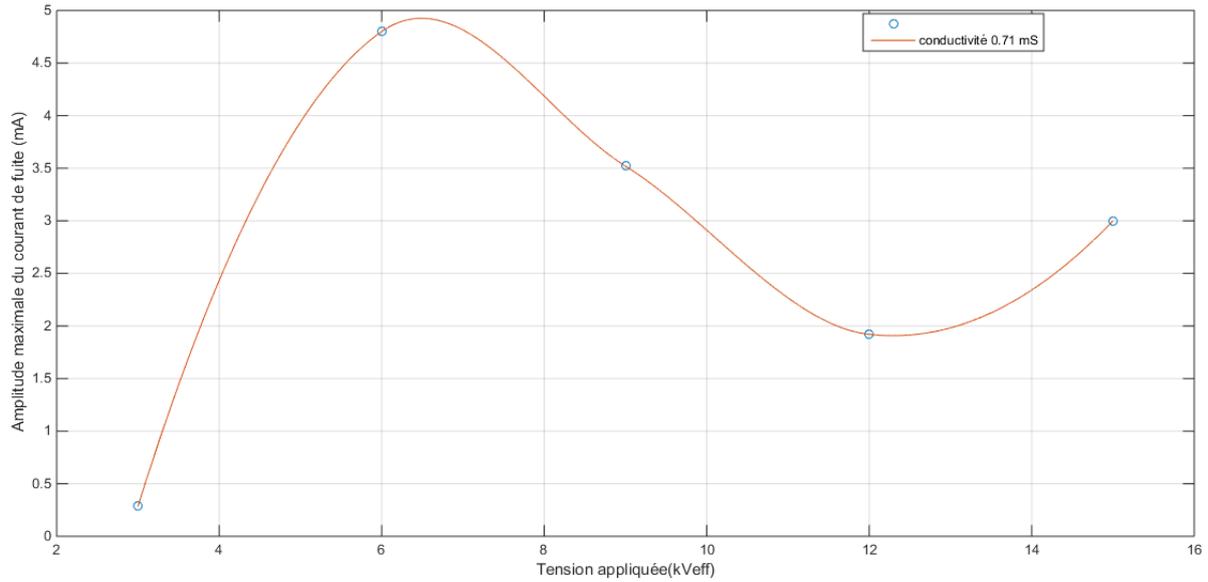


Figure III.8. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivité de 0.71 mS/cm.

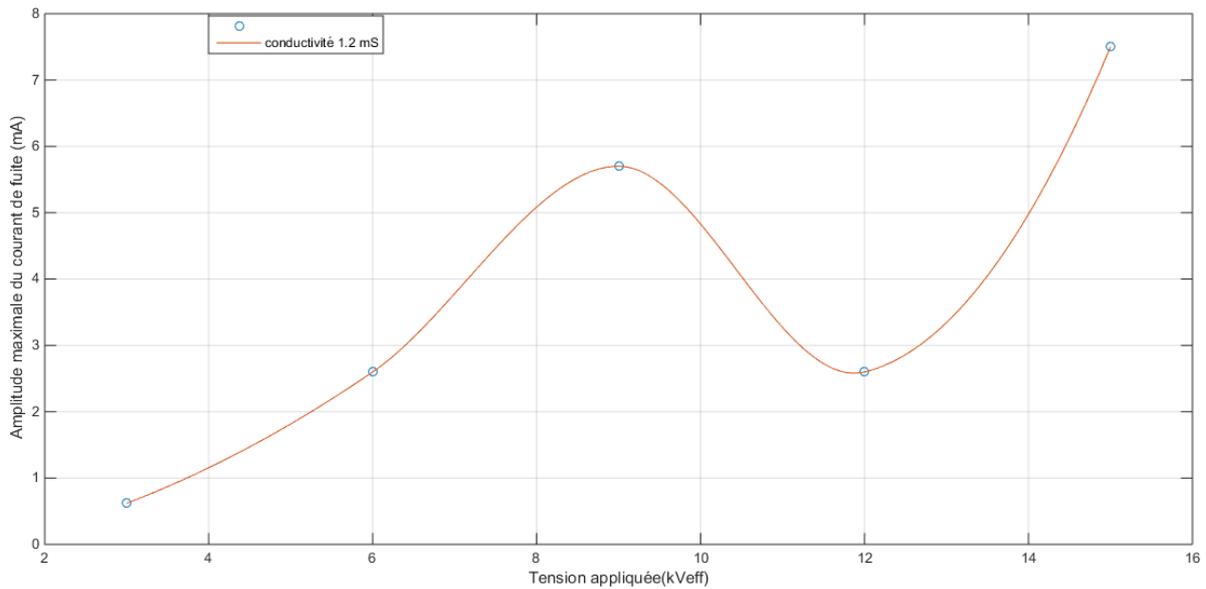


Figure III.9. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour une conductivité de 1.2 mS/cm.

Pour les conductivités 0.19, 0.71 et 1.2 mS/cm, on remarque une baisse de l'amplitude maximale du courant de fuite à partir de la tension 9 kV_{eff}. Cette baisse est due aux assèchements de la surface isolante (signe d'activité de décharges très développée).

2. Conductivité 10.1 mS/cm

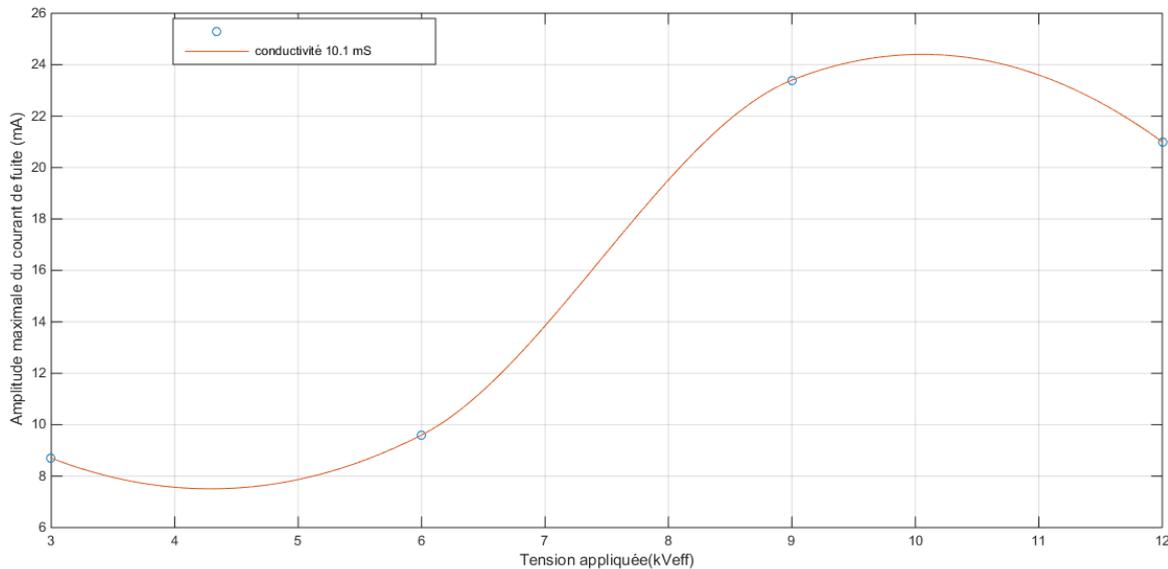


Figure III.10. Variation de l'amplitude maximale du signal du courant de fuite en fonction de la tension appliquée pour la conductivité 10.1 mS/cm.

Pour la conductivité 10.1 mS/cm, l'amplitude maximale du courant de fuite est la plus élevée. Elle commence à baisser à partir de 10 kV_{eff} puisque des décharges continuent d'avoir lieu à la surface de l'isolateur.

En effet, l'augmentation de l'amplitude maximale varie pour la faible conductivité de 0.01 mS/cm, les conductivités intermédiaires 0.19, 0.71 et 1.2 mS/cm et la conductivité très élevée de 10.1 mS/cm. Ceci due à l'apparition des décharges parallèles qui augmentent l'amplitude du courant de fuite pour certaines couches conductrices et diminuent celle-ci pour d'autres couches, à cause d'un assèchement partiel de la solution polluante qui se trouvant à la surface du l'isolateur plan.

III.2.2 Valeur moyenne du courant de fuite (RMS)

III.2.2.1 Mesure de la valeur moyenne

La valeur moyenne d'une grandeur $u(t)$ est l'aire sous la courbe sur un intervalle d'une période T et peut être calculée au moyen d'une intégrale définie par :

$$\langle U \rangle = \frac{1}{T} \int_{t_0}^{t_0+T} U(t) dt \quad (\text{III.1})$$

III.2.2.2 Variation de la valeur moyenne en fonction de la conductivité

Pour diverses valeurs de conductivité de pollution et pour une valeur de tension appliquée de 12 kV_{eff}, les valeurs moyenne du courant de fuite sont indiquées sur la figure III.11.

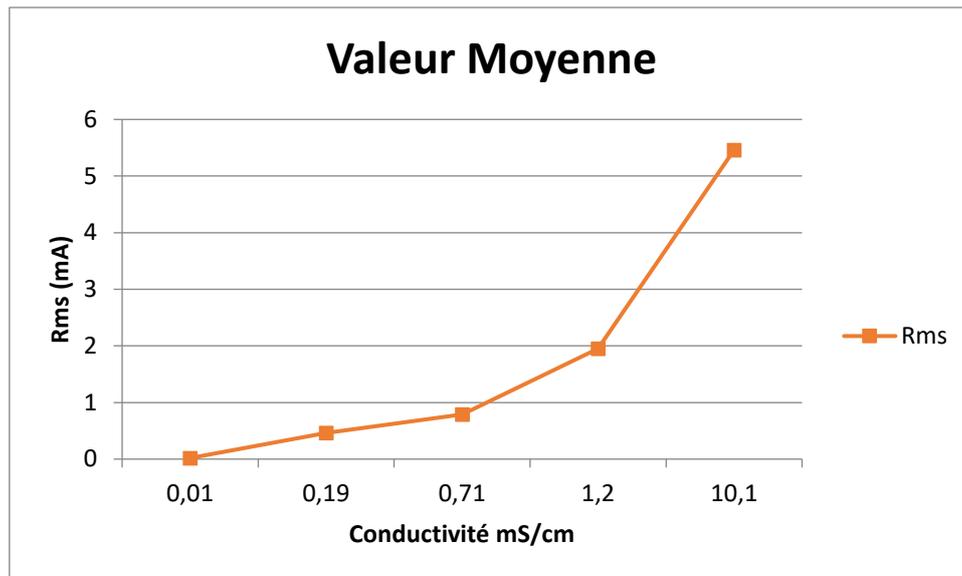


Figure III.11. Valeur moyenne du courant de fuite en fonction de la conductivité

A partir de la figure III.11, nous observons que plus que la conductivité augmente plus la valeur moyenne du courant de fuite augmente, par conséquent il y a une corrélation entre les deux. Ce qui nous permet d'utiliser la valeur moyenne du courant de fuite pour suivre la dégradation des performances de l'isolateur pollué.

III.2.3 Valeur de la déviation standard (STD)

III.2.3.1 Mesure de la déviation standard (STD)

La déviation standard (STD) indique l'ampleur de la variation ou de la dispersion par rapport à la moyenne [24]. Un faible STD indique que les points de données ont tendance à être très proche de la moyenne, tandis qu'un écart-type élevé indique que les points de données sont répartis sur une large plage de valeurs, définie par :

$$s = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (\text{III.2})$$

Où $\{x_1, x_2, \dots, x_N\}$ sont les valeurs mesurées, \bar{x} est la valeur moyenne et N le nombre d'observations dans l'échantillon.

III.2.3.2 Variation de la STD en fonction de la conductivité :

Comme indiqué sur la figure III.12, pour diverses valeurs de conductivité de pollution et pour une valeur de tension appliquée de 12 kVeff, on va calculer les valeurs de la STD du courant de fuite.

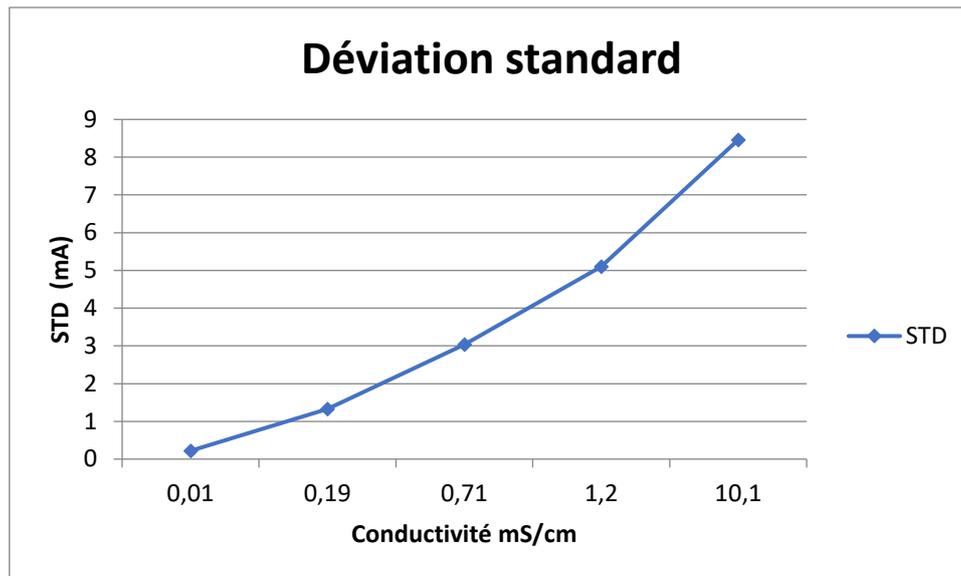


Figure III.12. Valeurs de STD en fonction de la conductivité.

Comme le montre la figure III.12, plus que la conductivité augmente, plus que les valeurs STD augmentent. Cette augmentation des valeurs des coefficients STD s'explique par la présence de décharges électriques sur l'isolant pollué [40].

Par conséquent, si les valeurs STD sont basses, les valeurs LC sont également basses et proches de la valeur moyenne ; il n'y a pas d'activité de décharge sur la surface de l'isolateur, donc ce dernier est en bon état. Cependant, si les valeurs de STD sont élevées ; il y a une forte activité de décharge sur la surface de l'isolateur, son état est dangereux et le contournement est imminent. [40].

III.3 Caractéristiques fréquentielles du courant de fuite

Le signal du courant de fuite présente une forme d'onde qui évolue avec l'état de la surface de l'isolateur, et elle peut être déformée par la présence d'arcs électriques partiels. Une telle déformation de la forme d'onde du courant de fuite est interprétée d'un point de vue fréquentiel par l'apparition d'harmoniques, multiples de la fréquence fondamentale de 50Hz, dont les amplitudes et les déphasages sont principalement dépendantes du degré de déformation de la forme d'onde.

Ainsi, l'analyse de l'évolution des harmoniques du courant de fuite et de leur déphasage (A_m 3 h, A_m 5 h et déphasage LC et V) peut donc permettre, de tirer des informations pertinentes sur l'évolution du courant de fuite en fonction de l'augmentation de la pollution dans le but de pouvoir identifier des signatures significatives indiquant une dégradation de la tenue diélectrique des isolateurs.

III.3.1 Sérier de fourrier

III.3.1.1 Définition

La transformée de Fourier est un outil pour traiter des signaux périodiques, ou des fonctions intégrables. Elle décompose celles-ci en leur spectre de fréquences élémentaires. Les coefficients obtenus sont appelés séries de Fourier.

L'étude des fonctions par cette méthode s'appelle l'analyse harmonique. Une façon de décrire les fonctions périodiques. La fonction est :

- Décomposable en suite de ses coefficients de Fourier (analyse) ; ou
- Reconstituée à l'aide de la suite de ses coefficients (synthèse).

III.3.1.2 Transformation de Fourier rapide (FFT)

La transformation de Fourier est un algorithme de calcul de la transformation de Fourier discrète, Cet algorithme est couramment utilisé en traitement numérique du signal pour transformer des données discrètes du domaine temporel dans le domaine fréquentiel. Son efficacité permet de réaliser des filtrages en modifiant le spectre et en utilisant la transformation inverse.

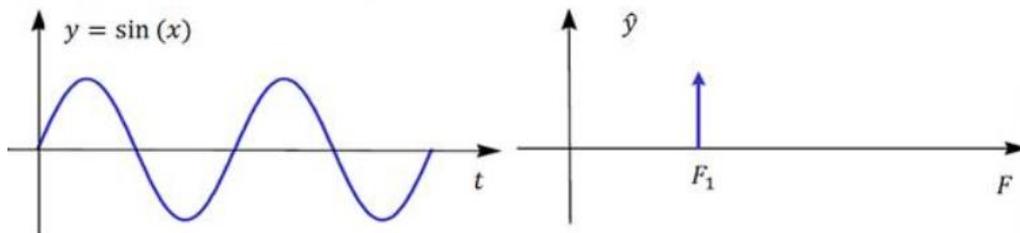


Figure III.13. Spectre d'une sinusoïde.

L'analyse fréquentielle de la fonction usuelle sinusoïdale montre qu'elle ne contient qu'un seul harmonique qui est le fondamentale et tous les autres harmoniques sont nuls.

III.3.2 L'amplitude du 3^{ème} et 5^{ème} harmonique du courant de fuite

La forme d'onde du courant de fuite des isolateur pollués change avec le changement de conductivité, l'analyse fréquentielle du courant de fuite pour les cinq classes de la conductivité et pour une valeur de tension appliquée de 12 kVeff a été réalisée en utilisant la transformée de Fourier. Les résultats sont présentés aux figures III.14 à III.18.

III.3.2.1 Analyse spectrale du courant de fuite en fonction de la conductivité

1. Conductivité $\sigma = 10 \mu\text{S/cm}$

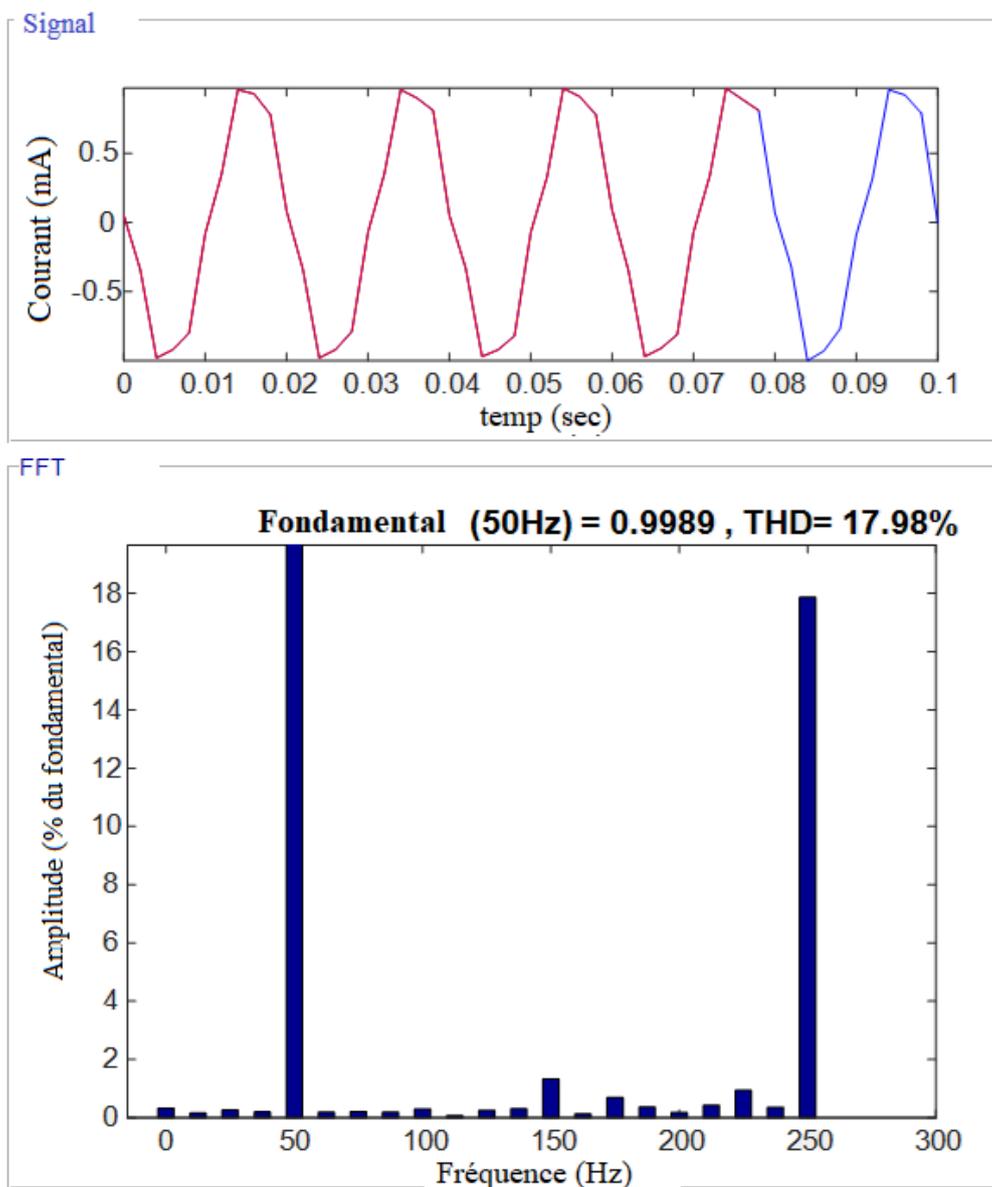


Figure III.14. Signal du courant de fuite et sa représentation fréquentielle (FFT).

Dans ce cas de pollution la conductivité est faible ($10 \mu\text{S/cm}$) alors ceci est caractérisée par un courant à dominante capacitive et sinusoïdale. A partir de la figure III.14 l'analyse fréquentielle illustre clairement la nature du courant au cours de cette période. L'amplitude du courant de fuite pour sa composante fondamentale (fréquence de 50 Hz) est significative devant celle des autres composantes harmoniques. Cela traduit la forme sinusoïdale du courant de fuite.

On peut aussi noter l'apparition d'autres harmoniques de faible amplitude causée principalement par les décharges couronnes. Cependant, l'amplitude du 3^{ème} (150 Hz) harmonique est très faible devant celle du 5^{ème} (250 Hz) harmonique qui est très importante par rapport aux autres.

2. Conductivité $\sigma = 190$, $\sigma = 710 \mu\text{S/cm}$

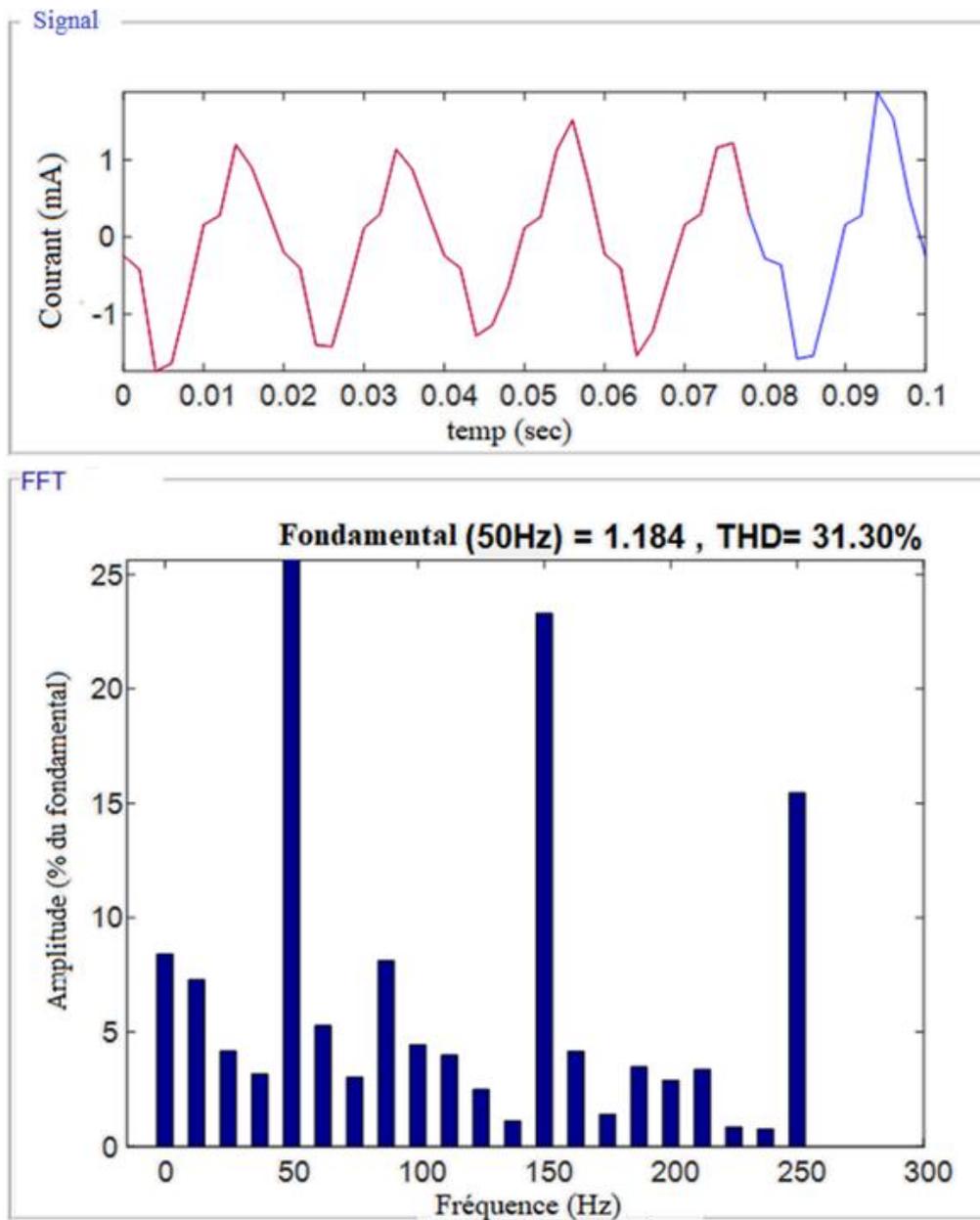


Figure III.15. Signal du courant de fuite et sa représentation fréquentielle (FFT) pour $\sigma = 190 \mu\text{S/cm}$.

On remarque qu'avec l'augmentation de la conductivité, le signal du courant de fuite commence à perdre sa forme sinusoïdale et l'amplitude du 3^{ème} (150 Hz) harmonique devient très significative par rapport à sa valeur dans le cas de conductivité (10 $\mu\text{S/cm}$).

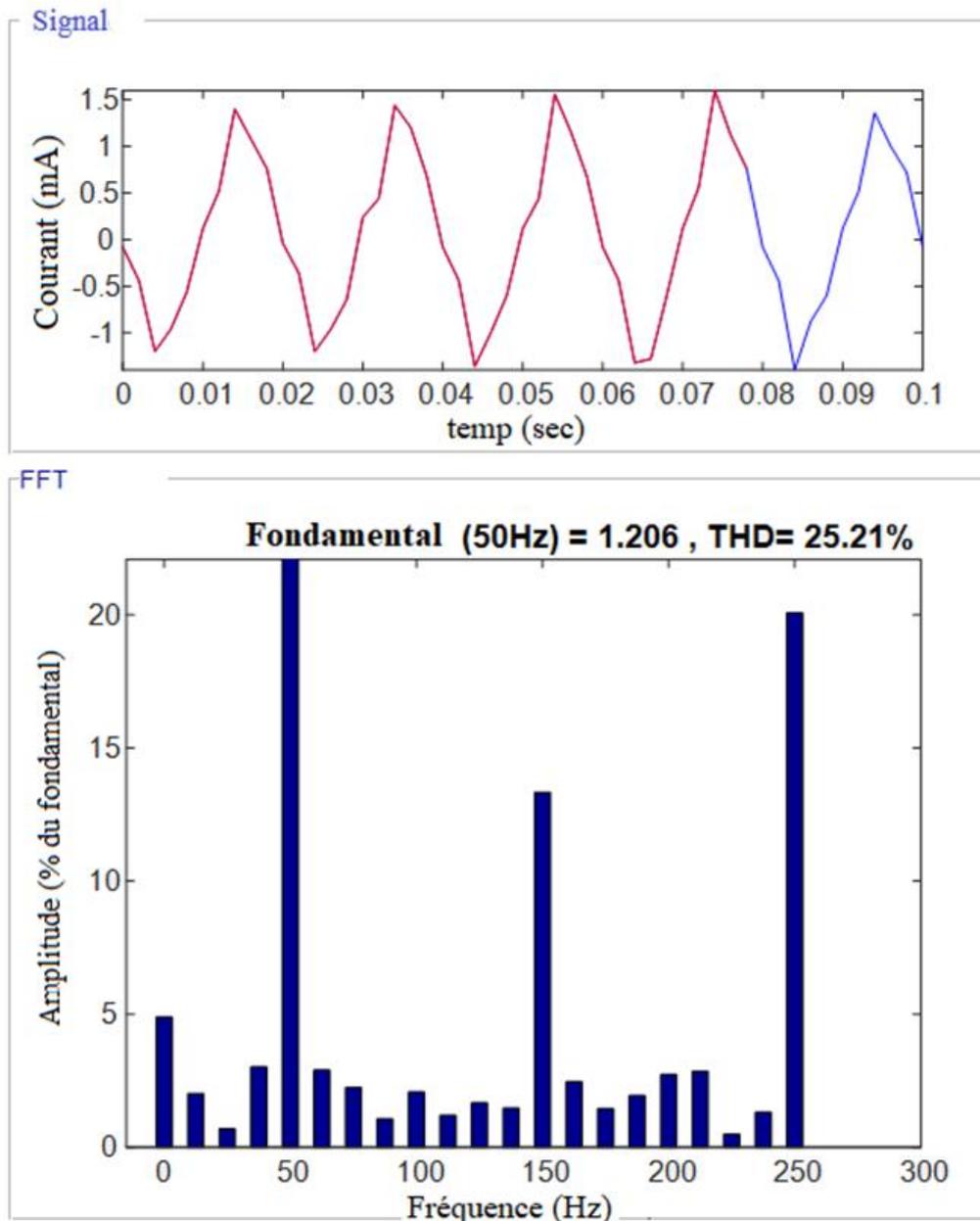


Figure III.16. Signal du courant de fuite et sa représentation fréquentielle (FFT) pour $\sigma = 710 \mu\text{S/cm}$.

Dans ce cas de pollution la conductivité commence à devenir de plus en plus importante ($710 \mu\text{S/cm}$), ce qui fait, des arcs partiels apparaissent de façon intermittente. Ces arcs sont caractérisés par des pics d'intensité de courtes durées, comme le présente la Figure III.15 et III.16, ils entraînent une modification considérable de la forme d'onde du courant de fuite qui en perd sa forme sinusoïdale. Comme le montre l'analyse fréquentielle, la distorsion de la forme du courant de fuite entraîne ainsi l'augmentation de l'amplitude des harmoniques et spécialement des troisième et cinquième harmonique par rapport à celle du fondamental.

3. Conductivité $\sigma = 1,2 \text{ mS/cm}$

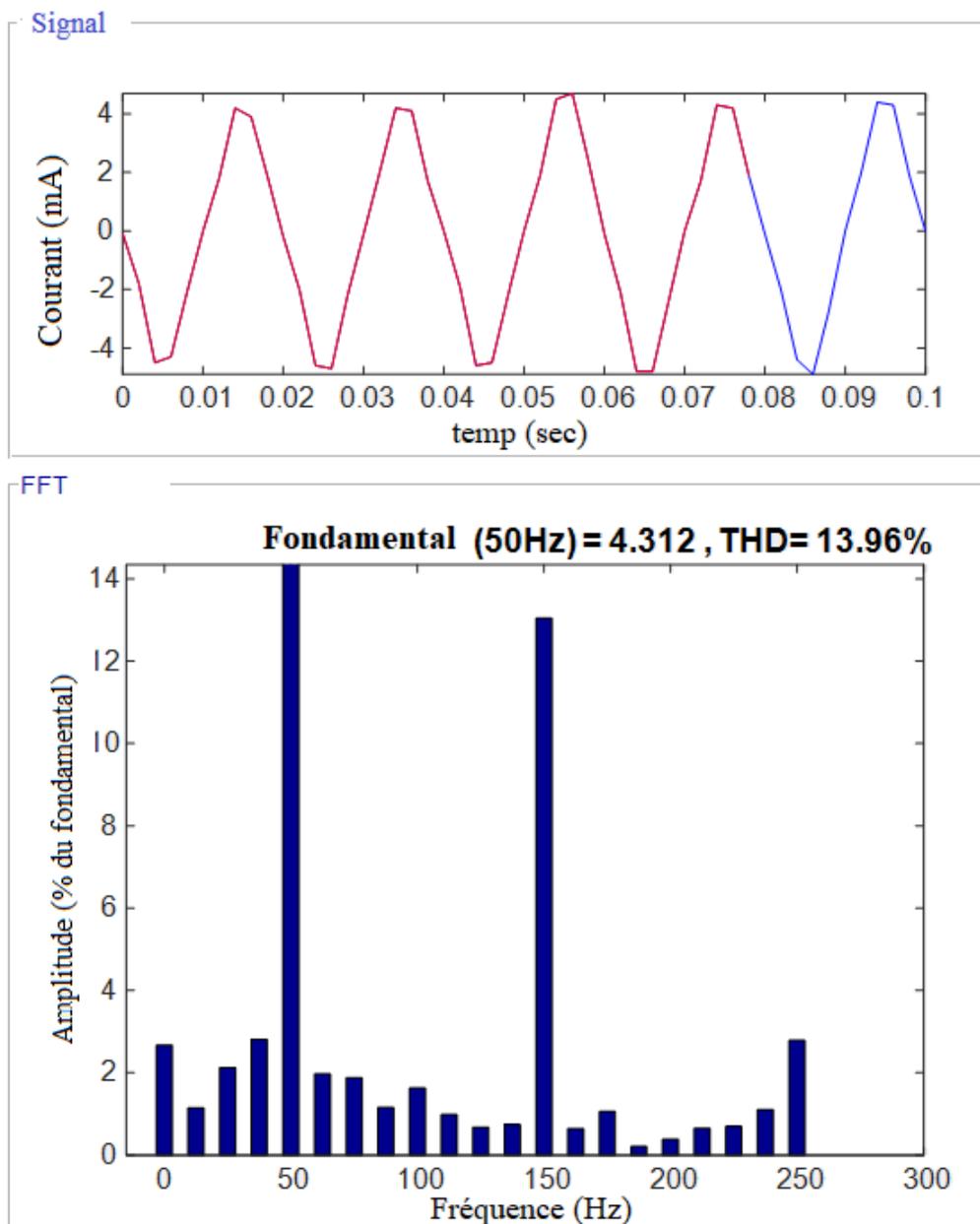


Figure III.17. Signal du courant de fuite et sa représentation fréquentielle (FFT).

Le signal du courant de fuite (Figure III.17) adopte une forme moins déformée qui se rapproche plus d'une forme sinusoïdale et ça due aux faits que le régime intermittent d'arcs partiels disparaît pour céder la place à un régime plus stable, l'analyse fréquentielle nous montre la diminution de l'amplitude des harmoniques. L'amplitude du cinquième harmonique (250 Hz) devient très faible devant l'amplitude du troisième harmonique (150 Hz) qui demeure significative par rapport à l'amplitude du fondamental.

La présence du troisième harmonique provient principalement de la présence d'arcs électriques qui s'éteignent et se ré-allument à chaque demi-période et de part la non-linéarité existant entre le courant d'arc et la tension appliquée [41-42].

4. Conductivité $\sigma = 10,1 \text{ mS/cm}$

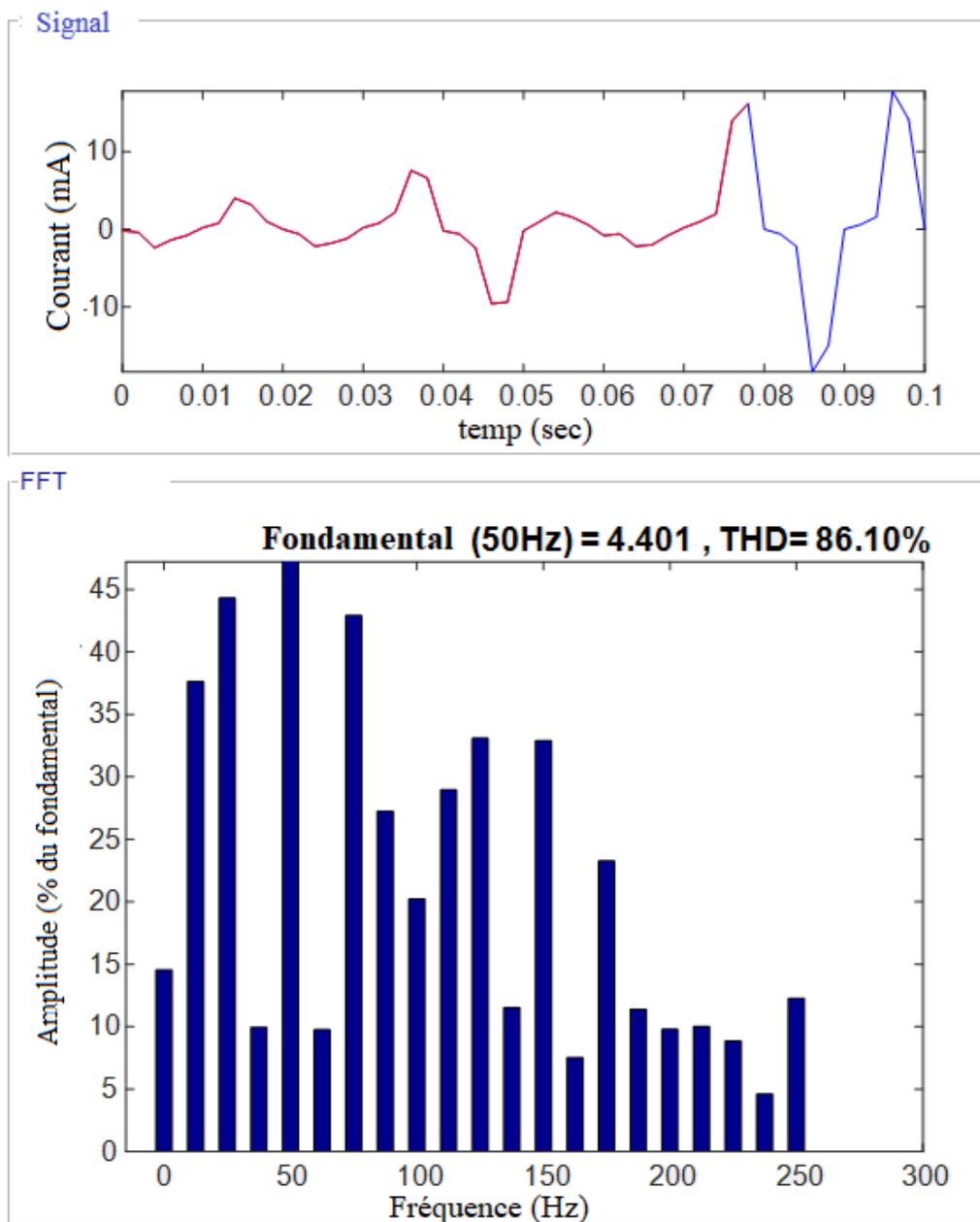


Figure III.18. Signal du courant de fuite et sa représentation fréquentielle (FFT).

Dans ce cas, la conductivité de la pollution est très importante, l'activité des arcs électriques s'intensifie. L'amplitude du courant de fuite augmente et les arcs électriques partiels deviennent plus intenses et moins stables. Cela affecte donc la forme d'onde du courant de fuite (Figure III.18) qui devient moins symétrique et perd sa forme sinusoïdale. Cela se traduit par une augmentation de l'amplitude du fondamental et proportionnellement celle des troisièmes et cinquièmes harmoniques.

III.3.3 Déphasage entre le courant de fuite et la tension appliquée

III.3.3.1 Mesure du déphasage

La mesure du déphasage courant de fuite-tension appliquée est déterminée à l'aide de la transformée de Fourier discrète basée sur la FFT, grâce à un Programme MATLAB® qui permet de calculer le déphasage entre le fondamental du courant de fuite et celui de la tension appliquée.

Le déphasage est défini par :

$$\varphi = |\Delta\varphi_c - \Delta\varphi_t| \quad (\text{III.3})$$

Avec :

$\Delta\varphi_c$ C'est le déphasage du fondamental du signal du courant de fuite

Et

$\Delta\varphi_t$ C'est le déphasage du fondamental du signal de la tension appliquée.

III.3.3.2 Etude du déphasage courant-tension en fonction de la conductivité

La variation du déphasage courant-tension en fonction de la conductivité est représentée dans la figure III.19 et III.23.

1. Conductivité 0.01, 0.19, 0.71 et 1.2 mS/cm

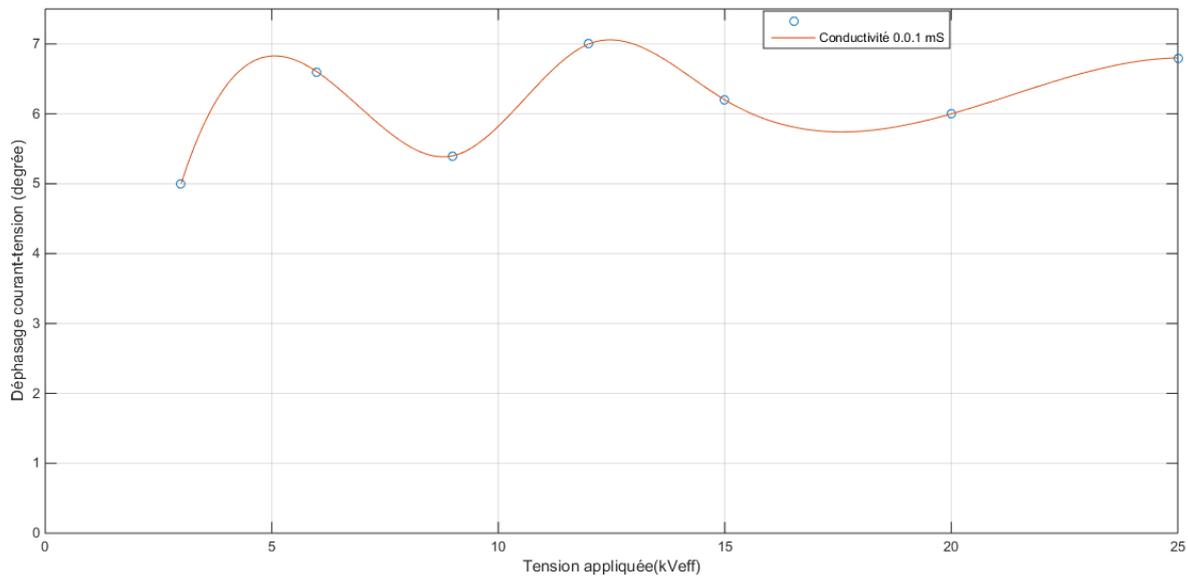


Figure III.19. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.01 mS/cm.

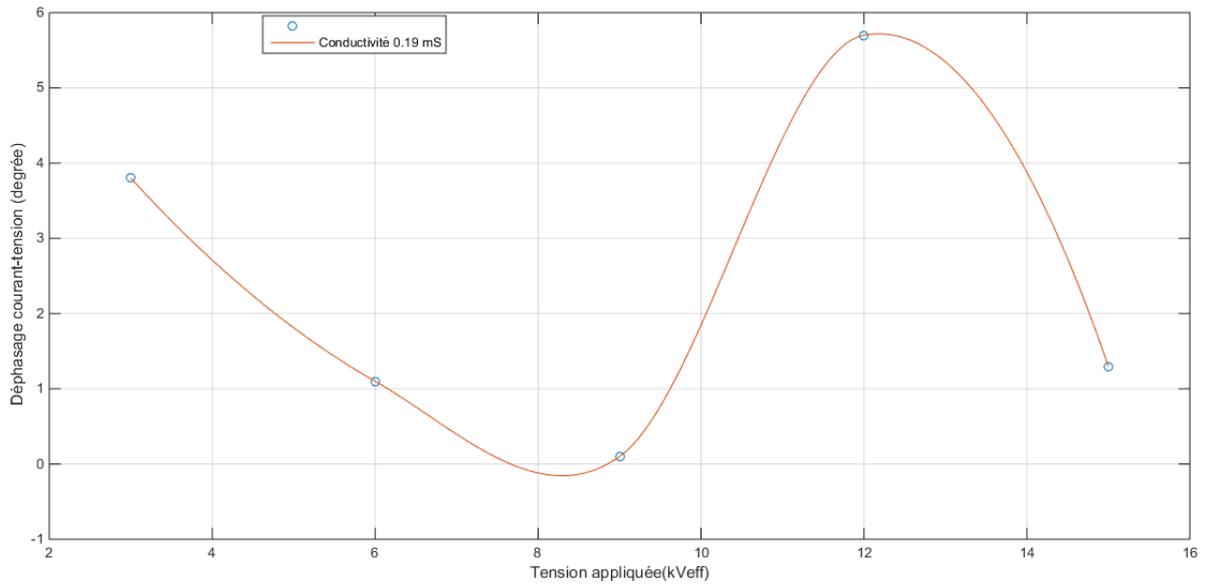


Figure III.20. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.19 mS/cm.

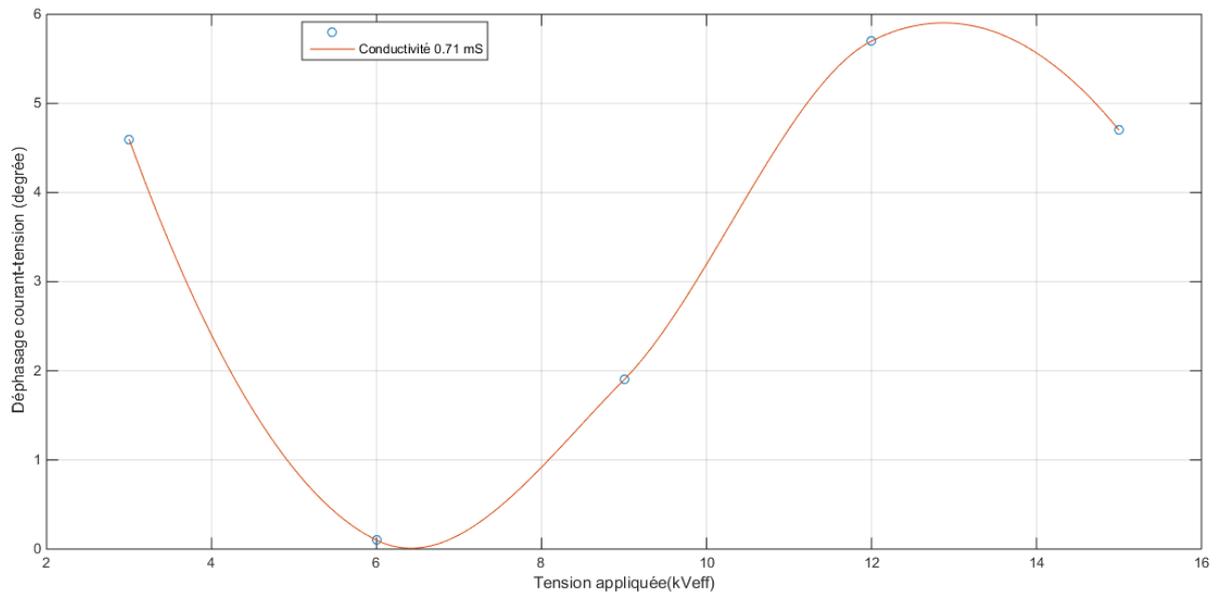


Figure III.21. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 0.71 mS/cm.

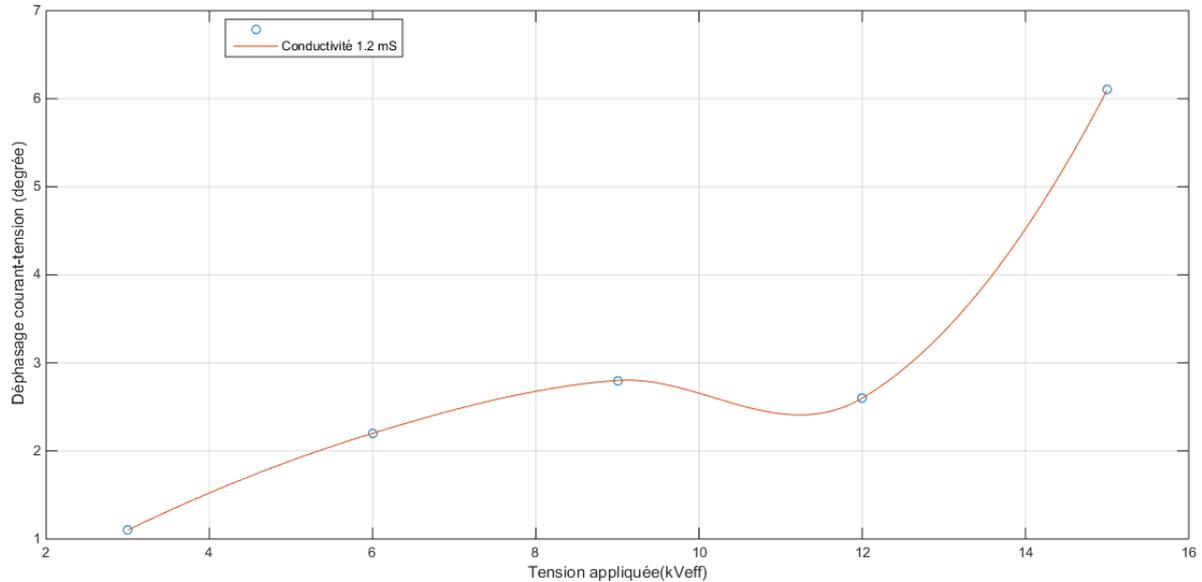


Figure III.22. Variation du déphasage courant-tension en fonction de la tension appliquée pour une conductivité 1.2 mS/cm.

On remarque que le déphasage courant-tension diminue avec l'augmentation de la conductivité du milieu (augmentation de la pollution). De plus, à partir ces résultats on peut observer que plus que la conductivité augmente plus que le courant de fuite devient en phase avec la tension appliquée. Ceci nous indique que le comportement de l'isolateur est en grande partie résistif avec un très léger effet capacitif.

A titre d'exemple, pour une tension appliquée de 12 kVeff, on observe à partir des figure III.19 a III.22 :

- Pour une conductivité de 0.01 mS/cm le déphasage est de 8°
- Pour une conductivité de 0.19 mS/cm le déphasage est de 6°
- Pour une conductivité de 0.71 mS/cm le déphasage est de 5.8°
- Pour une conductivité de 1.2 mS/cm le déphasage est de 2.8°

2. Conductivité $\sigma = 10,1 \text{ mS/cm}$

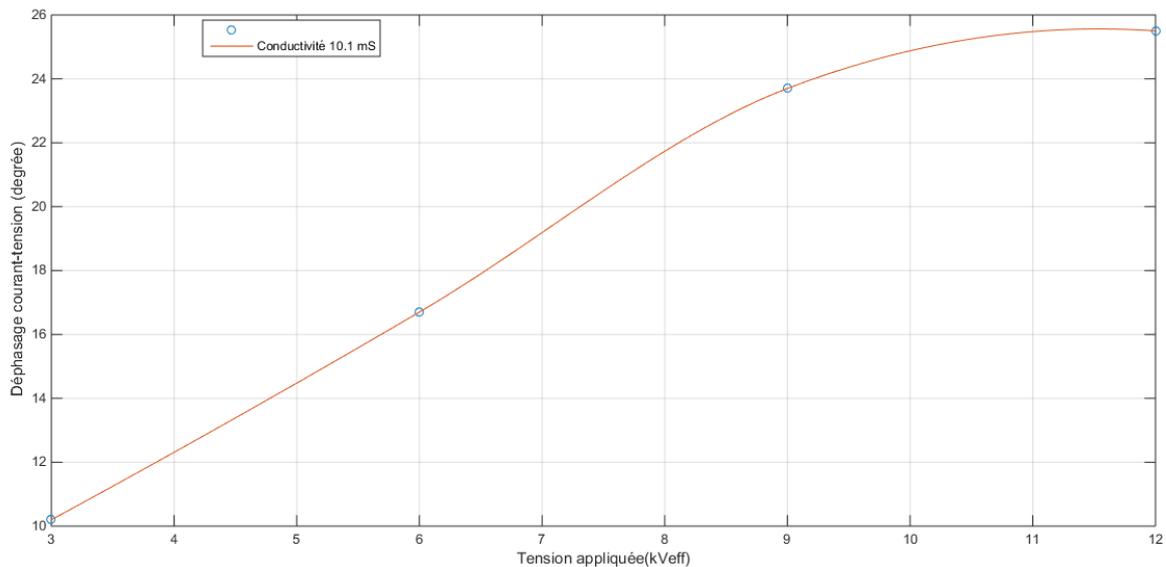


Figure III.23. Variation du déphasage courant-tension en fonction de la tension appliquée.

Pour la valeur de la conductivité $10,1 \text{ mS/cm}$, le déphasage est plus important par rapport aux autres conductivités. Comme le montre la figure III.23, l'activité des décharges est déjà importante à partir de la tension 3 kVeff , par conséquent y'aura une augmentation des zones sèches due à l'augmentation de la température. De plus, l'augmentation de la tension appliquée va favoriser la création de zones sèches, et au cours du temp le comportement résistif de l'isolateur pollué va donc diminuer et l'effet capacitif va augmenter.

III.4 Conclusion

L'analyse du courant de fuite nous permet de conclure que l'amplitude de celui-ci augmente en fonction de la tension appliquée et de la conductivité du milieu, par contre une diminution de l'amplitude est apparue et cela nous informe sur les premières décharges électriques qui commencent à apparaître, donc on peut dire que le calcul de l'amplitude du courant de fuite peut être considéré comme un bon indicateur d'initiation d'un contournement.

Les variations de la déviation standard (STD) et la valeur moyenne (RMS) sont très utiles pour caractériser la contamination par la pollution d'un isolateur, donc elles sont un bon indicateur pour surveiller les performances d'un isolateur pollué.

Analyse spectrale des harmoniques du signal du courant de fuite a montrée qu'il y a une corrélation entre les amplitudes du 3eme et 5eme harmonique du courant de fuite et la variation de la conductivité de la pollution.

L'analyse fréquentielle du déphasage courant-tension nous permet de conclure que ce dernier diminue en fonction de l'augmentation de la conductivité du milieu (augmentation de la pollution), De ce fait, déphasage courant-tension peut être considéré comme un bon indicateur pour surveiller la dégradation de la performance de l'isolateur.

**Chapitre IV Utilisation
des RNA pour prédire la
valeur de la conductivité
de la pollution à partir
du courant de fuite**

Introduction

Dans notre travail, nous avons pris les résultats des expériences déroulés aux laboratoires de haute tension de l'ENP, et à partir des informations collecter des ces expériences nous allons essayer de développer un modèle RNA capable de prédire la valeur de la conductivité d'un diélectrique d'une chaîne d'isolateurs réels 1512 L uniformément pollué, sans besoin d'intervention des experts.

Dans ce qui suit nous allons expliquer les étapes que nous avons suivis pour configurer et choisir un bon modèle RNA qui va nous aider à atteindre notre objectif qui est la prédiction de la valeur de la conductivité, en évitant les problèmes qu'on a rencontrés durant les essais qu'on a fait.

IV.1 Collection et organisation des données

La corrélation entre le courant de fuite étant prouvé à travers le chapitre 3, l'étape actuelle consiste à organiser notre base de données qui est composée de 103 échantillons sous la forme suivante :

IV.1.1 Partition de la base des données

L'organisation et la partition de notre de base de données sont basées sur l'évaluation de l'erreur de généralisation qui est en général fondée sur la partition de notre base des données disponibles (103 échantillons) en trois sous-ensembles.

Le premier sous-ensemble appelé ensemble d'apprentissage est utilisé pour optimiser les poids du réseau correspondant à une architecture donnée.

Le second sous-ensemble appelé ensemble de validation permet de comparer plusieurs architectures de réseaux et de retenir la meilleure. Le deuxième ensemble permet également de contrôler et de mesurer la généralisation du réseau au cours de l'apprentissage.

Enfin, le troisième sous-ensemble appelé ensemble de test sert à estimer l'erreur de généralisation du réseau déterminé d'après les ensembles d'apprentissage et de validation. Le lecteur pourra trouver une comparaison de ces méthodes dans [46].

Cette division des données est faite aussi pour éviter le problème de sur-apprentissage (overfitting) ou sous-apprentissage (underfitting).

- **Overfitting** : quand un modèle trop spécialisé sur les données d'entraînement, et qui se généralisera mal, ça veut dire que notre modèle peut prédire les données d'entraînement avec une grande précision, mais il prédira mal sur des données qu'il n'a pas encore vues lors de sa phase d'apprentissage.
- **Underfitting** : un modèle généraliste incapable de fournir des prédictions précises, donc il ne peut même pas prédire les points utilisés pendant la phase d'entraînement.

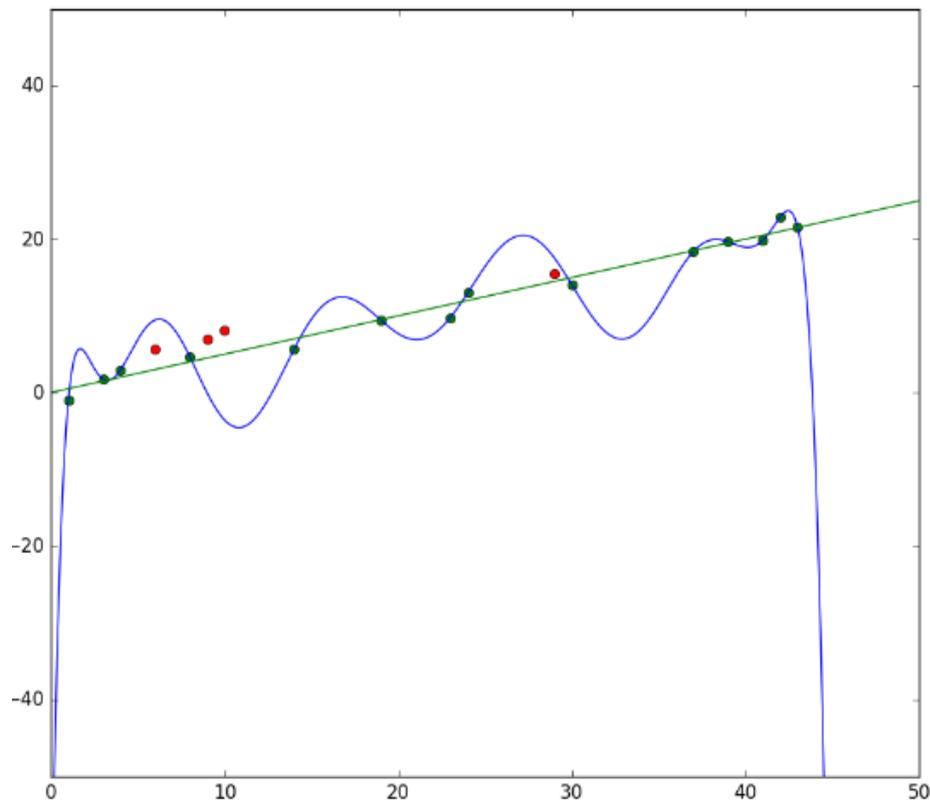


Figure IV.1 courbe montre le problème de sur-apprentissage [47]

Comme on voit dans la figure IV.1, la courbe tracée en blue c'est la fonction établie à partir de l'apprentissage d'un réseau utilisant les points en vert, on voit que cette fonction passe par tous les points verts par contre elle ne passe pas par les points rouges qui représentent les données de tests [47].

On peut diviser cet ensemble automatiquement en utilisant des fonctions prédéfinies dans MATLAB, mais il faut bien organiser cette base de données, en plus il faut savoir aussi comment ces fonctions vont diviser notre base de données, pour éviter toutes les contraintes.

Voici les fonctions utilisées dans notre étude pour diviser les données :

- La division par block (**divideblock**) : cette fonction nous permet de diviser automatiquement notre base de données en trois ensembles en utilisant des blocs d'indices [47].

[trainInd,valInd,testInd]= divideblock(103,0.5,0.25,0.25);

Comme on voit on peut juste donner les pourcentages de chaque sous-ensemble, donc il faut organiser notre ensemble de tel façon d'avoir les 5 classes avec les différents paliers de tensions dans chaque ensemble.

- **dividerand**: cette fonction nous permet de diviser la base de données en trois ensembles en utilisant des indices aléatoires [47].

[trainInd,valInd,testInd] = dividerand(Q,trainRatio,valRatio,testRatio)

La fonction de division aléatoire c'est la fonction la plus utilisée, on l'a utilisé dans nos essais, parce que on peut adapter les indices de chaque sous-ensemble pour avoir les 5 classes avec différents paliers de tensions.

IV.2 Choix des types des réseaux de neurones

Maintenant on passe au choix des types des réseaux des neurones qui ont été prouvé leurs efficacités de traiter ce genre de problème. Basant sur les articles lus et résumés dans chapitre I, on voit que la plupart des problèmes de prédiction sont traités en utilisant les deux types de RNA suivants :

IV.2.1 Feed-forward back propagation

Ce type de RNA représente le perceptron multicouche qui est parmi les réseaux de neurone les plus utilisés, possède plusieurs paramètres, qui sont :

- Les données d'entrées (Input data)
- Les données de sortie (Target data)
- Fonction d'entraînement (Training fonction)
- Fonction de performances (Performances function)
- Nombre des couches cachées (Nombre of layers)
- Nombre de neurones par couche (Nombre of neurones of each layer)
- Fonction de Transfer (Transfer function)

IV.2.2 ELMAN backpropagation

Ce type a été appliqué avec succès dans de nombreux domaines, en matière de prévision, de modélisation et de contrôle. Le réseau Elman est un réseau de neurones récurrents (RNN) qui ajoute des liens récurrents à la couche cachée en tant que connexion de retour, et il possède les mêmes paramètres du type précédent.

IV.3 Choix des paramètres des deux types de RNA

Les deux types de RNA choisis possèdent les mêmes paramètres, on va fixer quelques paramètres qui n'influence pas beaucoup sur les résultats par exemples les paramètres pour optimiser le temps d'apprentissage, nous allons les fixer d'une manière optimale d'après les travaux antérieures et varier le reste.

Dans ce qui suit nous allons expliquer chaque paramètre d'une manière succincte.

IV.3.1 Entrées (Inputs)

C'est l'ensemble d'apprentissage structuré comme indiqué dans la partie précédente.

Voici un exemple :

Tableau IV.1 Exemple d'une matrice d'entrée de type fréquentielle de classe 10 mS/cm.

Modelé	10.1mS-cm 3kV(m2)	10.1mS-cm 6kV(m2)	10.1mS-cm 9kV(m2)	10.1mS-cm 12kV(m2)
Déphasage LC et Vc (°)	10,2	6,7	17,7	25,5
3h/f(%)	12,01	17,19	22,88	29,31
5h/f(%)	3,28	7,02	19,34	10,37
V (kV)	3	6	9	12

L'entrée du modèle RNA sera par vecteur colonne de dimension 4, puis a partir de cela le réseau essayera de prédire la classe (la conductivité) que ce vecteur appartient.

IV.3.2 Sorties (Outputs)

C'est les valeurs de sortie désirée de notre réseau de neurones qu'on veut trouver en utilisant RNA à partir des entrées par exemples les sorties correspondant à l'exemple des entrées indiqué dans le tableau précédant sont :

Tableau IV.2 Les différentes valeurs de conductivités utilisées.

Conductivité (mS/cm)	10,1	10,1	10,1	10,1	10,1
-------------------------	------	------	------	------	------

IV.3.3 Fonctions d'entraînement (training fonction)

Il existe plusieurs fonctions d'entraînement, mais elles sont toutes des dérivées de retro-propagation et chaque fonction possède ses propres caractéristiques et ces utilisations, dans notre cas on a choisi deux fonctions d'entraînements qui sont les plus utilisés dans ce genre de problème.

IV.3.3.1 Fonction d'entraînement traingdx (Gradient descent with momentum and adaptive learning rate backpropagation)

Cette fonction d'apprentissage est une version améliorée de l'algorithme de gradient descent, elle est caractérisée par les paramètres suivants :

- Taux d'apprentissage (Rate Learning=0.001) :
- Nombre des itérations maximale pour arrêter le processus d'apprentissage (Epochs=1000), le choix des itérations est aléatoire.
- Le performance désiré (MSE=0)
- Le moment est utilisé pour remédier aux minimums locaux, il est généralement compris entre 0 et 1, (l'explication détaillées du moment est dans le chapitre II). Dans notre approche, sa valeur égale 0.8.

IV.3.3.2 Fonction d'entraînement Trainbr (Bayesian régularization backpropagation)

Trainbr est une fonction d'entraînement qui met à jour les valeurs de poids et de biais en fonction de l'optimisation de Levenberg-Marquardt. Il minimise une combinaison d'erreurs au carré et de poids, puis détermine la combinaison correcte de manière à produire un réseau qui généralise bien. Le processus s'appelle régularisation bayésienne.

La fonction trainbr contient les mêmes paramètres que traingdx (à part le moment), et dans nos essais on va fixer ces paramètres comme indiqué précédemment.

IV.3.4 Outil d'évaluation de la performance d'un modèle RNA

On peut suivre l'évaluation de performance d'un réseau RNA en utilisant les approches suivantes :

IV.3.4.1 Par une fonction d'erreur

Il existe trois types de fonctions d'erreur pour évaluer la performance d'un réseau de neurones qui sont prédéfinies sous MATLAB (développées dans le chapitre II), dans notre travail nous allons choisir MSE comme fonction d'estimation de l'erreur. Chaque sous-ensemble indiqué en haut est reliée à une fonction d'erreur MSE qui évalue l'erreur du réseau de neurone en fonction des itérations effectuées.

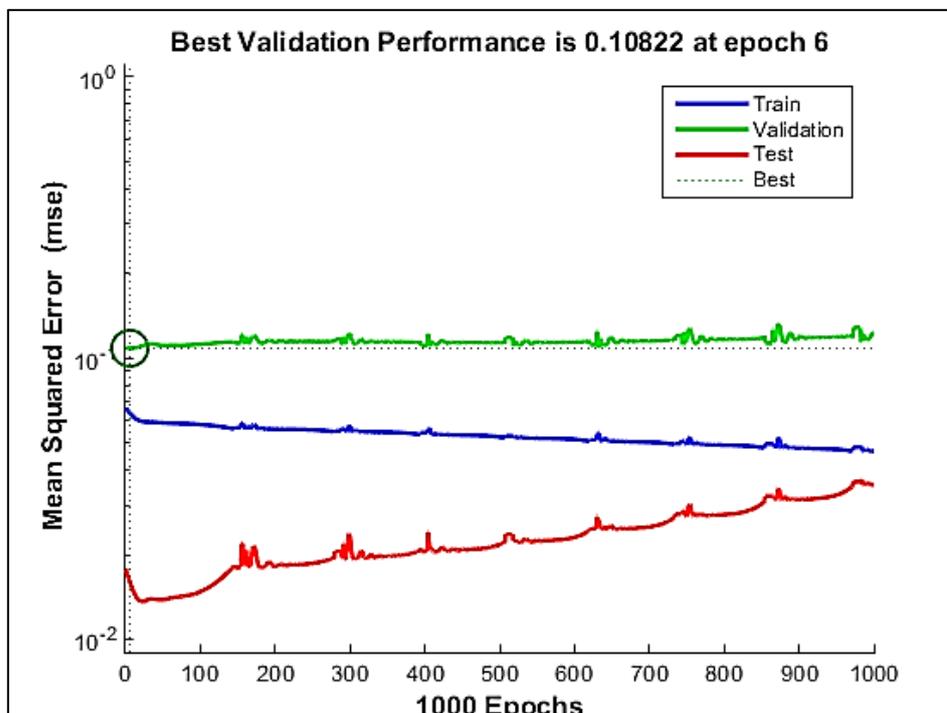


Figure IV.2 Courbe d'évolution de la fonction MSE

La figure IV.2 représente les valeurs de MSE de chaque sous-ensemble en fonction de nombre d'itérations.

IV.3.4.2 Régression linéaire

On peut aussi évaluer les performances d'un réseau RNA par la méthode de régression linéaire entre les sorties prédites par ces réseaux et les sorties désirés (valeurs réelles du conductivité) par :

$$\text{Output} = a * \text{Target} + b \quad (\text{IV.1})$$

avec :

- Outputs : les sorties de réseau
- a : la pente
- traget : les cibles ou les sorties désirés
- b : la meilleure régression linéaire reliant les cibles(Traget) aux sorties du réseau (outputs).

Si on considère un bon modèle parfait, ça veut dire cibles= sorties du réseau. Implique $a=1$ et $b=0$.

R : Coefficient de corrélation

Plus R est proche de 1 plus nous aurons un bon ajustement entre les cibles et les sorties du réseau, la figure ci-dessous montre un exemple d'une régression linéaire :

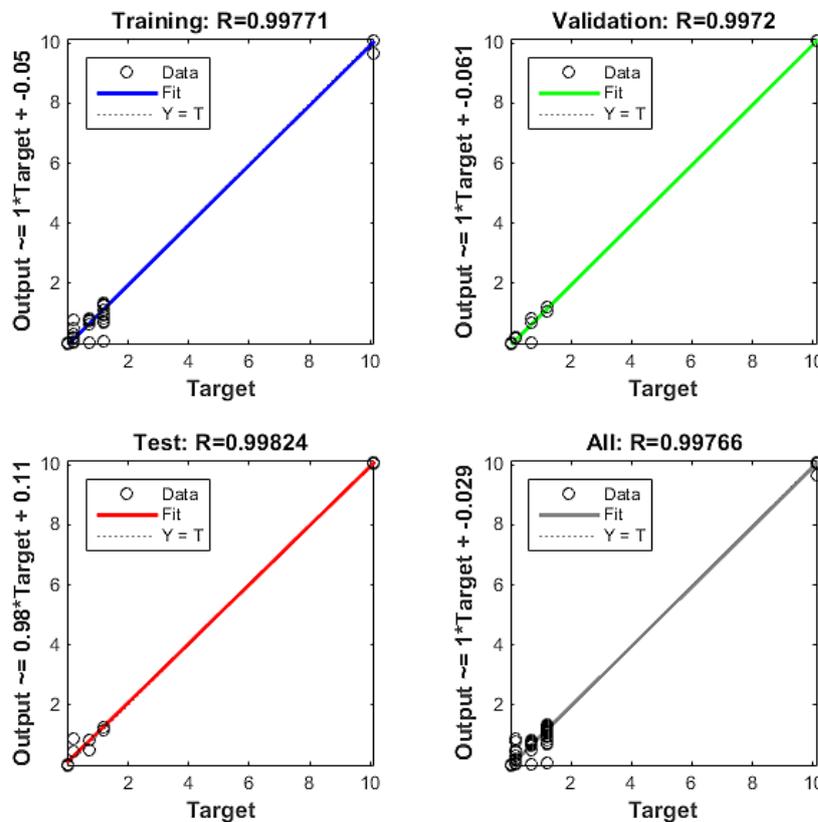


Figure IV.3 les performances d'un réseau de neurones par la méthode de régression linéaire.

La Figure IV.3 représente un exemple de la régression linéaire de nos trois sous-ensembles, ou on remarque que les deux lignes continue et discontinue sont superposés, ça traduit une corrélation presque parfaite entre les sorties de réseau les cibles.

IV.3.4.3 Erreur Relative

Dans notre étude pour suivre l'évolution de l'erreur de test d'un réseau après l'apprentissage, on calcule l'erreur relative de sous-ensemble de test par la formule suivante :

$$E(\%)=100 * \sum_1^{25} \left| \frac{(re-pre)}{re} \right| \quad (IV.2)$$

- Re : représente une matrice de dimension [25*4] des valeurs mesures des vecteurs de tests.
- pre=network1(1): les résultats de la prédiction en utilisant le modèle entraîné.

IV.3.5 Choix de nombre des couches cachées et des neurones de chaque couche

C'est l'étape la plus difficile de la configuration de n'importe quel RNA, il n'y a pas des lois ou des règles pour choisir le nombre des couches cachées ou de nombre des neurones de chaque couche, donc il faut faire plusieurs tests en utilisant les données du problème posé, et suivre l'évolution de performance du modèle.

Donc notre cas on a choisi de varier le nombre des neurones par couches de 2 à 16, pour une seule couche cachée puis par la suite pour deux couches cachées.

IV.3.6 Initialisation des poids et des biais

Il faut éviter d'initialiser les poids et les biais par des valeurs nulles, pour résoudre ce problème on peut utiliser des fonctions prédéfinies sous MATLAB comme :

- **rand** : fonction d'initialisation symétrique aléatoire des poids et biais.
- **randnc** : Fonction d'initialisation de poids de colonne normalisée.
- **randnr** : Fonction d'initialisation de poids de ligne normalisée.

Ces sont les fonctions la plus utilisés pour initialiser les poids et les biais au début d'apprentissage, mais quand on trouve un modèle avec une performance désiré, et on va voir si on peut l'améliorer, on prend les valeurs des poids et biais de ce modèle et on les met comme des valeurs initiales et on entraîne ce modèle encore avec nombre d'itérations réduits, et on suit l'évolution de performance, s'il s'améliore donc on va obtenir un meilleur modèle sinon on préserve ce modèle.

IV.3.7 Choix de fonction d'activation

Pour choisir les fonctions d'activation qu'on va travailler avec, nous allons étudier les performances d'un réseau de neurone de type BP avec une seul couche cachée utilisant le vecteur fréquentiel en variant les fonctions d'activation, pour se faire, nous allons effectuer les différents tests indiqués dans le tableau ci-dessous

Nos tests sont basés sur les fonctions suivantes :

Tableau IV.3. Configurations proposées pour choisir les fonctions d'activation

Configuration	FONCTION D'ACTIVATION		Nombre de neurones
	Couche cachée	Sortie	
1	LOGSIG	LOGSIG	De 2 à 16
2	LOGSIG	TANSIG	De 2 à 16
3	LOGSIG	PURELIN	De 2 à 16
4	TANSIG	LOGSIG	De 2 à 16
5	TANSIG	TANSIG	De 2 à 16
6	TANSIG	PURELIN	De 2 à 16
7	PURELIN	LOGSIG	De 2 à 16
8	PURELIN	TANSIG	De 2 à 16
9	PURELIN	PURELIN	De 2 à 16

Nous avons choisi 3 types de fonctions d'activation qui sont les plus utilisées (voir chapitre II) donc on obtient 9 arrangements donc 9 configurations possible (Tableau IV.3.) avec un réseau avec une seule couche cachée.

Dans chaque configuration on varie le nombre de neurone de 2 à 16, par conséquent on obtient 15 modèles différents, il y a 9 configurations donc en tout nous avons simulé 135 modèles.

Notre choix est basé sur les erreurs citées avant, nous avons choisi la configuration qui donne les erreurs minimales.

Les performances des différentes configurations sont :

- Erreur de sous-ensemble d'entraînement

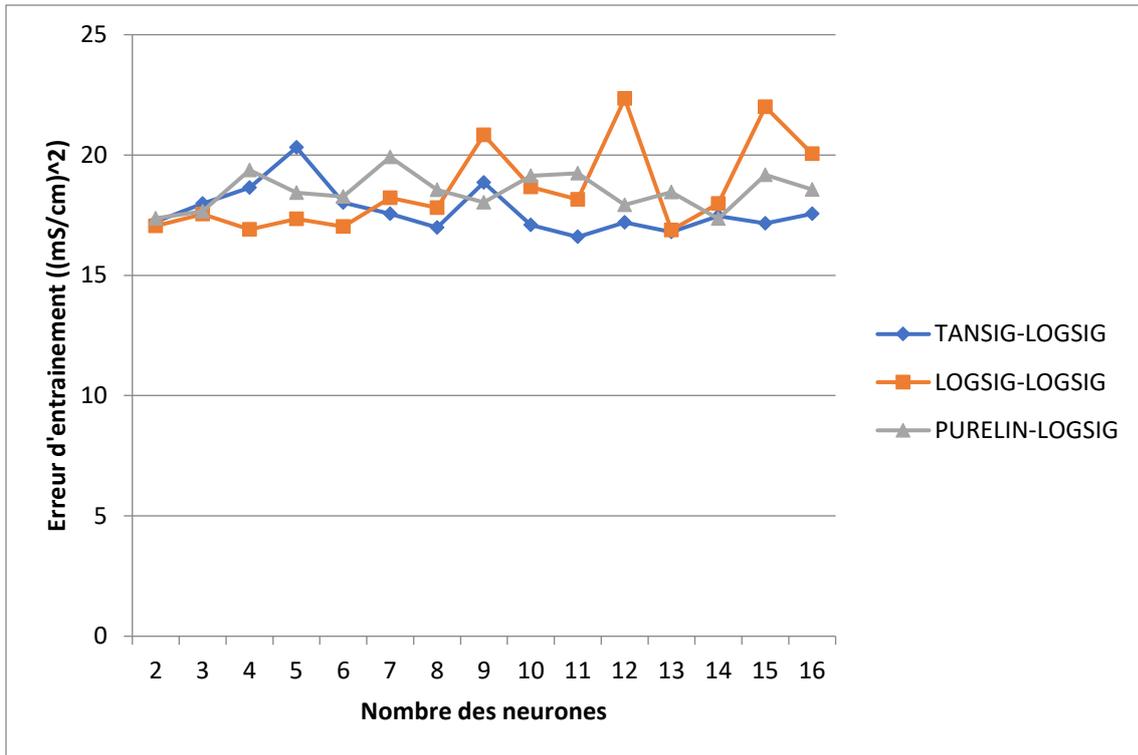


Figure IV.4. Erreur d'entraînement des trois configurations 4, 1 et 7.

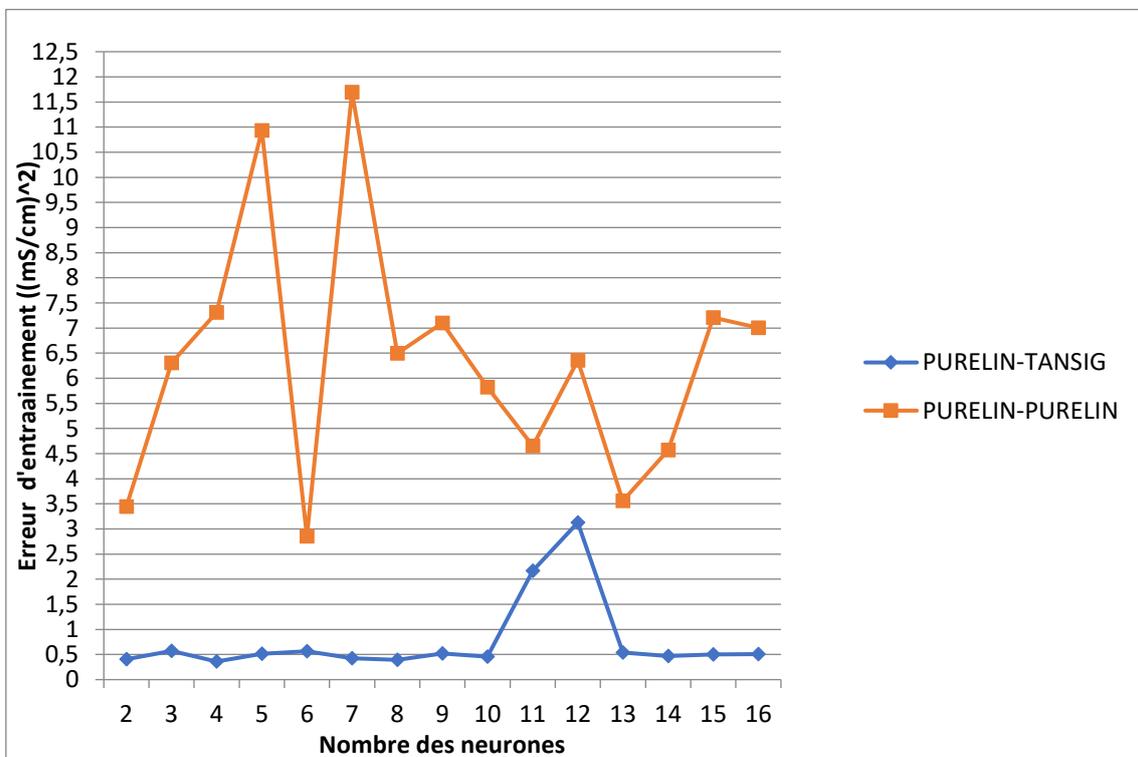


Figure IV.5. Erreur d'entraînement des trois configurations 8 et 9.

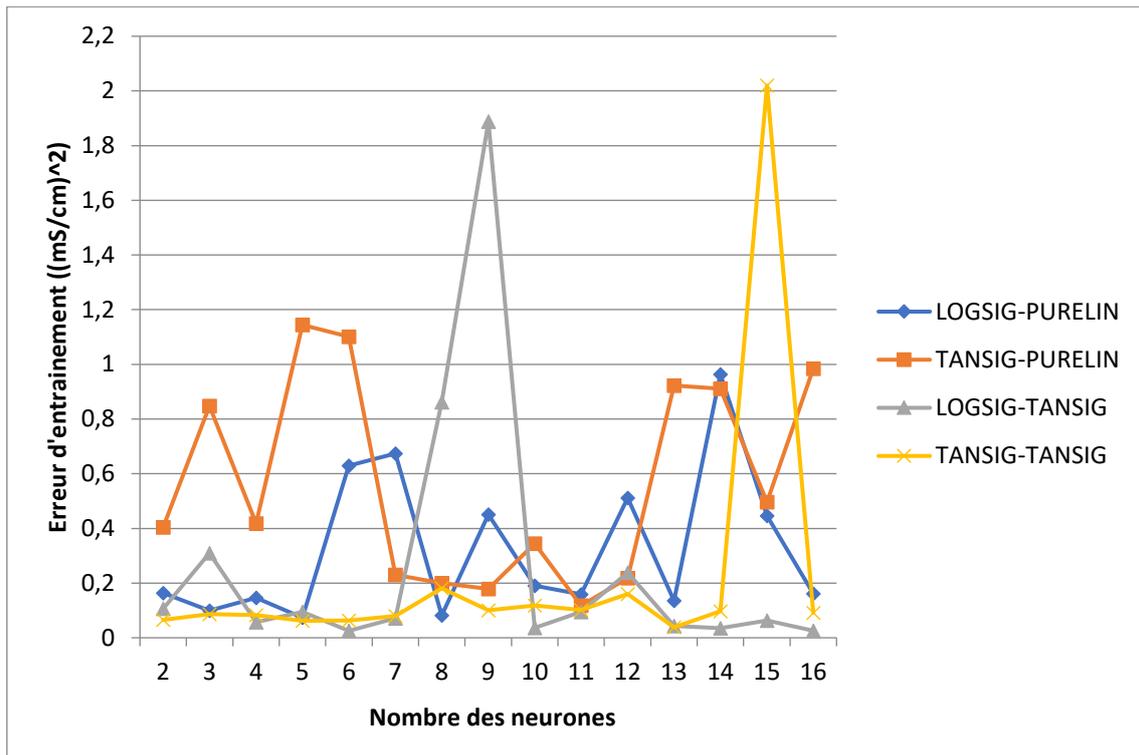


Figure IV.6. Erreur d'entraînement des quatre configurations 3, 6, 2 et 5

Interprétation

Dans la figure IV .4, on remarque que les configurations ne donnent plus des précisions acceptables, elles donnent des erreurs d'entraînements les plus élevées qui sont aux alentours de 17 mS/cm².

D'après la figure IV .5, on voit que la configuration (PURELIN-TANSIG) possède une erreur plus que la configuration (PURELIN-PURELIN).

On remarque dans la figure IV.6 que la plupart des configurations donnent des erreurs d'entraînement inférieur à 0.2 mS/cm², à part la configuration (TANSIG-PURELIN), ses erreurs en fonction des nombres des neurones comprennent entre 0.2 et 1 mS/cm².

Nous remarquons qu'ils existe des modèles qui donnent une erreur d'entraînement élevée (les pics) par rapport aux autres modèles de même configuration mais avec un nombre de neurone différent comme le modèle avec 15 neurone de la figure IV.6, ceci due au problème d'un sous-apprentissage pendant l'entraînement.

• Erreur de sous-ensemble de validation

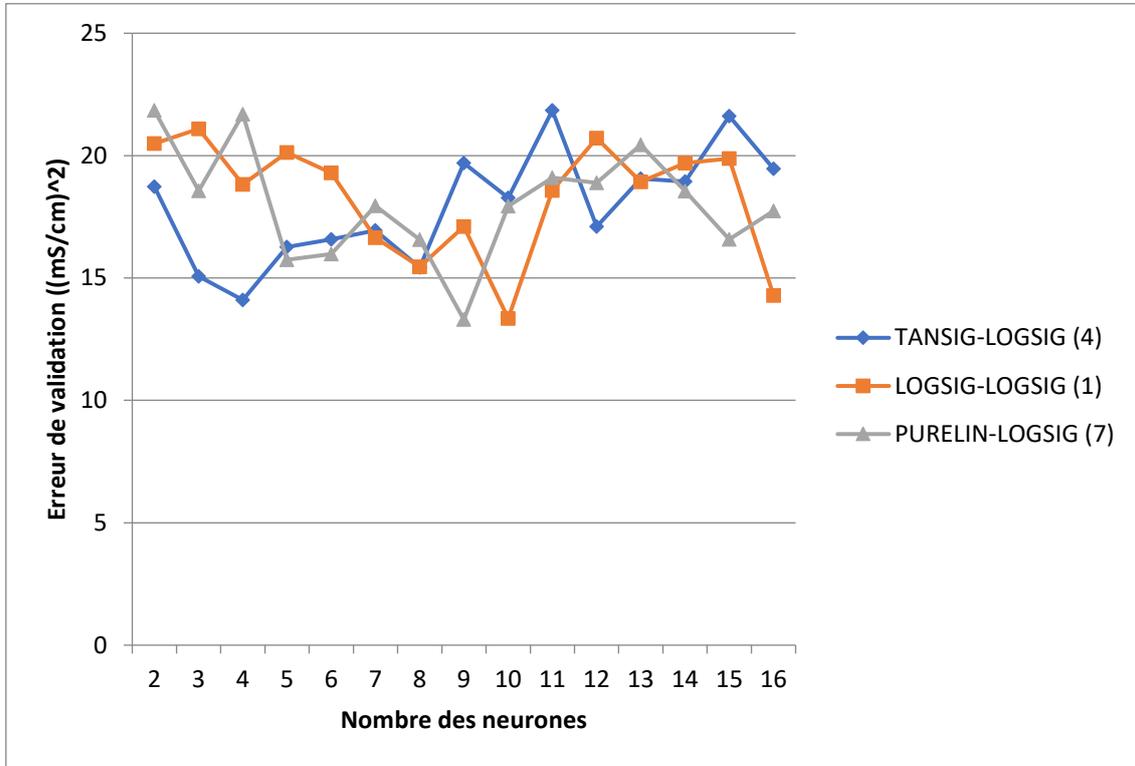


Figure IV.7. Erreur de validation des trois configuration 4, 1 et 7.

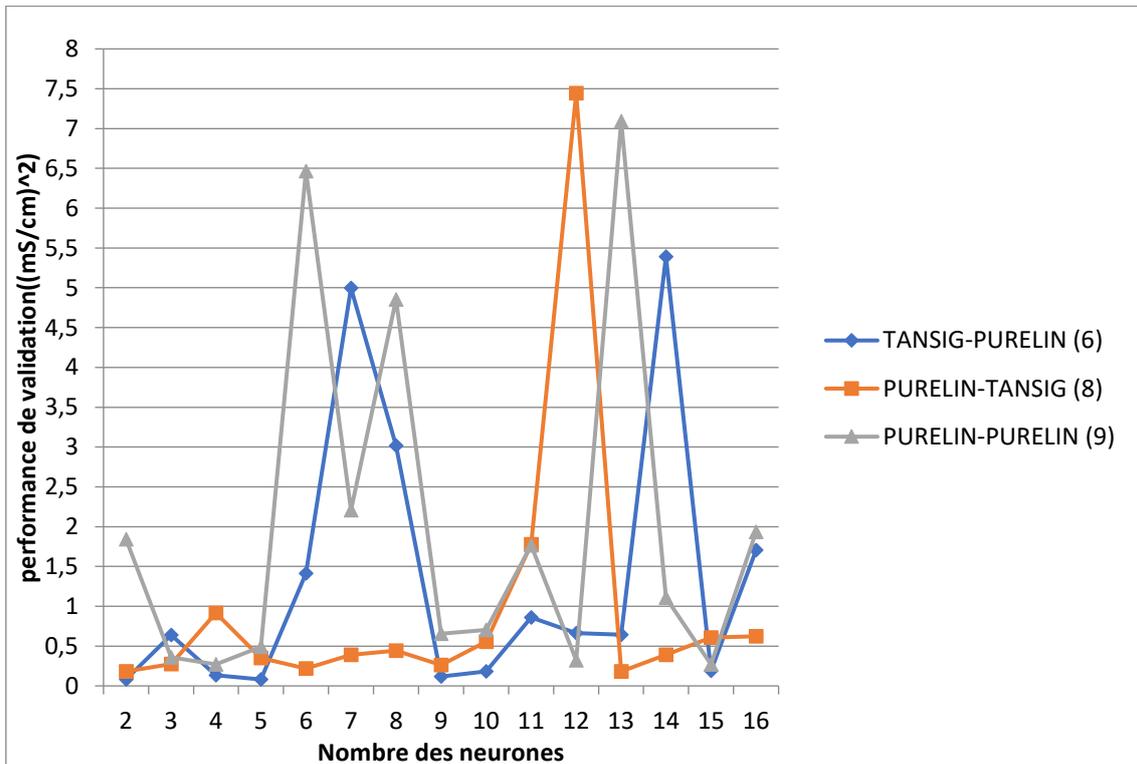


Figure IV.8. Erreur de validation des trois configuration 6, 8 et 9.

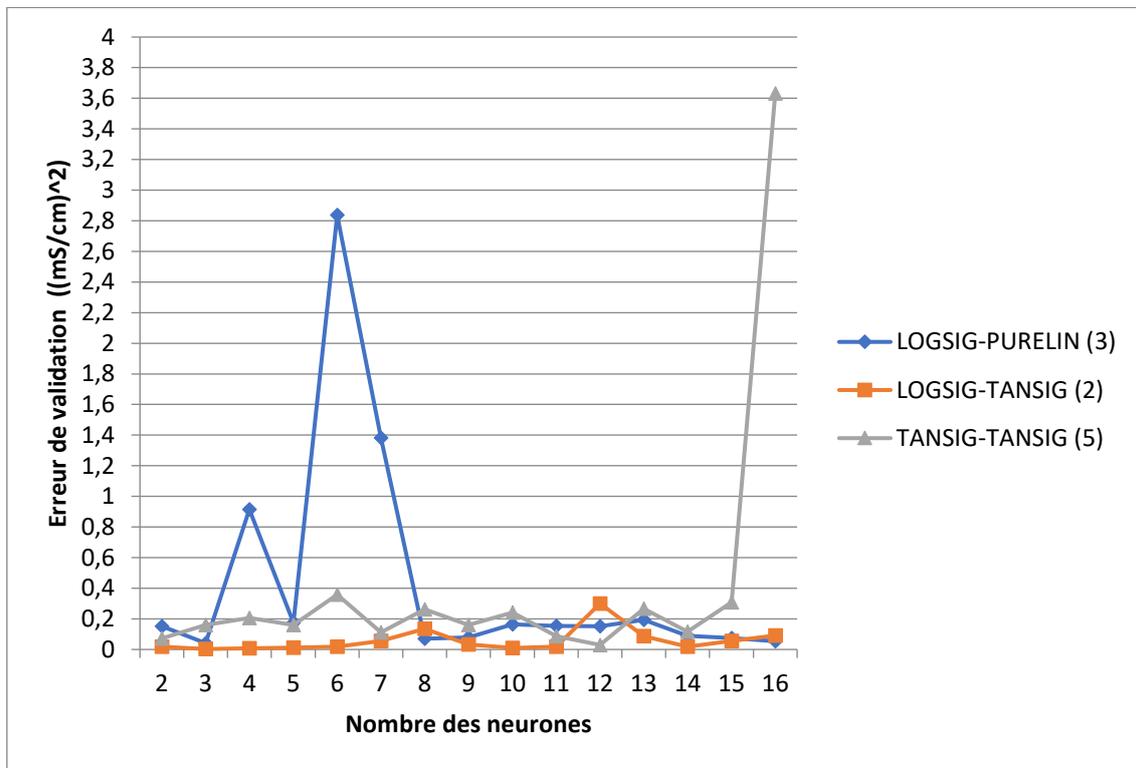


Figure IV.9. Erreur de validation des trois configuration 3, 2 et 5.

Interprétation

Dans la figure IV.7, les erreurs sont très importantes (de 15 jusqu'à $(23 mS/cm)^2$).

Dans la figure IV.8, les erreurs basculent entre des faibles valeurs ($(0.2 mS/cm)^2$) et des valeurs importantes jusqu'à $(7.5 mS/cm)$.

D'après la figure IV.9, l'erreur de validation pour la plupart des cas est autour de $0.2 mS/cm^2$, et les configurations (2) et (5) donnent encore des meilleures performances pour la plupart des neurones.

- Erreur de sous-ensemble de test

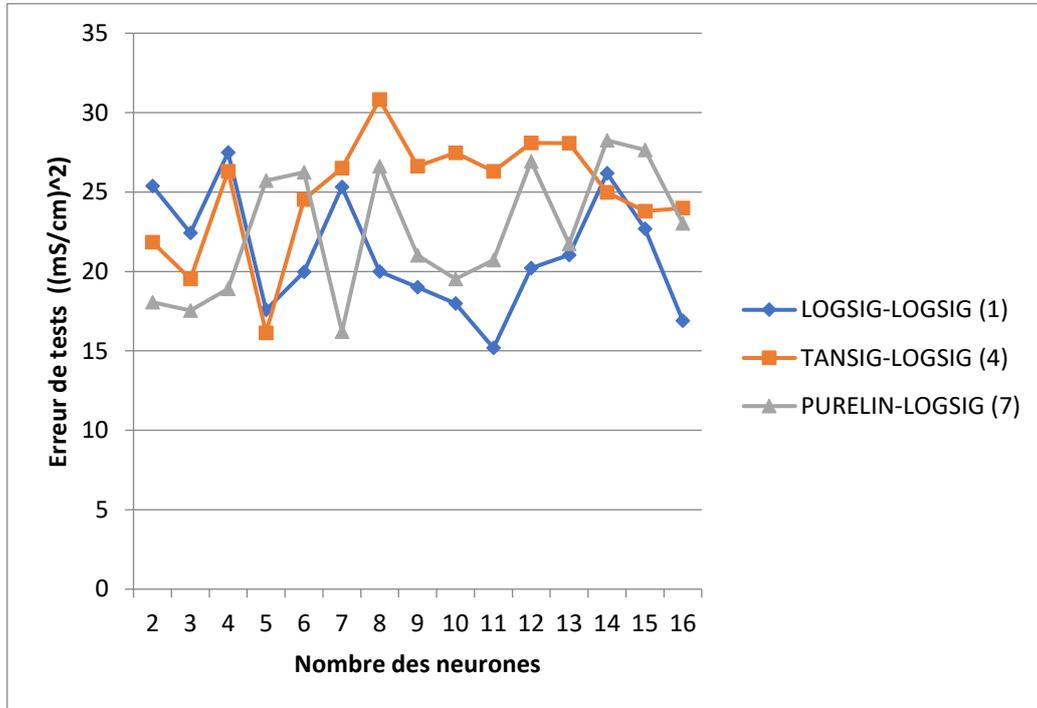


Figure IV.10. Erreur de test des trois configuration 4, 1 et 7.

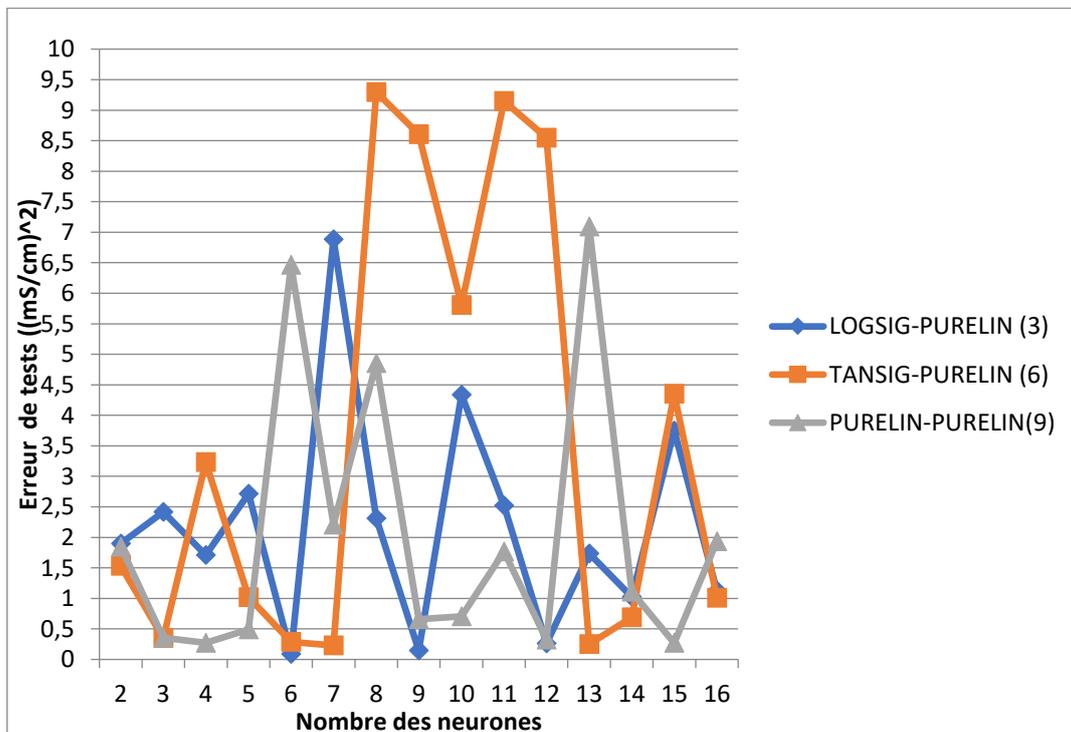


Figure IV.11. Erreur de test des trois configuration 3, 6 et 9.

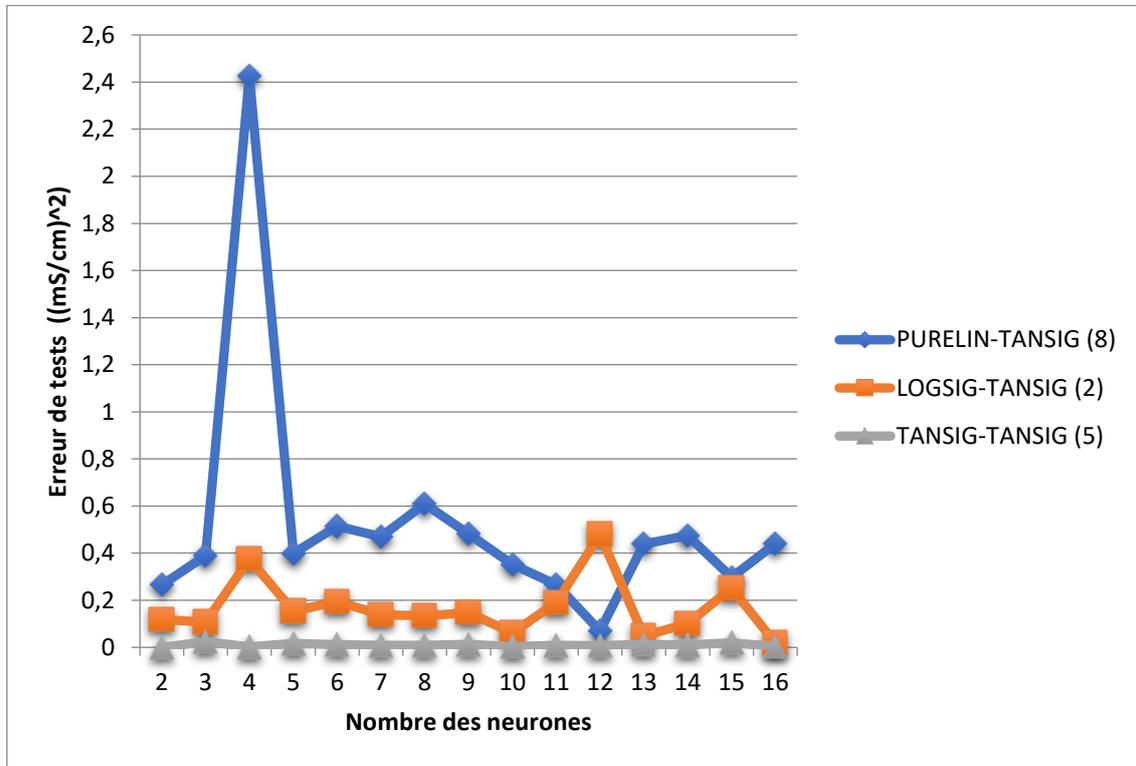


Figure IV.12. Erreur de test des trois configuration 8, 2 et 5.

Interprétation

D'après la figure IV.12, les erreurs de test comprises entre 0.2 et 2.4 (mS/cm)², et les configurations (2) et (5) donnent encore des meilleures performances pour la plupart des neurones.

- **Erreur relative de sous-ensemble de test Emin(%)**

Nous avons trouvé qu'il y a que ces trois configurations suivantes qui donnent une erreur relative pour la plupart des modèles inférieure à 100.

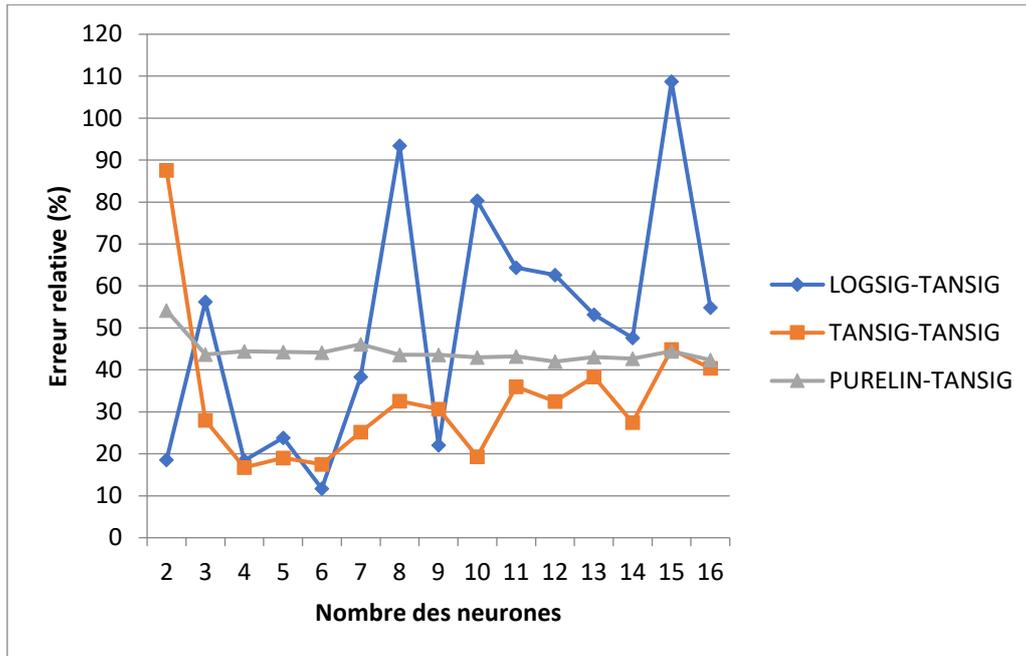


Figure IV.13. Erreur relative des trois configurations 2, 5 et 8

Interprétation

Dans la figure IV.13, les trois configurations sont les meilleurs en ce qui concerne la reconnaissance, on remarque que les erreurs relatives les plus bas (11 et 15%) sont données encore par les configurations 2 et 5 respectivement.

Conclusion

Les configurations 1, 4 et 7 donnent toujours des mauvaises performances que ce soit pour l'apprentissage ou pour la reconnaissance, donc ces configurations sont inutilisables dans notre type de problème.

Les deux configurations LOGSIG-TANGSIG et TANSIG-TANSIG donnent toujours des résultats satisfaisants en fonctions des neurones par couche caché, pour l'apprentissage ou pour la reconnaissance.

D'après les résultats obtenus dans le chapitre 3, la relation entre les entrées et la sortie n'est pas linéaire, donc les fonctions d'activations que nous allons choisir se sont **logsig** pour les couches cachées et **tansig** pour la sortie.

IV.4 Modèles proposés

Le tableau suivant représente les différentes configurations proposées en se basant sur le choix des paramètres mentionnées dans la section précédente.

Tableau IV.4 Les différentes configurations de réseau de neurones utilisées

Configuration	Fonction d'apprentissage	Type de RNA	Nombre de couches	Nombre de neurones
T1	Traingdx	BP	1	2 à 16
T2	Traingdx	Elman Bp	1	2à16
T3	Trainbr	BP	1	2à16
T4	Trainbr	Elman Bp	1	2à16
T5	Traingdx	BP	2	2à16 (CC1) - 2à16 (CC2)

Chaque configuration a une seule couche cachée comporte 15 modèles différents à simuler, et pour deux couches cachées comporte 225 modèles, en tous on a quatre configurations avec une seul couche cachée et une configuration avec deux couches cachées, donc nous avons simulé 285 modèles une fois avec le vecteur fréquentiel puis avec le vecteur temporel.

Pour choisir les bons modèles RNA il faut trouver le nombre de neurones optimal pour chaque couche, en outre, il y a pas une méthode directe pour choisir le nombre de neurones qui donnent des meilleur résultats donc pour se faire, nous allons effectuer plusieurs tests (comme indiqué dans la théorie chapitre II) pour chaque configuration en utilisant le vecteur fréquentiel puis le vecteur temporel, et à partir des résultats de ces tests nous allons comparer les performances des différents types de RNA et fonctions d'apprentissage en fonction des nombres de neurone, et à la fin nous allons désigner deux architectures finales de RNA.

IV.5 Les performances des configurations proposées

Les configurations citées dans le tableau précédents seront testées par les deux vecteurs, et les résultats obtenus vont nous aider à trouver le modèle le plus adéquat à nos problématiques, dans ce qui suit on va présenter les résultats de simulations de ces tests en utilisant logiciel MATLAB.

Il faut savoir que toutes les performances présentées ci-dessous sont obtenus après avoir répété le processus d'apprentissage 1000 fois pour chaque modele, puis après nous choisissons l'erreur la plus petite.

IV.5.1 Vecteur fréquentielle

On considère les entrées de notre réseau de neurones les caractéristiques fréquentielles du courant de fuite tirées précédemment, on va tracer les courbes de performances de chaque configuration on obtient les résultats suivants :

IV.5.1.1 Pour une couche cachée

1. Erreur d'entraînement

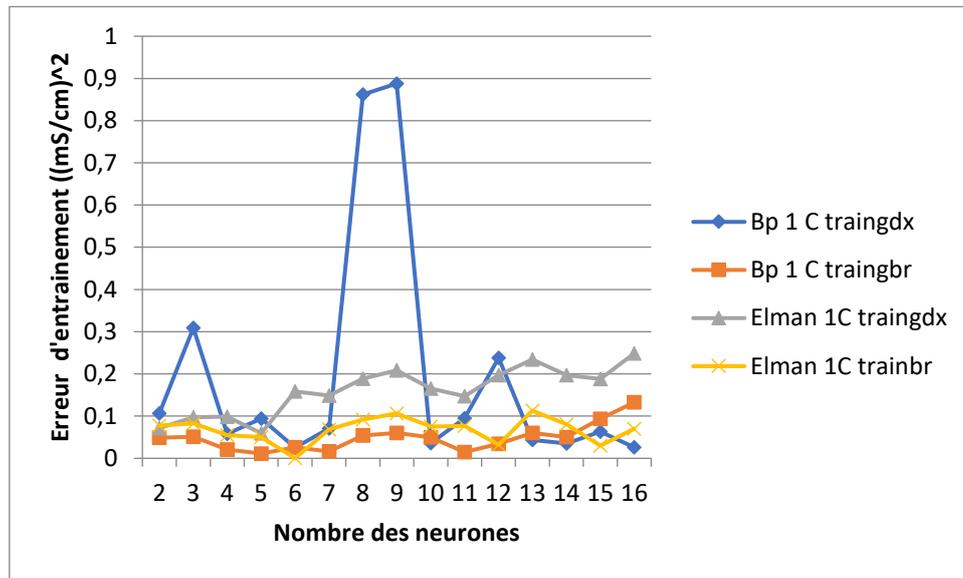


Figure IV.14. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle fréquentiel

2. Erreur de validation

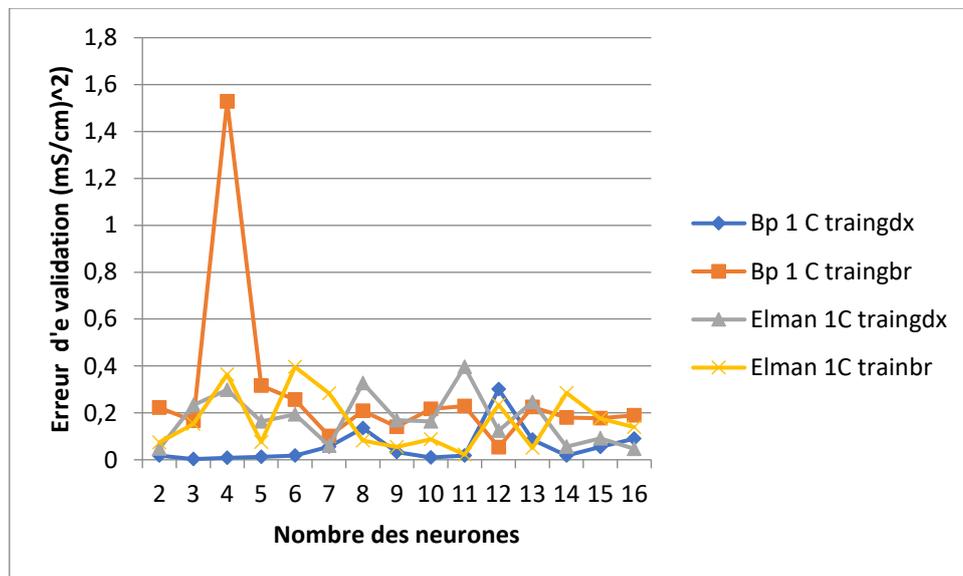


Figure IV.15. Erreur de validation en fonction du nombre de neurone par couche cachée modèle fréquentiel

3. Erreur de tests

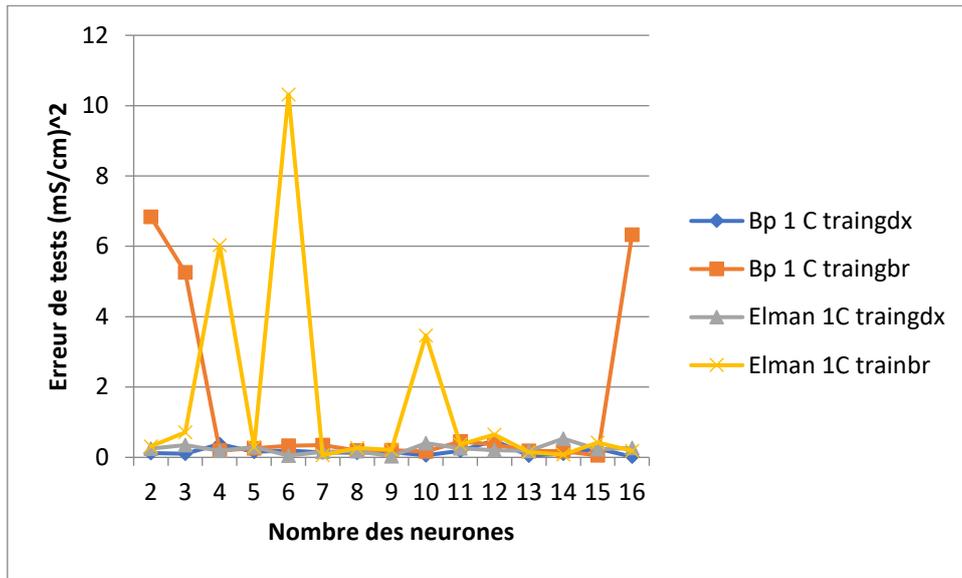


Figure IV.16. Erreur de test en fonction du nombre de neurone par couche cachée modèle fréquentiel

4. Erreur relative

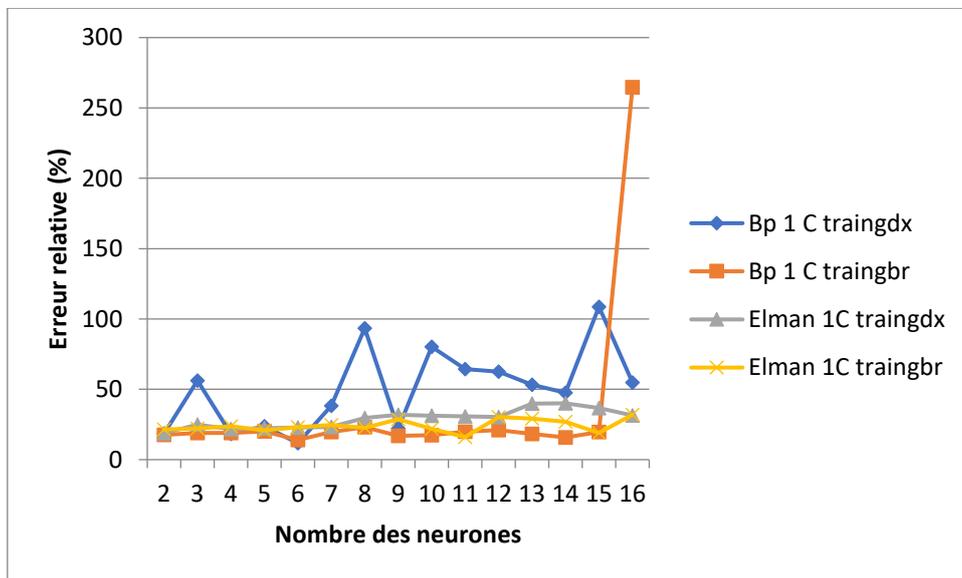


Figure IV.17. Erreur relative en fonction du nombre de neurone par couche cachée modèle fréquentiel

Interprétation

La configuration BP 1C trainbr possède une erreur d'entraînement acceptable compris entre (0.01 et 0.13 mS/cm²) pour presque tous les neurones par contre l'erreur de validation présente un pic jusqu'à 1.52 mS/cm² et l'erreur relative de cette configuration est de 13.97 % qui est une valeur acceptable mais qui reste supérieur à celle de BP 1C traingdx.

Les deux autres configurations Elman 1C possèdent des erreurs relatives supérieures à 20% qui est une valeur moins désirée par rapport aux résultats obtenus pour BP 1C.

IV.5.1.2 Pour deux couches cachées

Pour simuler la configuration BP 2C traingdx nous avons commencé par fixer le nombre de neurone dans la premier couche a 2 et en variant le nombre de neurone de la deuxième couche de 2 à 16, puis nous fixons 3 neurones dans la premier couche et en variant de 2 à 16 dans la deuxième...etc. jusqu' on arrive à 16 neurones dans la premier couche en variant de 2 à 16 les neurones dans la deuxième couche, par cela nous avons pu étudier tous les cas possibles d'un RNA a deux couches cachées avec une variation de 2 à 16 neurones.

Nous avons simulé 225 modèles et les résultats obtenus représenté dans les figures de IV.18 jusqu IV.25

1. Erreur d'entrainement

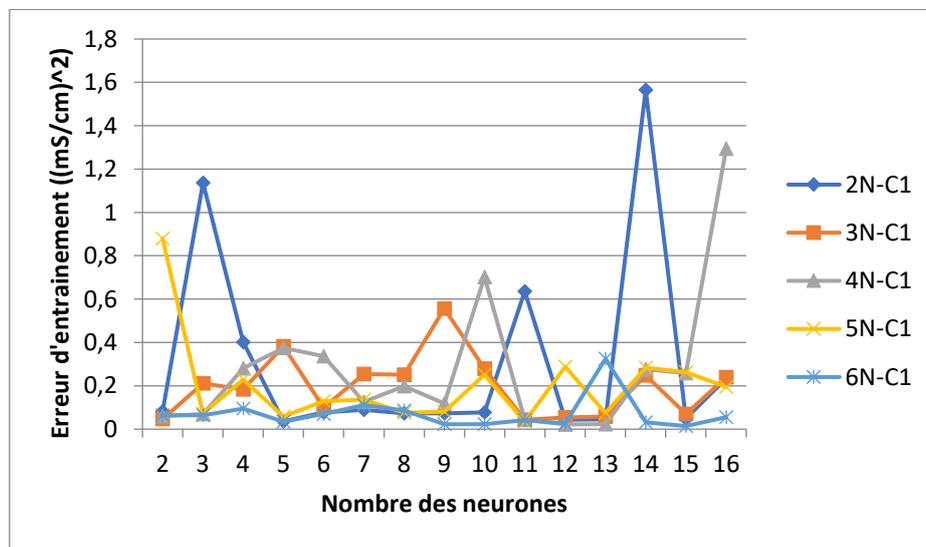


Figure IV.18. Erreur d'entrainement en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 a 6N-C1)

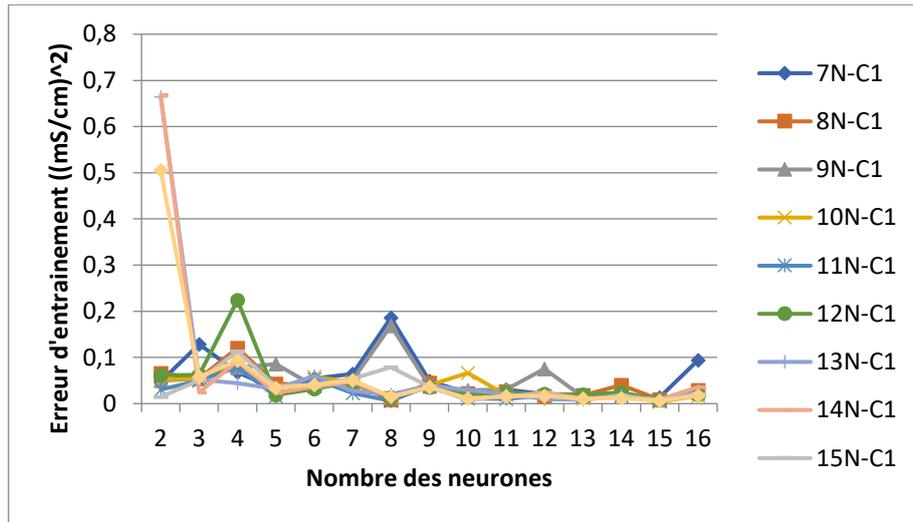


Figure IV.19. Erreur d'entraînement en fonction du nombre de neurone par couche cachée modèle fréquentiel (7N-C1 à 16N-C1)

2. Erreur de validation

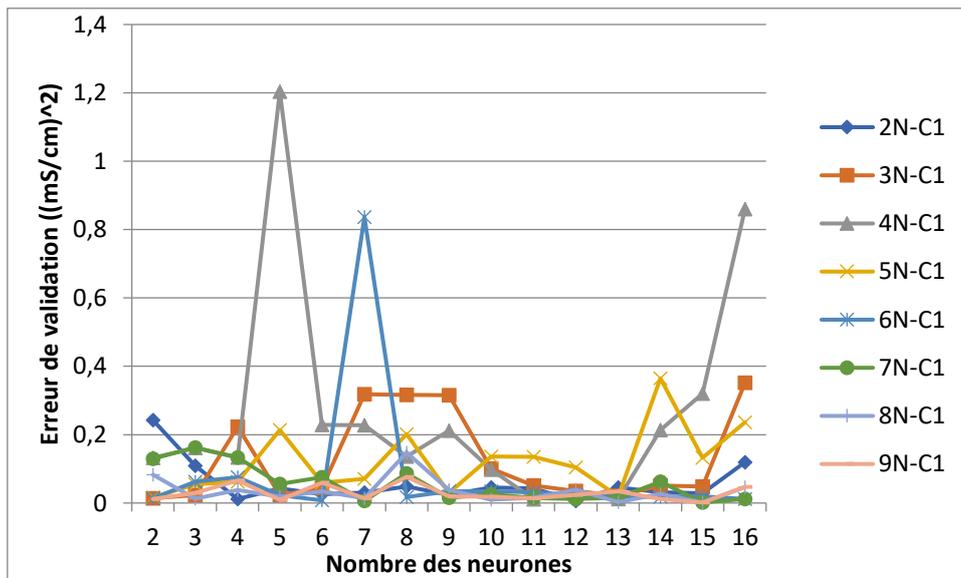


Figure IV.20. Erreur de validation en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 à 9N-C1)

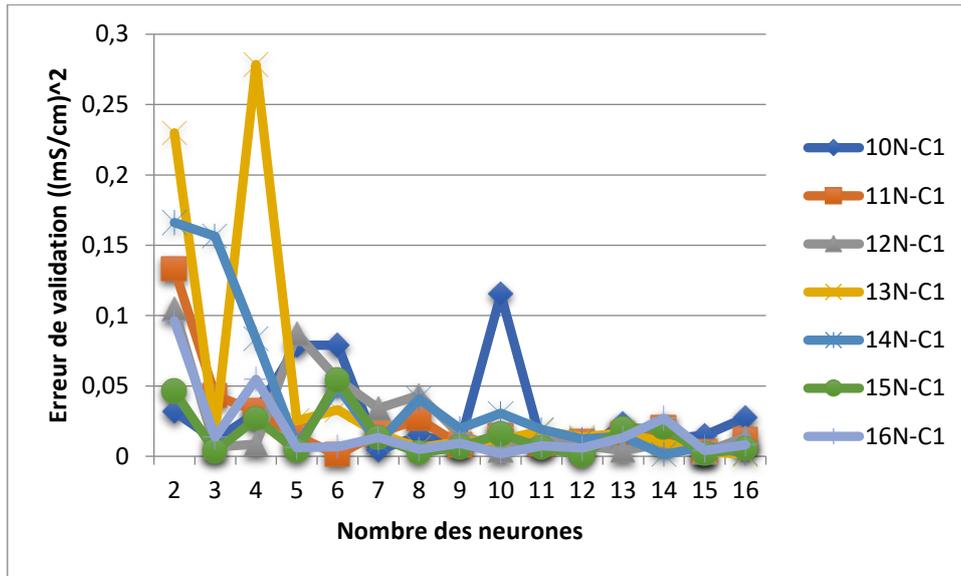


Figure IV.21. Erreur de validation en fonction du nombre de neurone par couche cachée modèle fréquentiel (10N-C1 à 16N-C1)

3. Erreur de tests

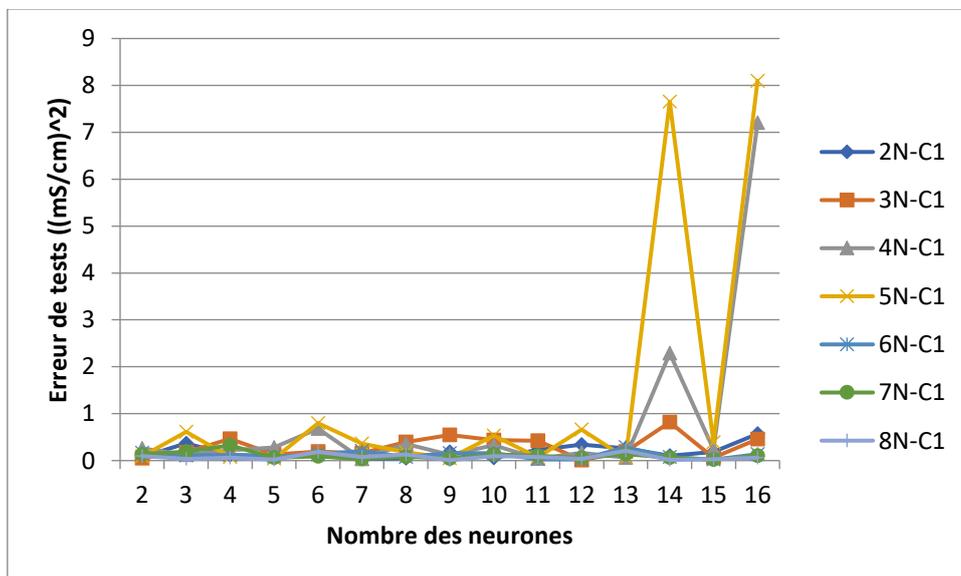


Figure IV.22. Erreur de test en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 à 8N-C1)

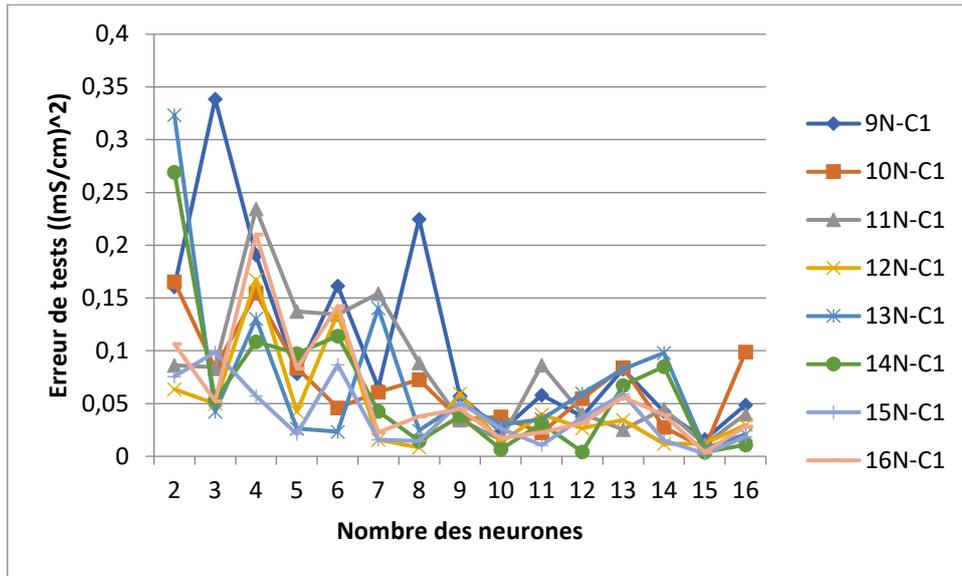


Figure IV.23. Erreur de test en fonction du nombre de neurone par couche cachée modèle fréquentiel (9N-C1 à 16N-C1)

4. Erreur relative

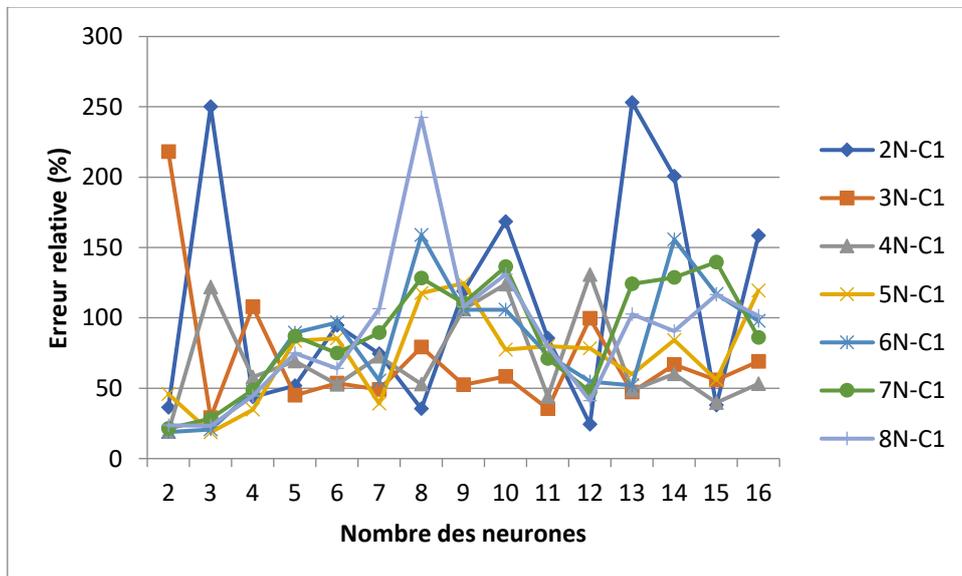


Figure IV.24. Erreur relative en fonction du nombre de neurone par couche cachée modèle fréquentiel (2N-C1 à 8N-C1)

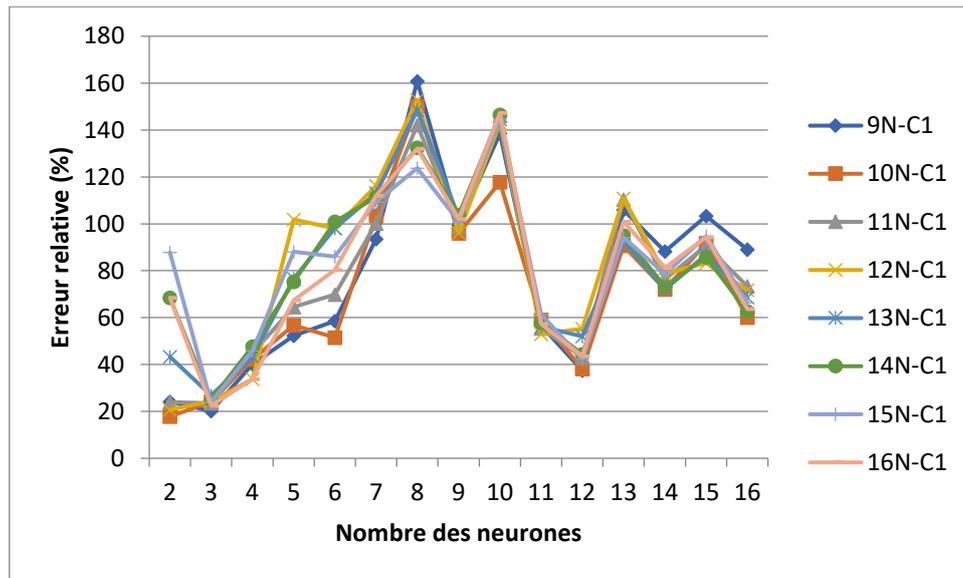


Figure IV.25. Erreur relative en fonction du nombre de neurone par couche cachée modèle fréquentiel (9N-C1 à 16N-C1)

Interprétation

A partir des figures de IV.18 à IV.23 les erreurs d'apprentissage, de validation et de test diminuent plus que le nombre de neurone dans la première et la deuxième couches augmente (atteint une erreur inférieure à 0.1 mS/cm²).

D'autre part, dans les figures IV.24 et IV.25 l'erreur relative augmente plus que le nombre de neurones dans la 2^{ème} couche augmente, c'est à dire des bonnes performances (de l'ordre de 20%) pour un nombre de neurone minimum dans la 2^{ème} couche.

Dans la figure IV.25 on remarque que à partir d'un nombre de neurone égale à 9 dans la première couche, les variations de l'erreur relative est presque identique pour presque tous les modèles.

IV.5.2 Vecteur temporel

Maintenant nous allons utiliser les caractéristiques temporelles pour refaire les mêmes tests.

IV.5.2.1 Une seule couche cachée

1. Erreur d'entraînement

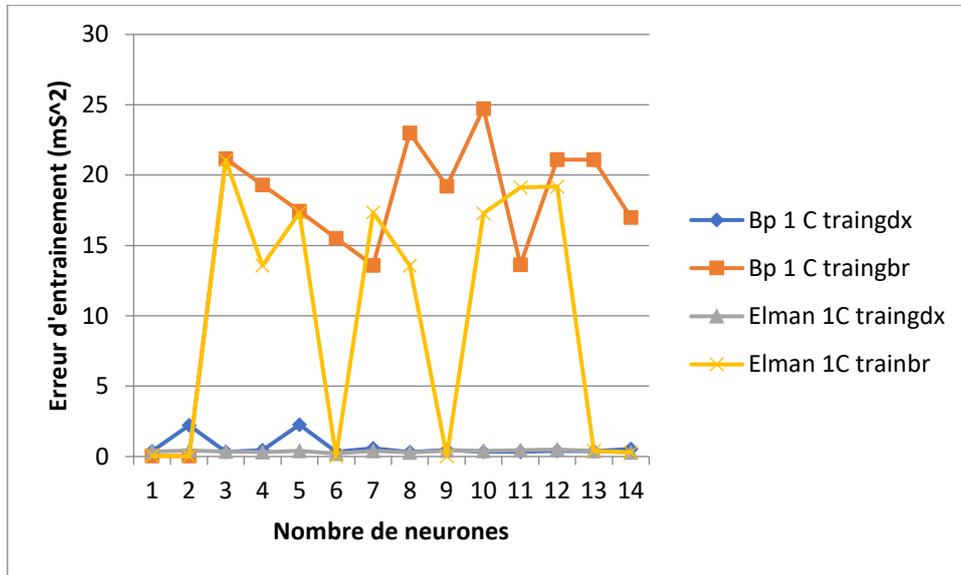


Figure IV.26. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel

2. Erreur de validation

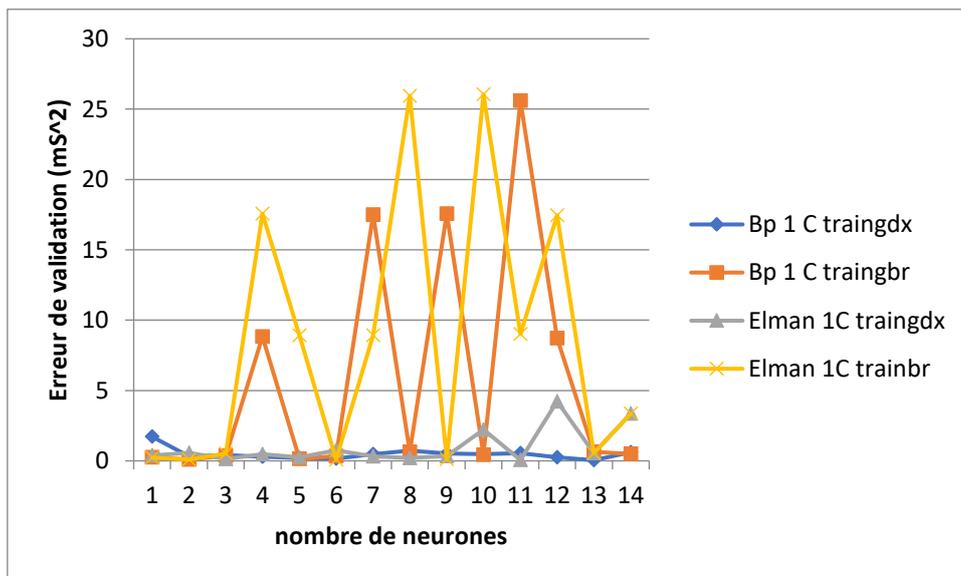


Figure IV.27. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel

3. Erreur de tests

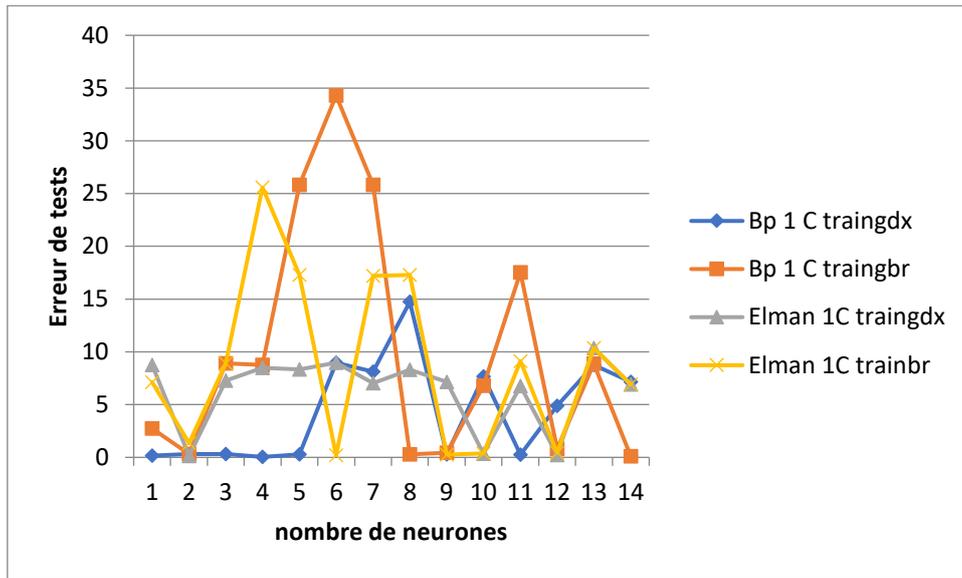


Figure IV.28. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel

4. Erreur relative

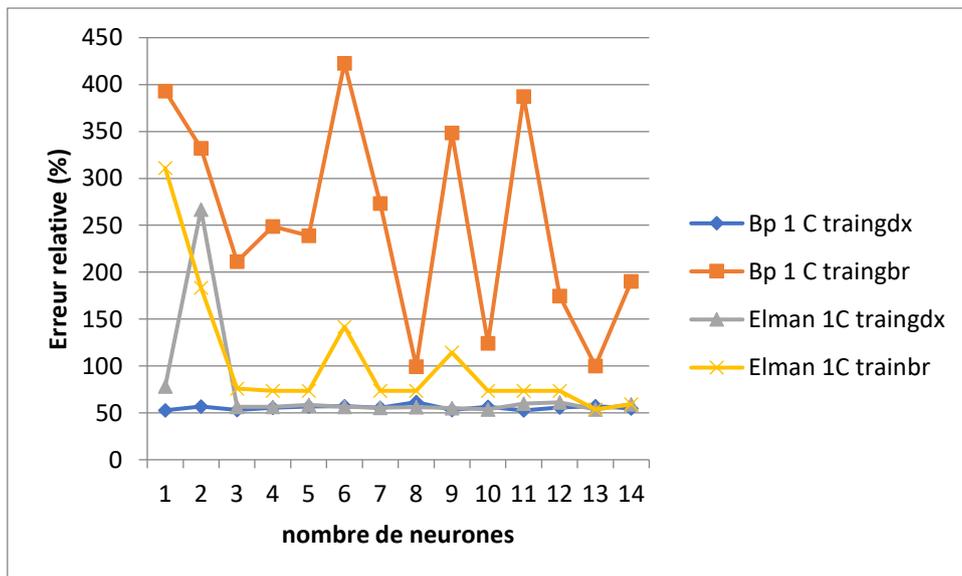


Figure IV.29. Erreur relative en fonction du nombre de neurone par couche cachée du modèle temporel

Interprétation

Pour les erreurs d'entraînement et de validation il y a que les configuration Elman 1C traingdx et BP 1C traingdx qui possèdent des valeurs acceptables comprise entre (0.20 et 0.45 mS/cm²) pour la majorité des neurones, par contre l'évolution de l'erreur de test pour ces deux configurations et moins symétrique en fonction des neurones et atteint des valeurs jusqu'à 15 mS/cm² à partir du nombre de neurone dans couche cachée égale à 5.

Les configurations BP 1C trainbr et Elman 1C trainbr atteignent des erreurs d'apprentissage, de validation et de test très importantes (35 mS/cm²) et présentent des changements brusques en fonction de nombre des neurones.

D'autre part l'erreur relative de toutes les configurations est supérieure à 50 % qui est une précision très grande et insuffisante pour pouvoir prédire d'une façon correcte à partir des valeurs hors de l'ensemble d'apprentissage.

IV.5.2.2 Pour deux couches cachées

Nous avons simulé 225 modèles avec le vecteur temporel et les résultats obtenus représenté dans les figures de IV.30 jusqu'à IV.37.

1. Erreur d'entraînement

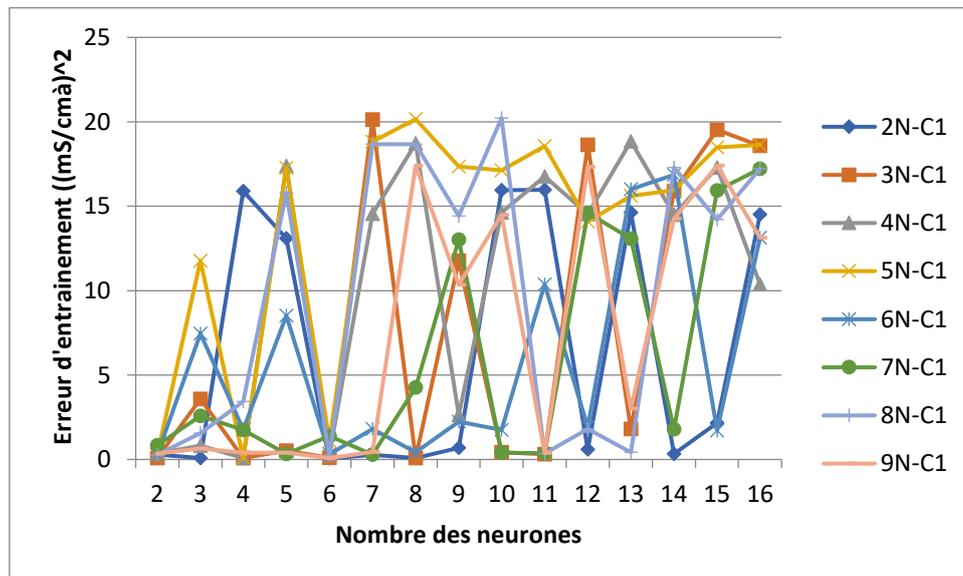


Figure IV.30. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 à 9N-C1)

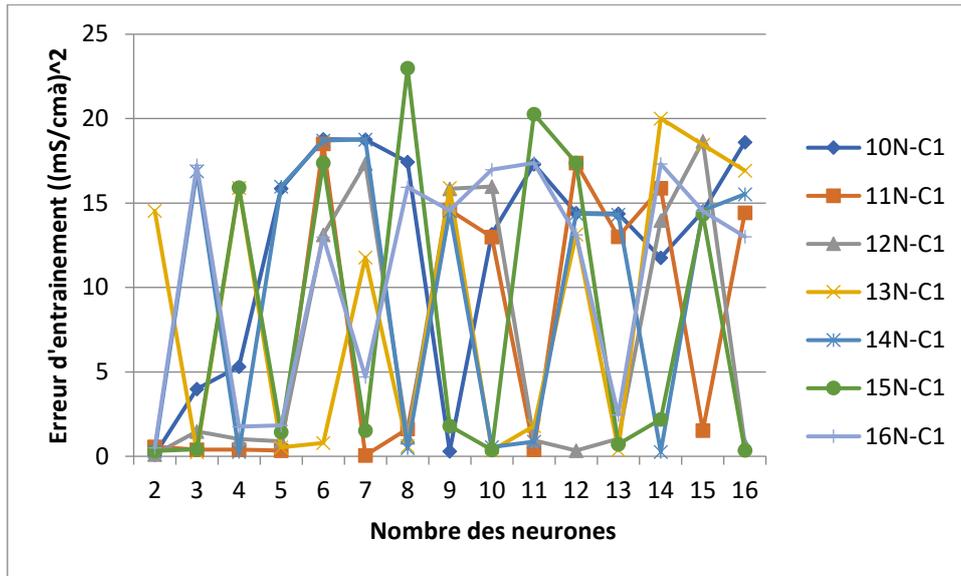


Figure IV.31. Erreur d'entraînement en fonction du nombre de neurone par couche cachée du modèle temporel (10N-C1 à 16N-C1)

2. Erreur de validation

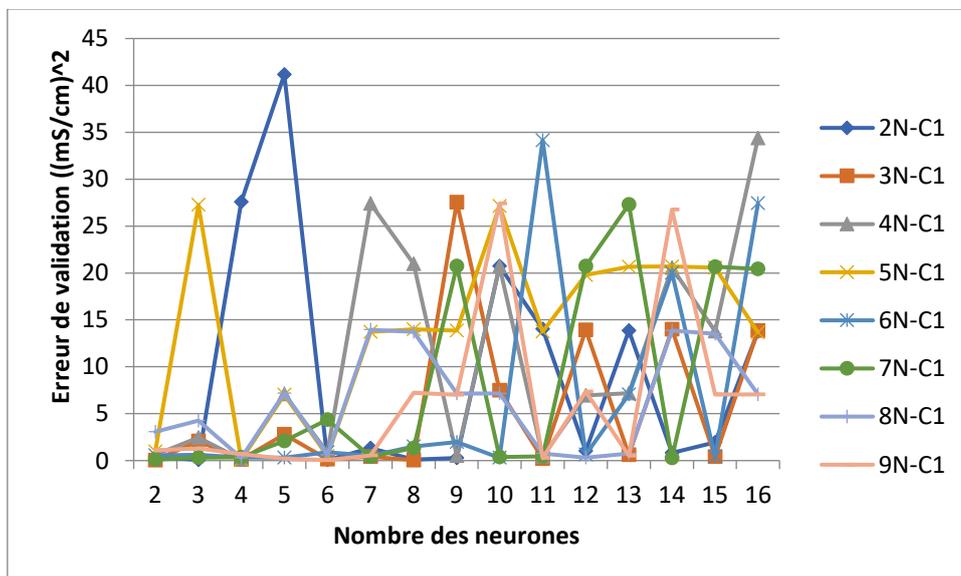


Figure IV.32. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 à 9N-C1)

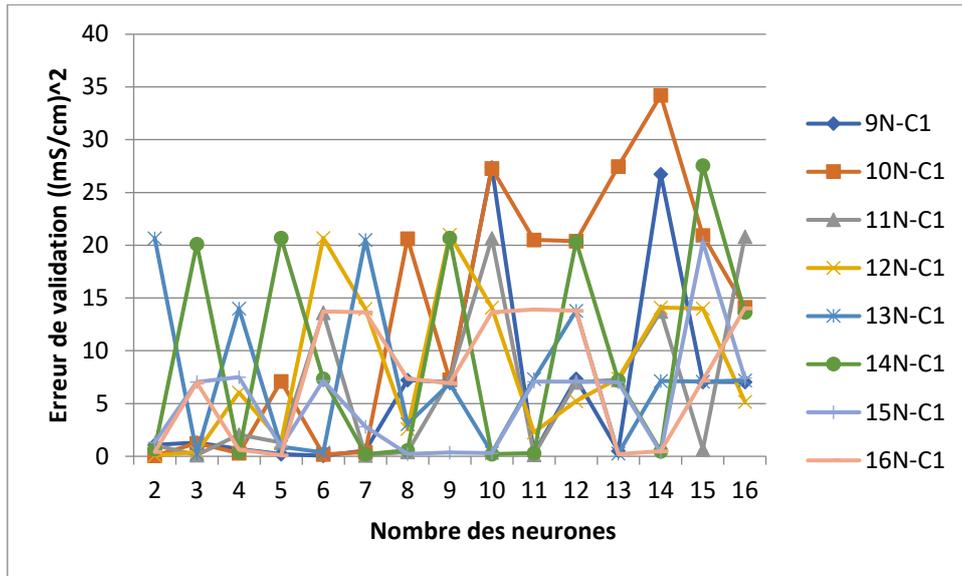


Figure IV.33. Erreur de validation en fonction du nombre de neurone par couche cachée du modèle temporel (9N-C1 à 16N-C1)

3. Erreur de tests

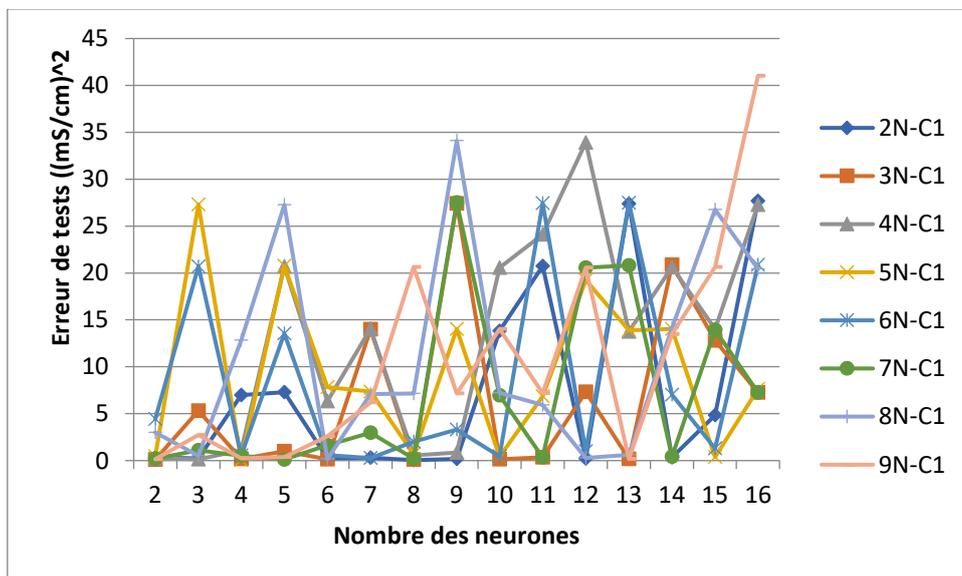


Figure IV.34. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 à 9N-C1)

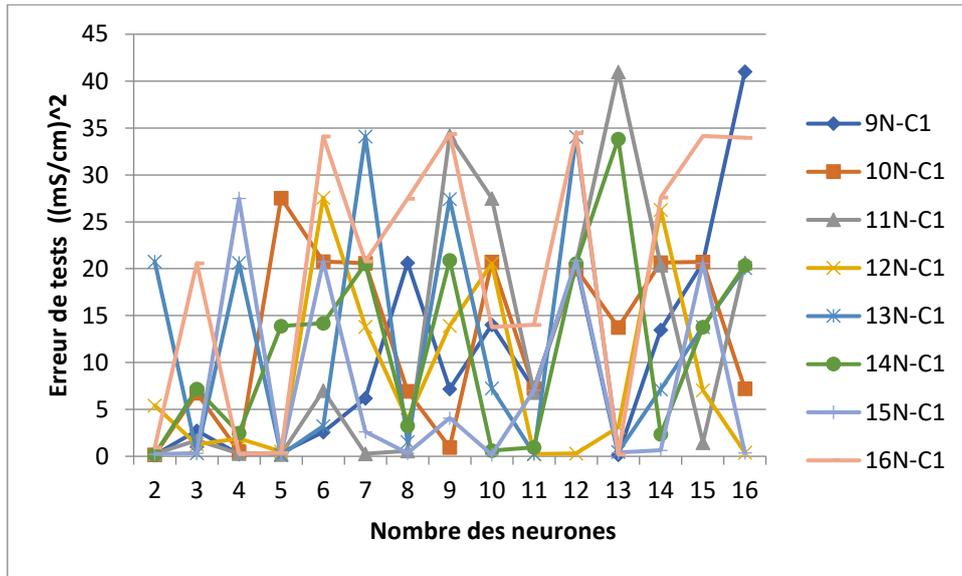


Figure IV.35. Erreur de test en fonction du nombre de neurone par couche cachée du modèle temporel (9N-C1 à 16N-C1)

4. Erreur relative

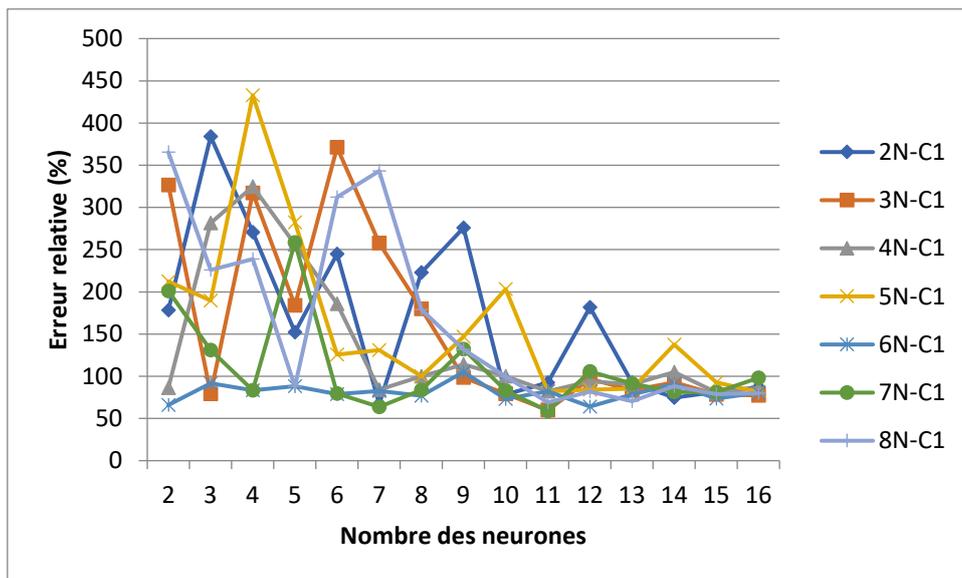


Figure IV.36. Erreur relative en fonction du nombre de neurone par couche cachée du modèle temporel (2N-C1 à 8N-C1)

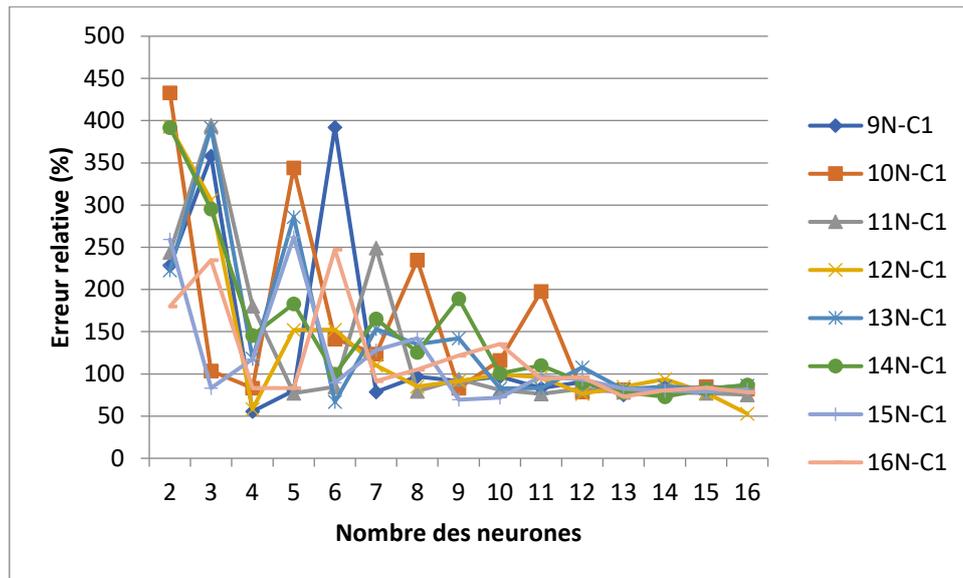


Figure IV.37. Erreur relative en fonction du nombre de neurone par couche cachée du modèle temporel (9N-C1 à 16N-C1)

Interprétation

Les résultats de simulation de la configuration BP 2C traingdx avec le vecteur temporel présentent aucun comportement symétrique des erreurs, aucune convergence n'est observée en changeant le nombre, en effet, nous avons noté aucun modèle cohérent.

IV.6 Les modèles choisis

A partir de l'analyse des résultats obtenus dans la simulation des différents modèles, on procède maintenant à choisir que deux modèles qui donnent une meilleure performance d'apprentissage et surtout un taux de reconnaissance proche de 100%.

IV.6.1 Pour le vecteur fréquentiel

A partir des résultats des tests effectués dans la partie précédente, le meilleur modèle qui donne une erreur relative de prédiction minimale, c'est un modèle tiré de la configuration T1 qui représente le type feedforward Backpropagation avec la fonction d'apprentissage **traingdx**, avec une erreur relative de généralisation de 11,5%.

Tableau IV.5. Paramètres finaux du modèle RNA.

Type du réseau	Feedforward Backpropagation
Fonction d'apprentissage	traingdx
Fonction de cout	MSE
Fonction d'activation	Logsig-Tansig
Nombre de couche cachée	1
Nombre de neurone par couche cachée	6

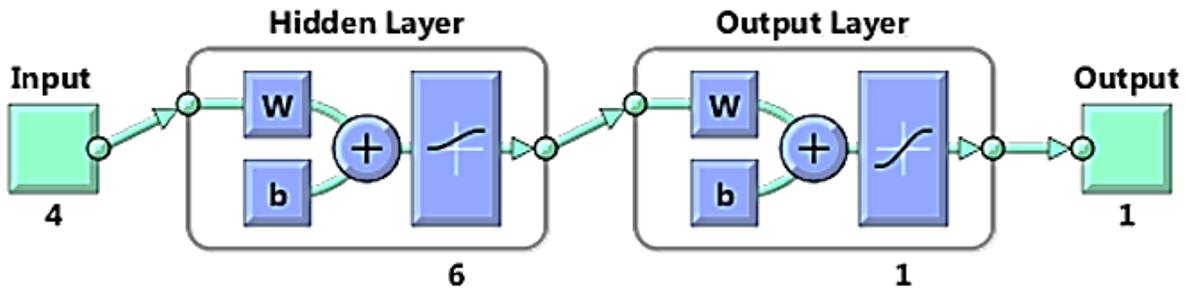


Figure IV.38 Structure du modèle fréquentielle.

IV.6.2 Pour le vecteur temporel

Le modèle qui donne une erreur relative minimal est tiré de la configuration T5 Elman BP traingdx avec deux couches cachées, avec une erreur relative de généralisation de 23.67%.

Tableau IV.6. Paramètres finaux du modèle RNA.

Type du réseau	Elman Backpropagation
Fonction d'apprentissage	traingdx
Fonction de cout	MSE
Fonction d'activation	Losig-Logsig-Tansig
Nombre de couche cachée	2
Nombre de neurone par couche cachée	04-mars

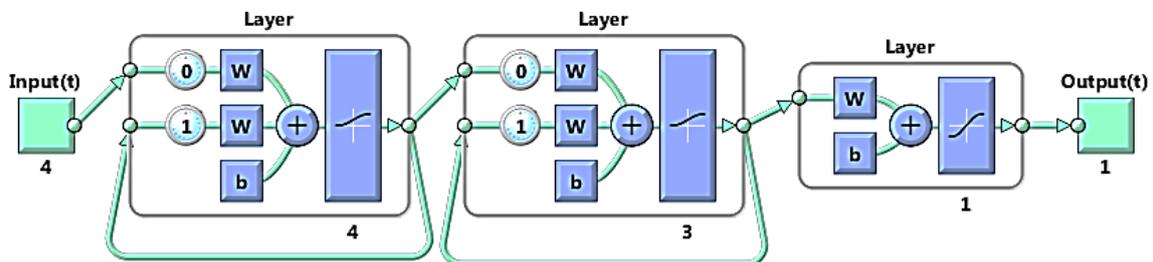


Figure IV.39. Structure du modèle temporelle

IV.7 Résultats

IV.7.1 Résultats du modèle fréquentiel

IV.7.1.1 Performance du modèle

Nous avons évalué les performances du modèle choisi par :

1. La fonction d'erreur MSE

La figure suivante donne les variations des trois performances en fonction des itérations effectués.

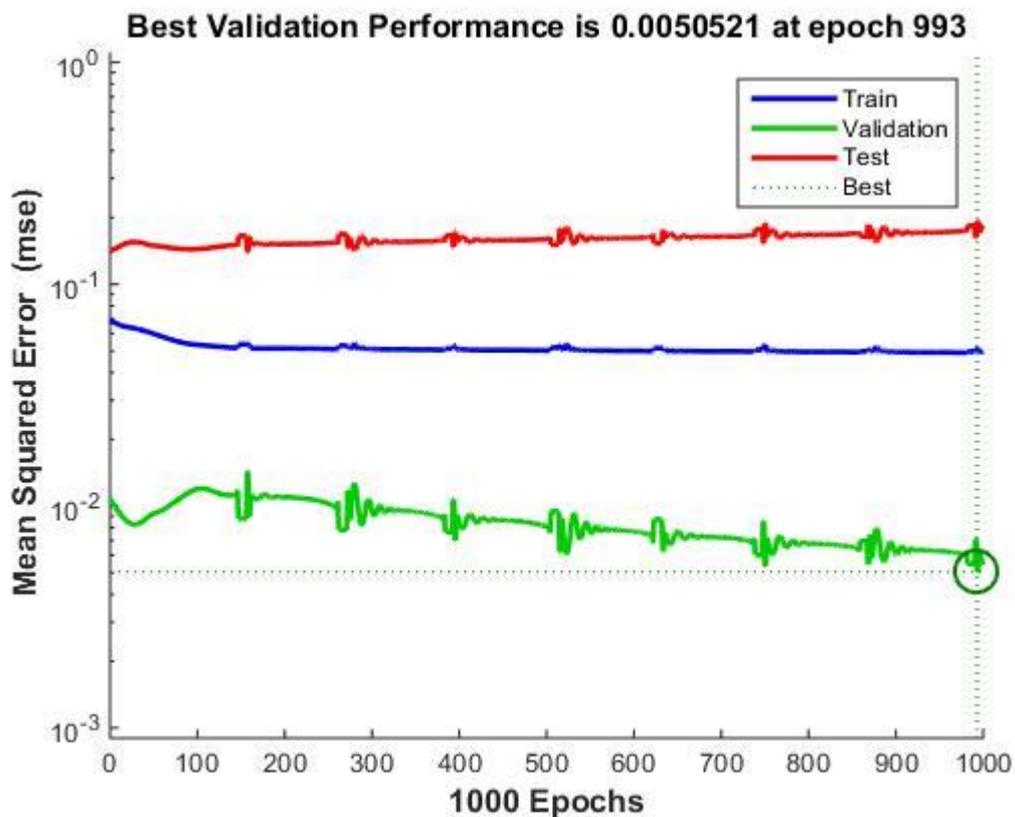


Figure IV.40. Evaluation de la fonction du cout du modèle fréquentiel en fonction du nombre d'itération.

L'évolution de l'erreur d'apprentissage est presque constante aux alentours de 10^{-1} pendant que l'erreur de validation converge doucement vers une valeur minimale de l'ordre de 0.0050521 à l'itération 993.

D'autre part la précision de l'ensemble des tests est de l'ordre de 10^{-1} ce qui veut dire que notre modèle prédit d'une façon correcte même avec des valeurs qui n'a jamais vu pendant le processus d'apprentissage.

2. Régression linéaire

Dans ce qui suit nous allons montrer la corrélation entre les valeurs prédits (conductivité) de chaque sous-ensemble et les cibles désirées.

- **Sous-ensembles d'apprentissage et de validation**

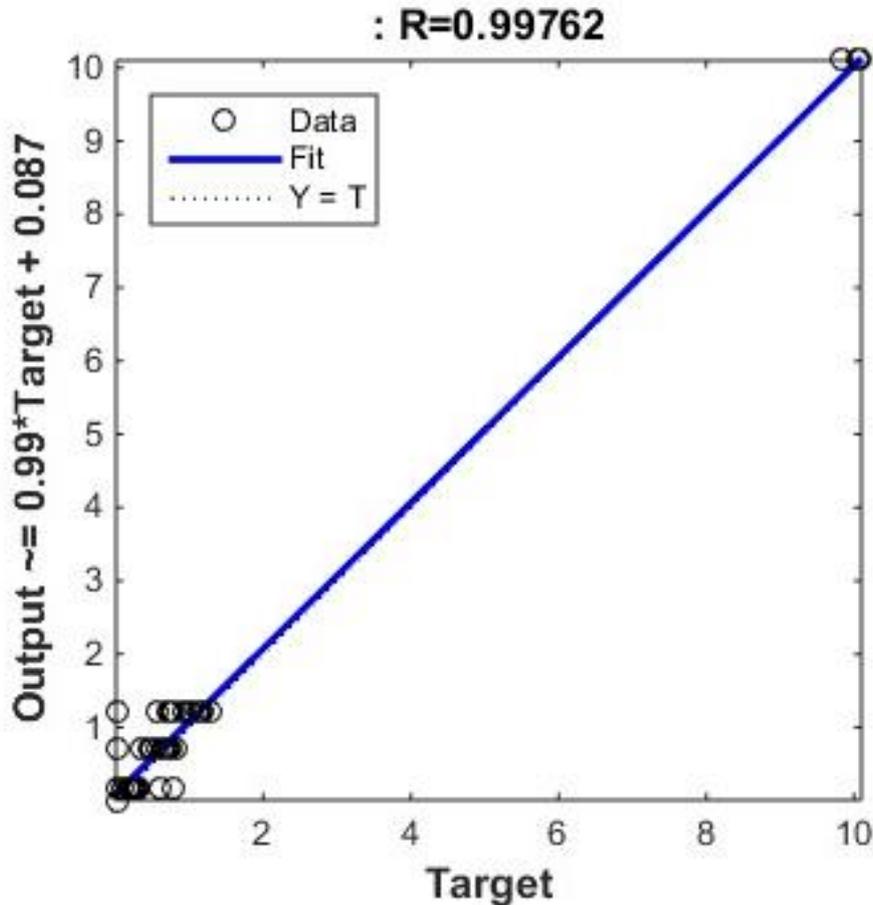


Figure IV.41 Performance du modèle fréquentiel par la méthode de régression linéaire

On a :

$$\text{Output} = 0.99 * \text{Target} + 0.087 \quad (\text{IV.3})$$

Avec : $a = 0.99$ étant très proche de 1 et $b = 0.087$ étant proche de 0, ce qui veut dire que la droite de régression tend vers la droite discontinue qui représente $\text{Output} = \text{Target}$.

Or que, $R = 0.99762$ étant très proche de 1 qui nous montre encore que la corrélation entre les sorties prédits et les cibles désirées est très bonne dans les deux sous-ensembles.

- **Sous-ensemble de test de reconnaissance**

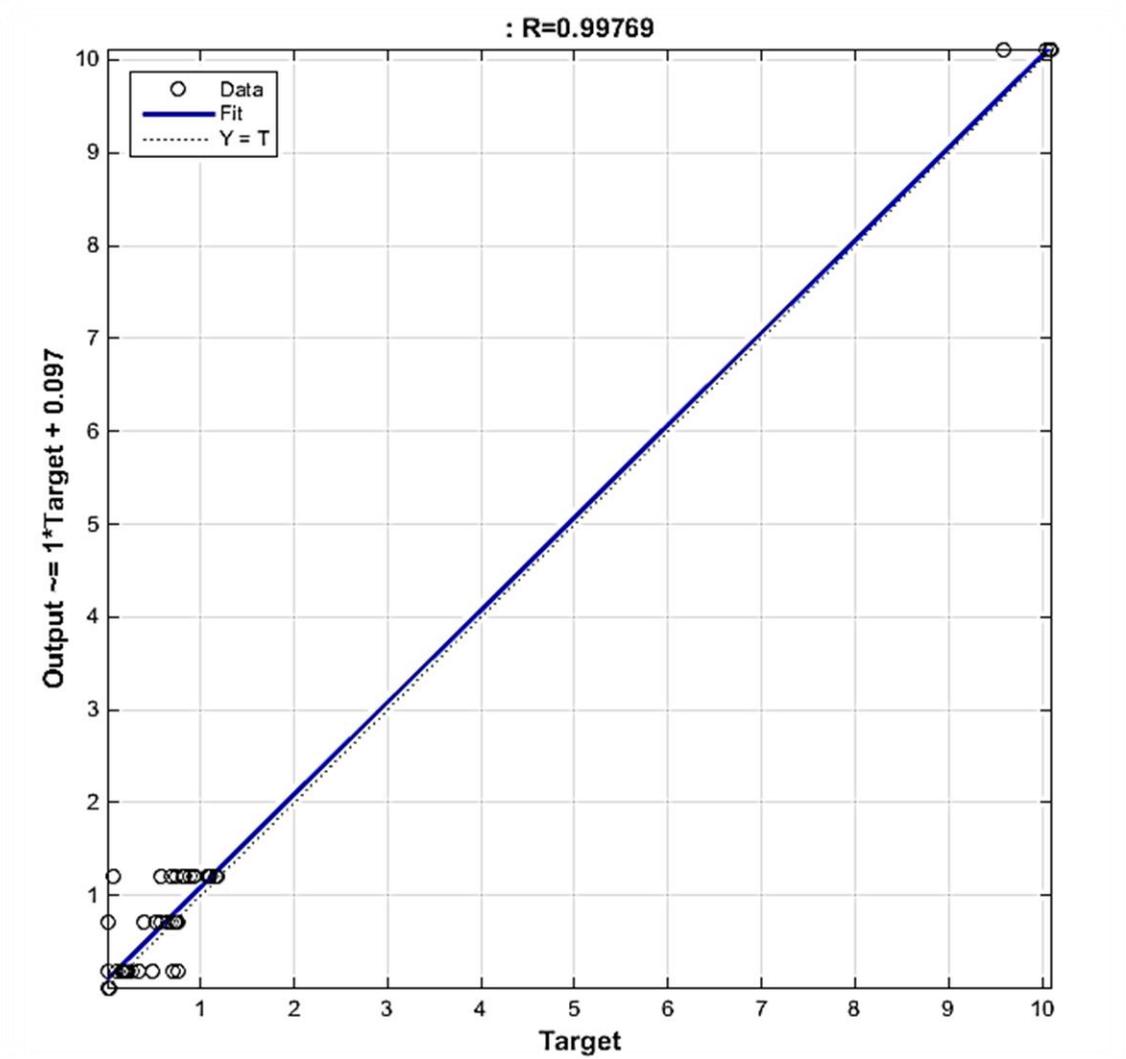


Figure IV.42 Performance du modèle fréquentiel par la méthode de régression linéaire du sous-ensemble de test.

On a:

$$\text{Output} = 1 * \text{Target} + 0.097 \quad (\text{IV.4})$$

Avec: a = 1

$$b = 0.097$$

R = 0.99769 étant très proche de 1.

A la lumière de ces résultats, nous pouvons juger qu'il y a une bonne corrélation aussi entre les sorties désirées (ciblées) les sorties du réseau de neurone dans l'ensemble de test.

D'autre part, à partir des deux figures IV.41 et IV.42, la performance du RNA est traduite par la concentration de la plupart des cercles autour de 0.01, 0.19, 0.71, 1.2 et 10.1 mS/cm² dans chaque sous-ensemble.

IV.7.1.2 Résultats de prédiction du modèle fréquentiel

Une fois le réseau de neurones est créé, des tests seront effectués afin de vérifier la qualité des prévisions du modèle choisi en lui présentant des nouveaux exemples d'entrées (sous-ensemble de test) qui ne font pas partie de l'ensemble d'apprentissage pour qu'il calcul les sorties correspondantes.

Ce tableau est composé les échantillons de test qui n'ont été pas utilisé durant le processus d'apprentissage, et les erreurs relatives et absolus de la prédiction utilisant le modèle fréquentiel du RNA :

Tableau IV.7 Erreur relative en (%) de prédiction obtenue par le modèle (RNA) choisi pour le vecteur fréquentiel

Indice	$\phi(^{\circ})$	3h/f(%)	5h/f(%)	V (kVeff)	Conductivité réelle (mS/cm)	Conductivité prédite (mS/cm)	Erreur Absolue(mS/cm)	Erreur relative (%)
1	0	14,48	7,24	3	1,2	1,98	0,784072	65,339
2	1	19,48	2,48	6	1,2	1,26	0,064	5,4154
3	4,3	26,25	5,03	9	1,2	1,17	0,023	1,982
4	5,9	26,61	2,94	12	1,2	1,109	0,09	7,5
5	5,5	30,49	3,88	15	1,2	1,106	0,0931	7,766
6	13,3	21,17	2,58	3	10,1	10,09	2,26E-06	0,000022
7	23,4	22,31	4,4	6	10,1	10,09	2,27E-06	0,00002
8	23,1	30,11	10,38	9	10,1	10,099	0,000113	0,00112
9	14,8	22,56	9,62	12	10,1	9,66	0,43	4,329
10	4,1	0,9	11,08	3	0,01	0,011	0,00191	19,175
11	5,1	1,9	16,44	6	0,01	0,01	1,98E-06	0,019
12	5	1,06	18,29	9	0,01	0,01	1,89E-07	0,0018
13	7,9	3,71	18,26	12	0,01	0,01	1,33E-06	0,013
14	7	2,65	18,86	15	0,01	0,01	6,57E-06	0,065
15	6,1	8,71	24,44	20	0,01	0,01	8,34E-06	0,083
16	6,9	10,54	30,7	25	0,01	0,01	1,3E-07	0,0013
17	0,6	4,68	11,71	6	0,19	0,17	0,029	15,748
18	4,3	22,01	18,91	9	0,19	0,079	0,11	58,26
19	0,3	17,42	22,15	12	0,19	0,189	0,037	11,49
20	0,2	4,8	8,14	3	0,71	0,49	0,21	30,77
21	3	18,61	10,13	9	0,71	0,708	0,0627	8,83
22	1,4	29,55	17,12	12	0,71	0,702	0,099	10,05
23	3,6	26,67	12,29	15	0,71	0,705	0,12	9,97

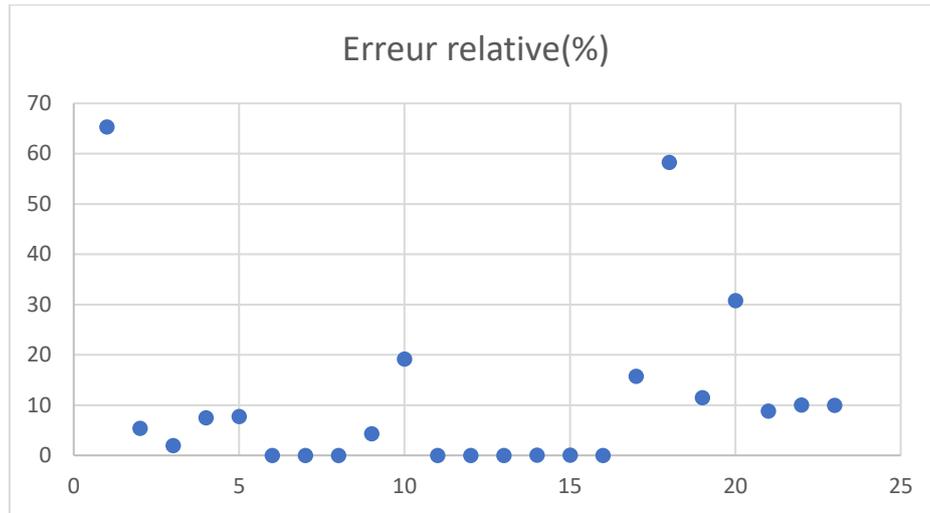


Figure IV.43 Représentation de l'erreur relative du modèle fréquentiel.

Il est à noter qu'il est bien clair du tableau précédent que l'erreur relative des prédictions se diffère d'un échantillon à un autre.

A partir de la figure IV.43, pour la majorité des échantillons l'erreur relative ne dépasse pas 10%.

Le modèle de RNA choisi donne des résultats satisfaisants pour la plupart des échantillons réservés au test (23 échantillons), cependant, il existe des valeurs des conductivités prédits en utilisant notre modèle, sont loin des valeurs réelles (cibles) comme les échantillons 1 et 18 (soulignée en gras) avec des erreurs relatives de 65% et 58% respectivement, ça revient aux conditions des mesures.

IV.7.2 Résultats du modèle temporel

IV.7.2.1 Performance du modèle

1. La fonction d'erreur MSE

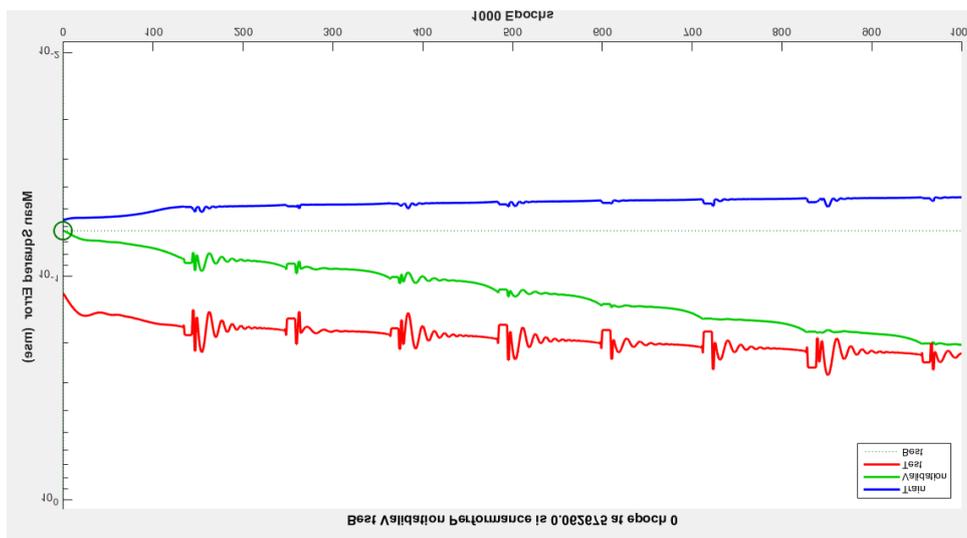


Figure IV.44. Évaluation de la fonction cout du modèle temporel en fonction du nombre d'itérations.

A partir de la figure IV.44 l'erreur d'apprentissage converge vers une valeur de 0.05 pendant que l'erreur de validation augmente en fonction du nombre d'itération mais en gardant toujours des valeurs faibles de l'ordre de 10^{-1} .

D'autre part l'erreur de test reste constante aux alentours de 10^{-1} , de ce fait on peut dire que le modèle choisi est bien généralisé.

2. Régression linéaire

- Sous-ensemble d'apprentissage et de validation

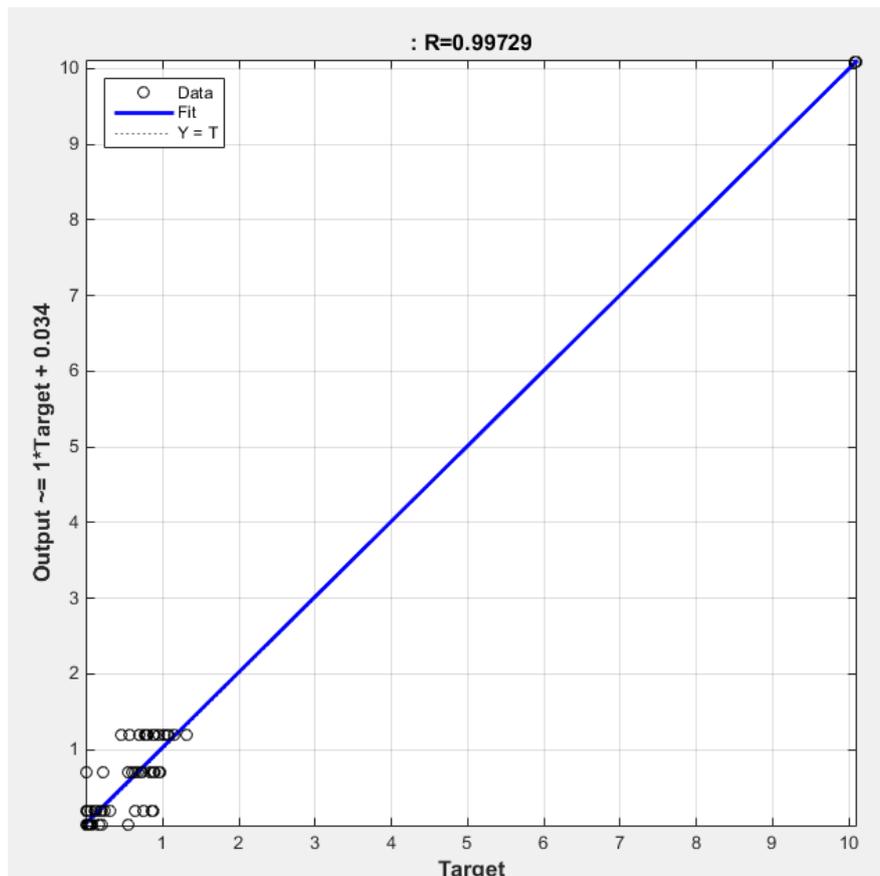


Figure IV.45 Performance du modèle temporel par la méthode de régression linéaire.

On a :

$$\text{Output} = 1 * \text{Target} + 0.034 \quad (\text{IV.5})$$

Avec : $a = 1$ et $b = 0.034$ étant proche de 0, ce qui veut dire que la droite de régression s'aligne presque parfaitement sur la droite discontinue qui représente $\text{Output} = \text{Target}$.

Or que, $R = 0.99729$ étant très proche de 1 ce qui veut dire que la corrélation entre les sorties prédits et les cibles désirées est très bonne dans les deux sous-ensembles.

- **Sous-ensemble de test de reconnaissance**

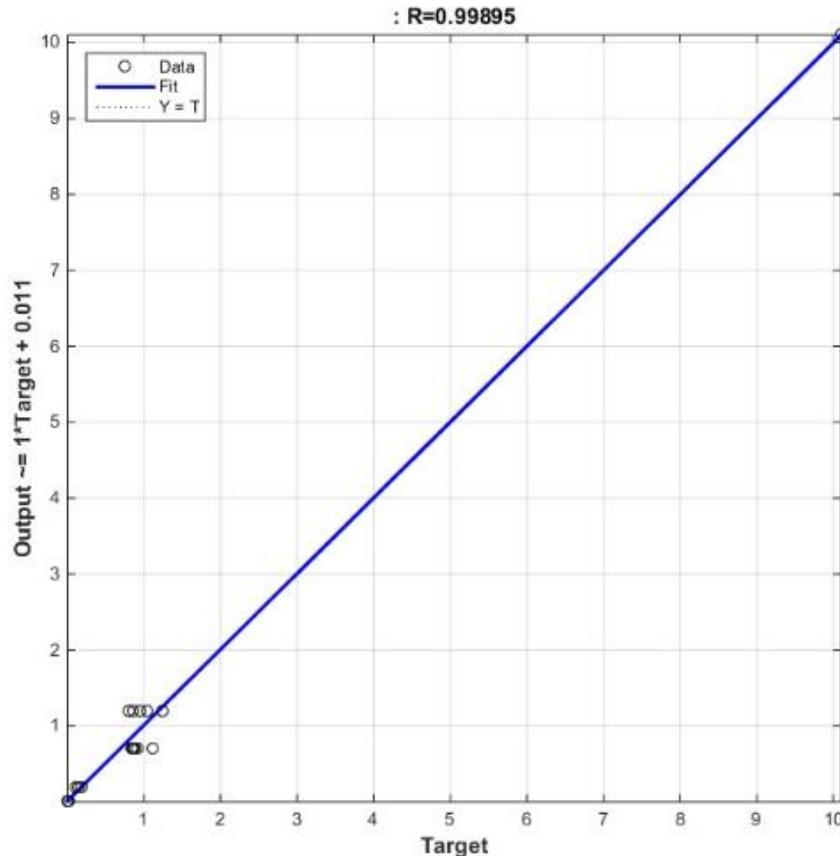


Figure IV.46 Performance du modèle temporel par la méthode de régression linéaire du sous-ensemble de test.

On a:

$$\text{Output} = 1 * \text{Target} + 0.097 \quad (\text{IV.6})$$

Avec: $a = 1$

$b = 0.011$ étant très proche de 1.

$R = 0.99769$ étant très proche de 1.

A partir de l'équation de la droite de régression et le coefficient de corrélation on en déduit que qu'il y a une bonne corrélation entre les sorties désirées (ciblées) les sorties du réseau de neurone dans l'ensemble de test.

D'autre part, à partir des deux figures IV.45 et IV.46, la performance du RNA est traduite par la concentration de la plupart des cercles autour de 0.01, 0.19, 0.71, 1.2 et 10.1 mS/cm² dans tous les sous-ensembles.

IV.7.2.2 Résultats de prédiction du modèle temporel

Une fois le réseau de neurones est créé, des tests seront effectués afin de vérifier la qualité des prévisions du modèle choisi en lui présentant des nouveaux exemples d'entrées (sous-ensemble de test) qui ne font pas partie de l'ensemble d'apprentissage pour qu'il calcul les sorties correspondantes.

Ce tableau est composé les échantillons de test qui n'ont été pas utilisé durant le processus d'apprentissage, et les erreurs relatives et absolus de la prédiction utilisant le modèle temporel du RNA.

Tableau IV.8 Erreur relative en (%) de prédiction obtenue par le modèle (RNA) choisi pour le vecteur temporel

Indice	Tension appliquée (Kv)	I _{max} (mA)	RMS(mA)	STD (mA)	Conductivité réelle (mS/cm)	Conductivité prédite (mS/cm)	Erreur Absolue (mS/cm)	Erreur relative (%)
1	3	8,2	4,326799	4,3233	10,1	10,099	0,00044	0,004358
2	6	9,4	4,074206	4,0567	10,1	10,099	0,00042	0,0041
3	9	24,6	5,17847	5,1167	10,1	10,09	0,00015	0,0015
4	12	20	6,776	6,73	10,1	10,099	0,00013	0,0013
5	3	1,56	0,90724	0,907	1,2	1,245	0,045	3,8
6	6	5	2,82507	2,8205	1,2	1,039	0,16	13,37
7	9	4,28	1,907271	1,90726	1,2	0,797	0,4	33,5
8	12	3,3	1,310796	1,37096	1,2	0,95	0,33	20,24
9	15	4,2	1,926426	1,92343	1,2	0,95	0,24	20,82
10	3	1,06	0,6964	0,6954	0,71	1,12	0,41	58,26
11	6	4,44	2,5443	2,5443	0,71	0,83	0,12	17,68
12	9	2,6	1,2276	1,2277	0,71	0,86	0,15	22,41
13	12	4,5	1,8033	1,8029	0,71	0,85	0,14	20,88
14	15	4,68	2,1246	2,1234	0,71	0,915	0,2	20,93
15	3	0,188	0,1177	0,1176	0,19	0,148	0,04	21,68
16	6	0,7424	0,32954	0,32931	0,19	0,103	0,08	45,59
17	12	2,225	1,409141	1,409142	0,19	0,18	0,004	2,22
18	15	3,24	1,8394	1,8385	0,19	0,19	0,0025	1,34
19	3	0,2	0,291685	0,2883	0,01	0,0087	0,0012	12,83
20	6	0,77	0,524185	0,5242	0,01	0,0073	0,0026	26,61
21	9	0,66	0,440711	0,4408	0,01	0,0031	0,0068	68,47
22	12	0,9	0,612723	0,6127	0,01	0,013	0,003	25,47
23	15	1,795	1,269942	1,269937	0,01	0,012	0,0025	25,46
24	20	3,24	1,576158	1,576158	0,01	0,0014	0,0085	85,49

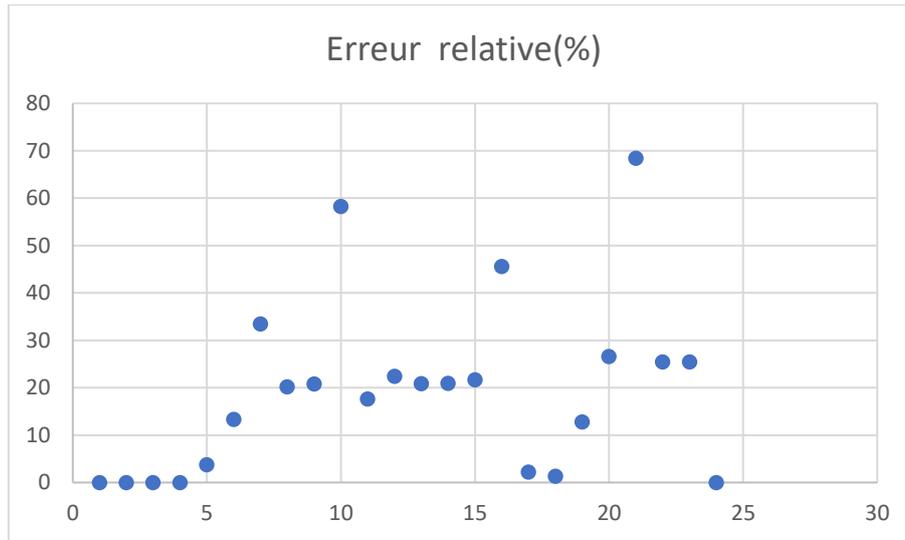


Figure IV.47 Représentation de l'erreur relative du modèle temporel.

Il est à noter qu'il est bien clair du tableau précédent que l'erreur relative des prédictions se diffère d'un échantillon à un autre.

A partir de la figure IV.47, pour la majorité des échantillons l'erreur relative ne dépasse pas 25%.

Le modèle de RNA choisi donne des résultats satisfaisants pour la plupart des échantillons réservés au test (24 échantillons), cependant, il existe des valeurs des conductivités prédits en utilisant notre modèle, sont loin des valeurs réelles (cibles) comme les échantillons 10, 21 et 24 (soulignée en gras) avec des erreurs relatives de 58%, 68% et 85% respectivement, et ça revient aux conditions des mesures.

IV.8 Comparaison entre les erreurs relatives de l'ensemble de test données par le vecteur fréquentiel et temporel

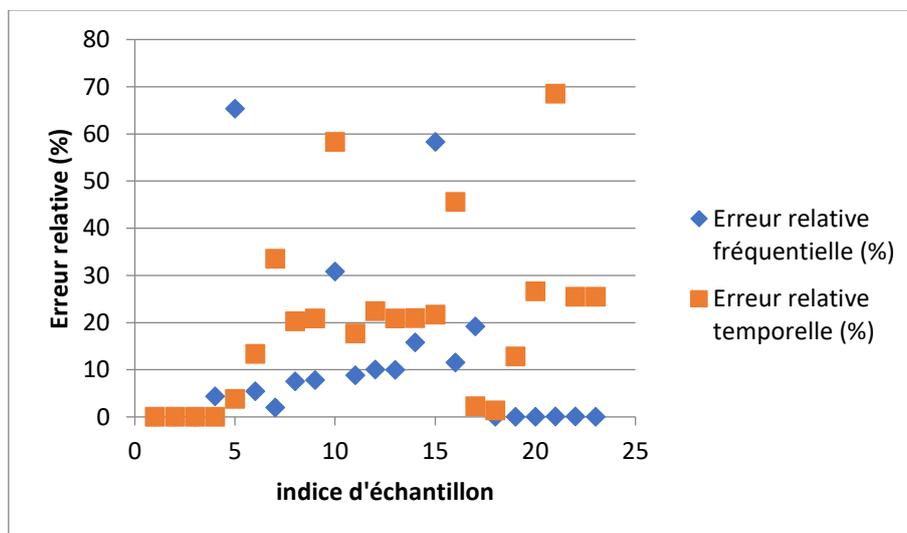


Figure IV.48 Erreurs relatives de modèle fréquentiel et temporel

On remarque que la plupart des points bleus sont situés au-dessous des point orange, ça montre que le modèle fréquentiel donne des bonnes approximations pour la majorité des indices d'échantillonnage et pour les défèrent calasse de conductivité.

IV.9 Conclusion

Dans ce chapitre, nous avons appliqué l'une des méthodes d'intelligence artificielle les plus utilisées, à savoir les RNA pour la prédiction de la conductivité de la pollution, parmi les avantages des réseaux de neurones mentionnés dans le Chapitre II , les réseaux de neurones ne nécessitent pas l'utilisation de modèles mathématiques très complexes pour leur fonctionnement, ainsi qu'un taux d'erreur très faible, nous avons atteint 11.5%.

Nous avons simulé plusieurs modèles de RNA utilisant un vecteur fréquentiel puis un vecteur temporel en changeant à chaque fois des paramètres qui influencent sur la précision du modèle a savoir les fonction d'activation, nombre de couches cachées, nombre de neurone par couche cachée et nous avons évalué les performances de tous les modèles puis nous avons tiré les deux modèles qui donnent les meilleurs performance.

A partir des résultats obtenus on en déduit que les caractéristiques fréquentielles donnent des précisions satisfaisants avec moins de couches cachées et de neurone par couche par rapport aux résultats obtenu par les caractéristiques temporelles, à savoir 11.5% pour le modèle fréquentiel et 23.67% pour le modèle temporel.

Il est noté que pour éviter toutes les contraintes de prédiction tel que le sur-apprentissage, sous-apprentissage ou divergence de la fonction de coût, il faut organiser la base de données en trois sous-ensembles pour que le processus d'apprentissage et de validation sera le mieux possible.

Conclusion Générale

Le travail de recherche présenté dans ce mémoire porte sur l'étude des réseaux de neurones artificiels pour la surveillance et le diagnostic des isolateurs de haute tension. Dans l'objectif de minimiser au maximum les interruptions et coupures de l'énergie transmises sur les lignes électriques due aux dégradations de la rigidité diélectrique des équipements de d'isolation.

Pour ce faire, nous avons pris des résultats des essais expérimentaux effectués au laboratoire de Haute Tension à l'ENP, effectués sur une chaîne d'isolateur réel simulant un modèle plan 1512L uniformément pollué.

Pour l'implémentation d'un réseau de neurones pour une application donnée, il faut tout d'abord étudier soigneusement l'application souhaitée, ainsi que les résultats désirés afin de valider son adaptabilité à une résolution efficace par des réseaux de neurones et précisant les objectifs à atteindre. Par la suite, il faut analyser les paramètres d'entrées des réseaux de neurones à savoir le choix du type de réseau, étude de la corrélation entre les entrées et les sorties et les différents paramètres internes (fonction d'apprentissage, etc...) en fonction des caractéristiques du problème étudié.

A cet effet, nous avons d'abord commencé par citer les travaux antérieurs qui ont été déjà traités à propos de problèmes similaires puis, dans un second lieu, nous avons présenté la théorie des réseaux de neurone ainsi que leurs domaines d'application et les différentes architectures et algorithmes d'apprentissages les plus utilisés. A la lumière des précédentes études qui ont été menés, il a été constaté que les perceptrons multicouches correspondent bien à notre cas d'application.

L'apprentissage du réseau dépend de plusieurs paramètres qui sont : l'erreur à atteindre par la fonction de coût, le nombre d'itérations, le taux d'apprentissage et la fonction d'activation. Un bon choix des paramètres mentionnés peut donner d'excellent résultats, ce qu'est montré dans la phase de test. Donc un bon apprentissage dépend aussi du bon choix des bases d'apprentissage pour obtenir bonne généralisation.

Après l'entraînement de chaque réseau de neurones nous avons évalué, leurs performances par plusieurs approches telles que : la fonction coût MSE, la méthode de la régression linéaire et l'erreur relative.

Pour tester la capacité de généralisation des réseaux, nous avons pris un ensemble de test constitués de 23 échantillons pour le premier modèle (fréquentiel) et 24 échantillons pour le deuxième modèle (temporel). Les résultats obtenus après la généralisation sont les valeurs de la conductivité avec des erreurs relatives qui ne dépassent pas 10% et 25% respectivement. Ceci montre la qualité d'apprentissage et la crédibilité des résultats des deux modèles du réseau de neurone que nous avons employés.

Aussi, comme stipulé par un certain nombre d'auteurs [38], nous confirmons que les données fréquentielles du courant de fuite donnent de meilleurs résultats de prédiction de la valeur de la conductivité, comparé aux données temporelles. D'ailleurs, la majeure partie des études menées ces dernières années se concentre exclusivement sur les méthodes de traitement fréquentiels, tels que les ondelettes continues ou discrètes.

Par ailleurs, étant donné que notre travail vise à mettre en place un RNA optimal, en minimisant l'erreur de prédiction de la valeur de la conductivité de la solution polluante. De ce fait, nous pouvons clairement affirmer que ce dernier est obtenu pour une fonction d'apprentissage de type traingdx, une seule couche cachée avec 6 neurones, ceci pour le vecteur fréquentiel.

Ce travail, non seulement, introduit les RNA pour la prédiction de la pollution des isolateurs, il met en avant l'importance de choisir des paramètres optimaux du RNA. Ces paramètres sont d'une importance capitale, car une infime variation concernant ce choix, aura des répercussions considérables sur la valeur prédite de la conductivité de pollution.

REFERENCES
BIBLIOGRAPHIQUES

- [1] M. Jabbari, Développement d'un modèle numérique générique de prédiction de la tension de contournement des isolateurs recouverts de glace, thèse de Doctorat en ingénierie, août 2018, Chicoutimi, Université du Québec. pp 35-90.
- [2] M. Fethi, Etude temporelle et fréquentielle du courant de fuite des isolateurs de poste recouverts de glace en vue du développement d'un système de surveillance et de prédiction en temps réel du contournement électrique, thèse de Doctorat en ingénierie, 2007, Chicoutimi, Université du Québec. pp 67-82.
- [3] M. Tegar, A. Mekhaldi et A. Boubakeur, Prediction of Polluted Insulators Characteristics using Artificial Neural Networks, IEEE Conference on Electrical Insulation and Dielectric Phenomena, 14 – 17 October, 2012, Delta Center – Ville, Montreal.
- [4] S.A Bessedik, H.Hadi, Modélisation statique des isolateurs pollués. ICEEE'08, 2nd International Conference on Electrical and Electronics Engineering, April. 2008, Algeria, Université de Laghouat.
- [5] S.A Bessedik, H.Hadi, Dynamic Arc Model of the Flashover of the Polluted Insulators. IEEE Conf. Electr. Insul. Dielectric Phenomena CEIDP, 2011, Cancun, Mexico. pp 550-554.
- [6] R. Boudissa, A. Haddad, Z. Sahli, A. Mekhaldi, Baersch, Performance of outdoor insulators under nonuniform pollution conditions., 14th Int. Symp. High Voltage Engineering, D-51. August 2005, China.
- [7] S.A.Bessedik, H.Hadi, Etude expérimentale du Modèle Ouvert., 4th International Conference on Electrotechnics. ICEL Novembre 2009, Université Oran, Algeria.
- [8] J.L. Rasolonjanahary, L. Krahenbuhl, A. Nicolas, Computation of electric fields and potential on polluted insulators using a boundary element method., IEEE Conference on Electrical Insulation and Dielectric Phenomena, 2015.
- [9] A. Bessedik, Contournement des isolateurs pollués, thèse de Doctorat en sciences, Université des sciences et de la technologie d'Oran Mohamed Boudiaf, 2015. pp 15-54.
- [10] P. Shaotong, L. Yunpeng, J. Xinxin, G. Jianghai, Z. Guangyang et W. Shenghui, UV-flashover evaluation of porcelain insulators based on deep learning, IET Science, 2018, Measurement & Technology, Vol. 12 Iss. 6, pp. 770-776.
- [11] M. H. Bui, M. Lech, E. Cheng, Object recognition using deep convolutional features transformed by a recursive network structure, IEEE Access, 2017. pp 10059–10066.
- [12] J. Patra, J.C. Juhola, M. Meher, Intelligent sensors using computationally efficient Chebyshev neural networks, IET Sci. Meas. Technol., 2008. pp 68–75.
- [13] Kadous Djamilia, Utilisation des réseaux de neurones comme outil du datamining d'un Génération de modèle comportemental d'un processus physique, thèse de Doctorat en ingénierie, 2015, Chicoutimi, Université du Québec. pp 17-25.
- [14] A. Czajka, K. Bowyer, M. Krumdick, Recognition of imageorientation-based iris spoofing', IEEE Trans. Inf. Forensics Sec., December 2017. pp 2184–2196.
- [15] B. Stamford, J. Peach, Scene detection using convolutional neural networks, IET Int. Conf. on Technologies for Active and Assisted Living. IET, 2016.
- [16] Tarek, M. Ouda, O. Hamza, Robust cancellable biometrics scheme based on neural networks', IET Biometrics, march 2016. pp 220–228.
- [17] O.E. Gouda et Dina M. Khalifa, Online monitoring of medium voltage overhead distribution lines polluted insulators severity, CIRED, Open Access Proc., Vol. 2017, Iss. 1. pp 372–375.
- [18] P. kamura, Monthly faults reports on Unified Egyptian Network, Egyptian Electricity Transmission Company, February and March 2014.
- [19] A. Nameer, E. Ayman, Bayesian Regularization of Neural Network to Predict Leakage CurrentinaSaltFogEnvironment, IEEE Transactions on Dielectrics and Electrical Insulation, Vol. 25, No. 2, April 2018.

- [20] M. Islam, T. Morimoto, Non-linear autoregressive neural network approach for inside air temperature prediction of a pillar cooler. *Int. J. Green Energy*, 2017. pp141–149.
- [21] Caswell, J.M. A Nonlinear Autoregressive Approach to Statistical Prediction of Disturbance Storm Time Geomagnetic Fluctuations Using Solar Data. *J. Signal Inf. Process.* 2014. pp 42–53.
- [22] F. Tsuruoka, Y. Tsujii, J. Ananiadou, Stochastic gradient descent training for l1-regularized log linear models with cumulative penalty, April 2009, Proceedings of the AFNLP/ACL.
- [23] V. Ferreira, D. André, F. Germano, K. Edson Guedes da Costa, Ultrasound and Artificial Intelligence Applied to the Pollution Estimation in Insulations, *IEEE Transactions on Power Delivery* Volume: 27 , Issue: 2 , April 2012. pp 583 - 589.
- [24] R. Menon, S. Kolambekar, N. J. Buch, and M. Ramamoorthy, Correlation of acoustic emission method and electrical method for detection of partial discharges in transformers, in *Proc. IEEE 7th Int. Conf. Solid Dielectric.*, Jun. 25–29, 2001, pp. 299–302.
- [25] B. Gajic, K. Paliwal, Robust parameters for speech recognition based on subband spectral centroid histograms, *IEEE Trans. Audio Speech, Language Process.*, vol. 14, no. 2, pp. 600–608, Mar. 20
- [26] B. Gajic, K. Paliwal, Robust parameters for speech recognition based on subband spectral centroid histograms, in *Proc. EUROSPEECH*, 2001, pp. 591–594.
- [27] M. Minsky, S. Papert, *Perceptrons And Introduction to Computational Geometry*. Cambridge, MA: MIT Press, 1969.
- [28] M. Riedmiller, H. Braun, A direct adaptive method for faster backpropagation learning: The RPROP algorithm, in *Proc. IEEE Int. Conf. Neural Netw.*, 1993, pp. 586–591.
- [29] L. Fausett, ed. *Fundamentals of Neural Networks Architectures, Algorithms, and Applications*, Prentice-Hal, 1994
- [30] M. Alp. *Introduction aux Réseaux de Neurones Artificiels Feed Forward*. Paris : Dunod, 2014. 926 p. Technique & ingénierie. ISBN 978-2-10-070545-0, 2008.
- [31] M. H. Hassoun, *Fundamentals of Artificial Neural Networks*, 1995, Cambridge, MA, USA: MIT Press.
- [32] H. Dinh, A study of cell-based genetic algorithms with applications to neural networks, *Mémoire de fin d'étude en informatique*, 2013, Long Beach, Université du Californie. pp 9-12.
- [33] D.S. Broomhead and D. Lowe. Multivariate functional interpolation and adaptive networks. *Complex Systems*, 2:321 355, 1988 pp. 1473–1476. 1992.
- [34] Priyanka M. et Jayant S. (2016), A comparative study of Elman backpropagation neural network and radial basis function neural network models for speech recognition, *International Journal of Engineering Applied Sciences and Technology*, Vol. 1, Issue 9, ISSN No. 2455-2143.
- [35] A. Amrani, surveillance et diagnostic d'une ligne de production par les réseaux de neurones, *Mémoire de fin d'étude en Electrotechnique*, 2010, Université M'Hamed Bougara Boumerdes. pp 56-83.
- [36] M.Basseville, A.Beneviste, Q.Zhangh, Surveillance d'installation industrielle démarche générale et conception de T'algorithmique », *INRIA*, Vol.2889, pp. 26-35, 1987.I
- [37] Y. Mi, S. Rui, Shoulong, C. Yao, Z. Zhao, Optical Fiber Sensor System for Monitoring Leakage Current of Post Insulators Based on RBF Neural Network, *IEEE Transactions on Dielectrics and Electrical Insulation*, 2010. Pp 11-18.
- [38] C. Volat, F. Meghnefi, M. Farzaneh et H. Ezzaidi, Monitoring Leakage Current of Ice-covered Station Post Insulators using Artificial Neural Networks, *IEEE Transactions on Dielectrics and Electrical Insulation* Vol. 17, No. 2, 2010. pp 17-26.

- [39] M. Douar, M. Bouzidi, Caractérisation de l'état de surface d'un isolateur pollué à l'aide des signaux du courant de fuite et de la tension appliquée, Mémoire du projet de Fin d'Etudes, Ingénieur d'Etat en Electrotechnique, 2009, Ecole National Polytechnique d'Alger. Pp 56-104.
- [40] A. K. Chaou, A. Mekhaldi et M. Tegua , Wavelet Packet Transform based Multi Resolution Analysis Technique for Classification of LC Waveforms on Polluted Insulating Surfaces, 2014 Annual Report Conference on Electrical Insulation and Dielectric Phenomena.
- [41] J. Zhang et M. Farzaneh, Propagation of AC and DC Arcs on Ice Surface, IEEE Trans. On Dielectrics and electrical Insulation, Vol.7, no.2, pp. 269276, avril 2000.
- [42] M. Farzaneh, I. Fofana, C. Tavakoli et X. Chen, Dynamic Modelling of DC Arc Discharge on Ice Surfaces, IEEE Transactions on Dielectrics and Electrical Insulation, 2016.
- [43] S. Saha, A Comprehensive Guide to Convolutional Neural Networks, Synthesis Lectures on Computer Vision, Morgan & Claypool Publishers, February 13, 2018, 389 p.
- [44] ujjwalkarn. An Intuitive Explanation of Convolutional Neural Networks [en ligne]. Posté le 11 aout 2016. [Consulté le 15 mai 2019]. Disponible sur: <<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>>
- [45] S. Albawi, Understanding of Convolutional Neural Network (CNN)—Deep Learning, The International Conference on Engineering and Technology, 26-28 January 2017, Antalya, Turkey.
- [46] Cong shuang, Neural Network Theory and Application based on MATLAB Toolbox, Hefei, 1998, University of Science and Technology of China Press.
- [47] Cong shuang, Facing the MATLAB Toolbox of Neural Network Theory and Application, Hefei,1998, University of Science and Technology of China Press.