

# **Relais de perte de synchronisme par apprentissage profond**

par

**Philippe LEFEBVRE**

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE  
AVEC MÉMOIRE EN GÉNIE ÉLECTRIQUE  
M.Sc.A

MONTRÉAL, LE 7 AOÛT 2020

**ÉCOLE DE TECHNOLOGIE SUPÉRIEURE  
UNIVERSITÉ DU QUÉBEC**

©Tous droits réservés, Philippe Lefebvre, 2020

©Tous droits réservés

Cette licence signifie qu'il est interdit de reproduire, d'enregistrer ou de diffuser en tout ou en partie, le présent document. Le lecteur qui désire imprimer ou conserver sur un autre media une partie importante de ce document, doit obligatoirement en demander l'autorisation à l'auteur.

**PRÉSENTATION DU JURY**

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Ambrish Chandra, directeur de mémoire  
Département de génie électrique à l'École de technologie supérieure

M. Innocent Kamwa, codirecteur de mémoire  
Institut de recherche d'Hydro-Québec

M. Nicolas Constantin, président du jury  
Département de génie à l'École de technologie supérieure

M. Esmaeil Ghahremani, membre du jury  
Institut de recherche d'Hydro-Québec

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 10 JUILLET 2020

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE



## **REMERCIEMENTS**

Cette œuvre est l'achèvement d'une aventure pour moi qui a commencé grâce au soutien et à l'insistance de plusieurs personnes chères pour moi le long de ma carrière.

Tout d'abord cette aventure a commencé avec l'aide de mon ami Subhadeep Bhattacharya, qui m'a poussé à entamer des études supérieures et m'a offert un soutien à la fois moral et technique. Sans lui cette aventure n'aurait jamais commencé.

Ensuite je tiens à remercier le professeur Ambrish Chandra pour son ouverture et grande générosité de temps pour ses conseils. Durant cette aventure le sujet de recherche a changé énormément à travers le temps et il a su m'aider à m'orienter vers les bons choix à prendre tout au long.

Je tiens également à remercier monsieur Innocent Kamwa qui a su m'ouvrir sa porte et m'offrir une opportunité de travailler sur un magnifique projet dans ce passionnant domaine de la stabilité de réseau électrique et à m'offrir l'aide et le support afin de bien mener ce projet.

Finalement, je voudrais remercier profondément les membres de ma famille, mes amis et tous mes collègues qui m'ont apporté leurs support, la motivation et malgré les difficultés, la volonté de persévérer à l'accomplissement de ce tout premier projet de recherche pour moi.



# **Relais de perte de synchronisme par apprentissage profond**

Philippe LEFEBVRE

## **RÉSUMÉ**

L'objectif de cet exercice est de pouvoir explorer la possibilité d'utiliser l'apprentissage profond pour concevoir un relais de perte de synchronisme. Grâce au simulateur de transitoire électromagnétique EMTP, une série de simulations sont produites et ensuite classées grâce à un algorithme d'apprentissage supervisé de multiples séries temporelles de type « Multivariate Attentional Long Short Term Memory Fully Convolutional Network » (MALSTM-FCNs). Un réseau d'étude est modélisé avec le simulateur EMTP pour représenter une région sous étude.

Dans ce réseau, quatre emplacements seront choisis pour prendre des mesures pour le relais de perte de synchronisme. Les mesures comprennent les tensions triphasées temporelles, les courants triphasés temporels ainsi que les puissances actives et réactives triphasées temporelles. Ces mesures sont ensuite importées dans un outil de traitement de grosse matrice nommée MATLAB. Une fois importées dans MATLAB, les données sont séparées par générateur et sont manipulées pour cataloguer les simulations en deux classes selon leur stabilité.

Par la suite, pour chaque générateur, une base d'entraînement et de test est construite à partir de ces simulations classées et les cas stables et instables sont distribués de façon aléatoire dans ces deux ensembles. Par la suite l'algorithme d'apprentissage profond MALSTM-FCN est employé sur l'ensemble d'entraînement pour permettre à l'algorithme d'apprendre à reconnaître les situations réseaux stables et instables. Enfin la performance de l'algorithme est évaluée sur un ensemble de test pour déterminer sa capacité à généraliser son apprentissage à un ensemble de cas non connus.

La nouveauté de l'approche réside dans l'emploi de données brutes non filtrées et d'un petit nombre de variables d'entrée, soit les tensions et courants triphasés temporels ainsi que les puissances actives et réactives temporelles. L'approche est aussi différente des autres méthodes d'apprentissage machine du fait qu'il n'y a pas de « feature Extraction » (extraction d'attributs) qui consiste à faire un traitement sur les données pour extraire de l'information supplémentaire, comme estimer l'état des génératrices.

**Mots-clés :** Apprentissage machine, intelligence artificielle, stabilité transitoire, stabilité de réseau, perte de synchronisme



# **Out-of-step relay by deep learning**

Philippe LEFEBVRE

## **ABSTRACT**

The objective of this exercise is to explore the possibility of using Deep Learning to develop an out of step relay. With the transient electromagnetic simulator EMTP, a series of simulations are produced and then classified with a machine learning algorithm for multiple time series called MALSTM-FCN (Multivariate Attentional Long Short Term Memory Fully Convolutional Network). Using EMTP, a test network is modelled to represent a region to study.

In this network, 4 places will be chosen to take measures for the loss of step relay. The measures include three-phase voltages, three-phase currents and active and reactive three-phase powers. These measures are than imported in a matrix treatment tool called MATLAB well suited to handle big matrices. Once imported in MATLAB, the data are separated per generator and catalogued in two classes of stable or unstable simulation.

Following this, for each generator, a training database and a test database are constructed from the set of stable and unstable simulations randomly so that they are evenly represented in those two databases. Then the Deep Learning algorithm MALSTM-FCN is used on the training database to allow the algorithm to learn to recognize stable and unstable network situations. Finally, the performance of the algorithm is evaluated in the training database to determine its generalization capability on an unlearned set of simulations.

The novelty of this approach resides in the use of unfiltered raw data from a small number of inputs such as three phase voltage and currents and tree-phase active and reactive power. The approach is always markedly different from other machine learning methods in that there is no feature extraction phase used to extract additional information from the inputs such as additional information concerning the state of the generators.

**Keywords:** Machine learning, artificial intelligence, transient stability, power system stability and control, loss of synchronism



## TABLE DES MATIÈRES

	Page
INTRODUCTION .....	1
<b>CHAPITRE 1 INTRODUCTION THÉORIQUE .....</b>	<b>5</b>
1.1 Notions et concepts théoriques .....	5
1.2 Principe de fonctionnement du relais.....	7
1.3 Réseau de neurones artificiels classique.....	8
1.3.1 Principe général .....	8
1.3.2 Composantes .....	9
1.3.3 Modèle de neurones .....	10
1.3.4 Connexions et poids.....	12
1.3.5 Fonction de propagation .....	13
1.3.6 Apprentissage.....	13
1.3.7 Modèle de réseau en Deep Learning.....	13
1.4 Réseau de neurones récurrent .....	15
1.4.1 Réseau de neurones “Fully Recurrent” .....	15
1.4.2 Long Short-Term Memory RNN .....	19
1.5 Le réseau multi variable LSTM pleinement convolutionel .....	20
1.5.1 Architecture réseau .....	20
1.6 Évaluation du modèle .....	22
1.6.1 Matrice de confusion.....	22
<b>CHAPITRE 2 RÉSEAU D’ÉTUDE.....</b>	<b>25</b>
2.1 Logiciel utilisé EMTP-RV .....	25
2.2 Réseau identifié.....	25
2.3 Représentation du réseau d’étude sur EMTP.....	27
2.3.1 Description détaillée du réseau d’étude .....	27
2.3.2 Générateurs .....	27
2.3.3 Transformateurs .....	28
2.3.4 Lignes.....	29
2.3.5 La compensation statique.....	31
2.3.6 Les charges.....	31
2.3.7 Sous-circuit de mesures .....	32
2.4 Simulations .....	34
2.4.1 Load-Flow.....	34
2.4.2 Zone 1 .....	35
2.4.3 Zone 2 .....	36
2.4.4 Zone 3 .....	37
2.4.5 Les types de défauts .....	38
2.4.6 Script des simulations .....	38

CHAPITRE 3 TRAITEMENT DES DONNÉES.....	39
3.1 Importations des données sous MATLAB.....	39
3.2 Classement des simulations en cas stables et instables.....	41
3.2.1 Introduction théorique au centre d'inertie.....	41
3.2.2 Critères d'instabilité.....	43
3.2.3 Application sur l'ensemble de données .....	45
3.2.4 Vecteur d'état.....	45
3.3 Création des différents ensembles .....	46
3.4 Exportation des données .....	47
CHAPITRE 4 APPRENTISSAGE DES DONNÉES.....	49
4.1 TensorFlow .....	49
4.2 Détermination du nombre d'époques optimal.....	49
4.2.1 100 époques .....	51
4.2.2 200 époques .....	52
4.2.3 300 époques .....	53
4.2.4 400 époques .....	54
4.2.5 500 époques .....	55
4.2.6 600 époques .....	56
4.2.7 700 époques .....	57
4.2.8 800 époques .....	58
4.2.9 900 époques .....	59
4.2.10 1000 époques .....	60
4.2.11 2000 époques .....	61
4.2.12 Résumé et analyse.....	62
4.3 Phase de test du réseau de neurones .....	63
4.3.1 Générateur 1.....	64
4.3.2 Générateur 2.....	71
4.3.3 Générateur 3.....	76
4.3.4 Générateur 4.....	79
CONCLUSION.....	85
RECOMMANDATIONS .....	91
ANNEXE I LISTE DES SIMULATIONS EN FONCTION DES FICHIERS DES GÉNÉRATEURS.....	93
ANNEXE II VALEUR DE PRÉDICTION DE L'ENSEMBLE DE TEST .....	101
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES.....	109

## LISTE DES TABLEAUX

	Page	
Tableau 1.1	Matrice de confusion.....	23
Tableau 1.2	Terminologie et dérivation de la matrice de confusion .....	23
Tableau 2.1	Détail des générateurs .....	28
Tableau 2.2	Détail des transformateurs .....	29
Tableau 2.3	Détail des lignes .....	29
Tableau 2.4	Détail des charges .....	31
Tableau 2.5	Générateur pour le Load-Flow 1 .....	35
Tableau 2.6	Résultat du Load-Flow 1.....	35
Tableau 2.7	Générateur pour Load-Flow 2.....	36
Tableau 2.8	Résultat pour Load-Flow 2 .....	36
Tableau 2.9	Générateur pour Load-Flow 3.....	37
Tableau 2.10	Résultat pour Load-Flow 3 .....	37
Tableau 4.1	Matrice de confusion du générateur 3 avec 100 époques .....	51
Tableau 4.2	Taux mesurés pour la matrice de confusion du générateur 3 avec 100 époques .....	51
Tableau 4.3	Matrice de confusion du générateur 3 avec 200 époques .....	52
Tableau 4.4	Taux mesurés pour la matrice de confusion du générateur 3 avec 200 époques .....	52
Tableau 4.5	Matrice de confusion du générateur 3 avec 300 époques .....	53
Tableau 4.6	Taux mesurés pour la matrice de confusion du générateur 3 avec 300 époques .....	53
Tableau 4.7	Matrice de confusion du générateur 3 avec 400 époques .....	54
Tableau 4.8	Taux mesurés pour la matrice de confusion du générateur 3 avec 400 époques .....	54
Tableau 4.9	Matrice de confusion du générateur 3 avec 500 époques .....	55

Tableau 4.10	Taux mesurés pour la matrice de confusion du générateur 3 avec 500 époques .....	55
Tableau 4.11	Matrice de confusion du générateur 3 avec 600 époques .....	56
Tableau 4.12	Taux mesurés pour la matrice de confusion du générateur 3 avec 600 époques .....	56
Tableau 4.13	Matrice de confusion du générateur 3 avec 700 époques .....	57
Tableau 4.14	Taux mesurés pour la matrice de confusion du générateur 3 avec 700 époques .....	57
Tableau 4.15	Matrice de confusion du générateur 3 avec 800 époques .....	58
Tableau 4.16	Taux mesurés pour la matrice de confusion du générateur 3 avec 800 époques .....	58
Tableau 4.17	Matrice de confusion du générateur 3 avec 900 époques .....	59
Tableau 4.18	Taux mesurés pour la matrice de confusion du générateur 3 avec 900 époques .....	59
Tableau 4.19	Matrice de confusion du générateur 3 avec 1000 époques .....	60
Tableau 4.20	Taux mesurés pour la matrice de confusion du générateur 3 avec 1000 époques .....	60
Tableau 4.21	Matrice de confusion du générateur 3 avec 2000 époque.....	61
Tableau 4.22	Taux mesurés pour la matrice de confusion du générateur 3 avec 2000 époques .....	61
Tableau 4.23	Ensemble des simulations stables et instables .....	63
Tableau 4.24	Matrice de confusion du générateur 1.....	64
Tableau 4.25	Taux mesurés pour la matrice de confusion du générateur 1.....	65
Tableau 4.26	Matrice de confusion du générateur 2.....	71
Tableau 4.27	Taux mesurés pour la matrice de confusion du générateur 2.....	72
Tableau 4.28	Matrice de confusion du générateur 3.....	76
Tableau 4.29	Taux mesurés pour la matrice de confusion du générateur 3.....	76
Tableau 4.30	Matrice de confusion du générateur 4.....	79

Tableau 4.31	Taux mesurés pour la matrice de confusion du générateur 4.....	80
Tableau 4.32	Ensemble des simulations stables et instables .....	86
Tableau-A I-1	L'ensemble de test pour le générateur 1 .....	93
Tableau-A I-2	L'ensemble de test pour le générateur 2 .....	95
Tableau-A I-3	L'ensemble de test pour le générateur 3 .....	97
Tableau-A I-4	L'ensemble de test pour le générateur 4 .....	99
Tableau-A II-1	Valeur de prédiction pour le générateur 1.....	101
Tableau-A II-2	Valeur de prédiction pour le générateur 2.....	103
Tableau-A II-3	Valeur de prédiction pour le générateur 3.....	104
Tableau-A II-4	Valeur de prédiction pour le générateur 4.....	106



## LISTE DES FIGURES

	Page
Figure 1.1	N Série temporelle associée à M série temporelle .....6
Figure 1.2	N Série temporelle associée à M sorties .....7
Figure 1.3	Colored neural network.....9
Figure 1.4	Perceptron .....10
Figure 1.5	The Mark I perceptron .....12
Figure 1.6	Deep neural network .....14
Figure 1.7	Recurrent neural network unfold .....15
Figure 1.8	Architecture of a deep recurrent neural network .....17
Figure 1.9	The logistic sigmoid function .....18
Figure 1.10	A diagram for a one-unit long short-term memory.....19
Figure 1.11	Multivariate attentional Long Short Term Memory Fully Convolutionel Network.....21
Figure 1.12	Block Squeeze-and-Excite .....22
Figure 2.1	Réseau Anderson .....26
Figure 2.2	Réseau Anderson dans EMTP-RV.....27
Figure 2.3	Détail d'une ligne sur EMTP-RV .....30
Figure 2.4	Détail du sous-circuit de mesures .....33
Figure 2.5	Détail du sous-circuit de mesures pour une phase .....34
Figure 3.1	Workspace de MATLAB .....40
Figure 3.2	Centre d'inertie d'angle des quatre générateurs.....44
Figure 3.3	Centre d'inertie de vitesse des quatre générateurs.....45

Figure 3.4	Précision de la prédiction en fonction de la taille du groupe d'entraînement .....	47
Figure 4.1	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 100 époques.....	51
Figure 4.2	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 200 époques.....	52
Figure 4.3	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 300 époques.....	53
Figure 4.4	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 400 époques.....	54
Figure 4.5	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 500 époques.....	55
Figure 4.6	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 600 époques.....	56
Figure 4.7	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 700 époques.....	57
Figure 4.8	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 800 époques.....	58
Figure 4.9	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 900 époques.....	59
Figure 4.10	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 1000 époques.....	60
Figure 4.11	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 2000 époques.....	61
Figure 4.12	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 1 .....	64
Figure 4.13	Vitesse Angulaire (pu) de la simulation faux positif no 2 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	66
Figure 4.14	Phase (radian) de la simulation faux positif no 2 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	67

Figure 4.15	Vitesse Angulaire (pu) de la simulation instable no 9 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	68
Figure 4.16	Phase (radian) de la simulation instable no 9 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	68
Figure 4.17	Vitesse Angulaire (pu) de la simulation faux négatif no 30 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	69
Figure 4.18	Phase (radian) de la simulation faux négatif no 30 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	70
Figure 4.19	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 2 .....	71
Figure 4.20	Vitesse Angulaire (pu) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	73
Figure 4.21	Phase (radian) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	74
Figure 4.22	Tension (volt) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu.....	75
Figure 4.23	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 3 .....	76
Figure 4.24	Vitesse Angulaire (pu) de la simulation prédictée instable no 7 en rouge et la simulation prédictée stable no 40 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	78
Figure 4.25	Vitesse Angulaire (pu) de la simulation prédictée instable no 7 en rouge et la simulation prédictée stable no 40 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	79
Figure 4.26	Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 4 .....	79
Figure 4.27	Vitesse Angulaire (pu) de la simulation prédictée instable no 4 en rouge et la simulation prédictée stable no 20 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert. ....	80

Figure 4.28	Phase (radian) de la simulation prédictive instable no 4 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert.....	81
Figure 4.29	Vitesse Angulaire (pu) de la simulation prédictive instable no 18 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert. ....	82
Figure 4.30	Phase (radian) de la simulation prédictive instable no 18 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert.....	83

## **LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES**

ALSTM	Attentional Long Short Term Memory
ALSTM-FCN	Fully Convolutional Attentional Long Short Term Memory
ANN	Artificial Neural Network
C	Capacité
COI	Center of Inertia
COIA	Center of Inertia of Angle
COIΩ	Center of Inertia of Speed
cy	Cycles
DCG	Development Coordination Group
EMTP	Logiciel pour des études transitoires électromagnétiques
F	Farad
FCN	Fully Convolutional Network (réseau pleinement convolutionnel)
FN	Faux négatif
FP	Faux positif
Gb	Gigabyte
Gen	Générateur
H	Henry
Hz	Hertz
I	Courant
kVar	kiloVar
L	Inductance
LSTM	Long Short Term Memory (Réseau récurrent à longue mémoire à court terme)

n	Nombre total de simulations
Mb	Megabyte
MALSTM	Multivariate Attentional Long Short Term Memory
MATLAB	Logiciel de calcul Matriciel
$M_t$	Somme de tous les moments d'inertie des génératrices
MVA	Méga Voltampère
MW	Mégawatt
$\Omega$	Ohm
P	Puissance active
PQ	Barre de type charge de puissance active et réactive
p.u	Per Unit (par unité)
PV	Barre de type Puissance Tension
Q	Puissance réactive
R	Résistance
RLC	Charge résistive, inductive et capacitive
RMS	Root Men Square
RNN	Recurrent Neural Network (Réseau de neurones récurrent)
SCV	Swing Center Voltage
T	Temporel
us	microseconde
V	Tension
VN	Vrai négatif

VP	Vrai positif
$V\delta$	Barre de type Tension Angle
$\delta_{COI}$	Angle moyen de toutes les machines
$\omega_{COI}$	Vitesse angulaire moyenne de toutes les machines



## INTRODUCTION

Malgré la très grande fiabilité des grands réseaux de transport électrique, la principale difficulté à leur opération est la gestion de la stabilité. Cette stabilité des réseaux de transport a toujours été un enjeu pour l'exploitation de grands réseaux électriques et des réseaux voisins. Les problèmes de stabilité sont au cœur de plusieurs incidents au fil du temps sur de nombreux réseaux électriques à travers le monde.

Pour relever le défi des problèmes de stabilités, de nombreuses méthodes ont été étudiées dans le passé pour permettre de stabiliser les réseaux à l'aide de diverses technologies. Plusieurs relais de protection de synchronisme ont déjà été développés. Ceux-ci mesurent les caractéristiques concentriques et le taux de changement de l'impédance, pour ne citer qu'elles (Kundur, 1994).

Avec l'ère numérique, de nouvelles approches sont maintenant envisageables pour résoudre des problèmes de stabilité (Kirby et al., 2011) ; (Perron et al., 2018). Récemment, l'apparition de nouvelles méthodes permettent d'entraîner les ordinateurs à reconnaître certains phénomènes. Cette approche prend un nouvel envol grâce aux puissances de calcul des ordinateurs actuels (Zhang, Wang, Liu & Bao, 2018). L'apprentissage machine est une discipline de l'intelligence artificielle qui vise à faire apprendre les ordinateurs. Cet ouvrage vise à explorer la possibilité d'utiliser l'apprentissage machine pour des applications en problèmes de stabilité de réseaux électriques.

Les méthodes d'apprentissage machine existaient depuis longtemps. Leur mise en application n'a toutefois commencé que depuis quelques années grâce à la plus grande capacité des ordinateurs (Chothani, Bhalja & Parikh, 2014).

Parmi de nombreuses recherches dans le domaine de l'intelligence artificielle, on peut citer que dans les années 1940, McCulloch et Pins furent les premiers à publier des travaux sur leur modélisation de réseaux de neurones, puis Farley et Clark, en 1954, furent les premiers à

utiliser des ordinateurs pour simuler un réseau de neurones. En 1958, Rosenblatt créa le modèle du perceptron. Puis en 1965, le premier réseau de neurones artificiel fonctionnel avec plusieurs couches de neurones fut publié par Ivakhnenko et Lapa. Toutefois, les technologies ont changé moins rapidement dans le domaine électrique. Le réseau électrique possède les mêmes composantes qu'avant. La seule différence est qu'il y a de plus en plus de consommateurs d'énergie pour les mêmes infrastructures. De nouvelles infrastructures électriques sont difficiles à développer pour des raisons d'acceptabilité sociale.

Cette situation amène l'exposition des réseaux contemporains à plus de problèmes d'instabilité. Lorsqu'il y a une perturbation du réseau lors d'un défaut de ligne ou lors d'une diminution de la charge ou encore l'ouverture de plusieurs lignes, il y a une accélération ou un ralentissement des machines électriques. Si ces perturbations sont trop brusques, il y a un risque que les machines décrochent et entraînent des pertes de service sur le réseau (Kundur, 1994). Ce fut le cas de la ville de New York en 2003 qui a perdu le service électrique pour des millions de consommateurs d'énergie.

L'idée de ce rapport est de développer une façon de prévoir les instabilités pour des applications sur un relais de perte de synchronisme qui a pour rôle d'isoler certaines sections d'un réseau lorsque la fréquence accélère ou décèle de façon importante, ce qui peut entraîner une perte de la synchronisation de la fréquence d'un réseau. Ce relais peut prendre des décisions comme déconnecter des machines ou des charges pour éviter ces écarts de fréquences trop élevés.

Nous pouvons créer une base de données de cas stables et instables pouvant servir d'ensemble d'entraînement à un algorithme d'apprentissage machine. L'idée est d'entraîner l'algorithme sur des cas où la stabilité est prédéfinie pour ensuite lui permettre de repérer ces cas lui-même avec un minimum de temps de détection. Dans le livre de Kundur (1994), l'auteur crée une base de données de Load-Flow et, en fonction de la durée et de l'état de charge du réseau, il est possible de prédire les courts-circuits entraînant une instabilité.

Le but de ce document est de déterminer si c'est envisageable de faire un relais de protection de synchronisme en temps réel qui apprend à partir des données de cas instables d'un réseau. Ceci permettrait d'avoir des relais versatiles qui peuvent prendre des décisions en fonction de l'unicité des réseaux à protéger.

Ce rapport est présenté comme suit : le premier chapitre décrit les principes et notions qui sont utilisés tout au long du rapport. Il décrit ensuite le principe de fonctionnement du relais et pour terminer il présente au lecteur un aperçu théorique de l'apprentissage machine et des réseaux de neurones artificiels.

Le deuxième chapitre expliquera le réseau électrique qui sera utilisé et les divers éléments qui le composent (les lignes, les transformateurs, les générateurs, etc. seront détaillés). De plus, ce chapitre présentera les diverses simulations temporelles et de Load-Flow qui ont été effectuées.

Dans le troisième chapitre, le lecteur trouvera les informations décrivant le processus de création, de manipulation et de transformation de la base de données. Ce chapitre présentera en plus la méthode de tri des simulations stables et instables pour former ces deux catégories de simulations pour l'ensemble d'apprentissage supervisé.

Le chapitre quatre présentera la phase d'entraînement des algorithmes d'apprentissage machine en utilisant la base de données de simulations créée, ainsi que la validation des performances sur le système.

Finalement une conclusion résume les travaux et les résultats présentés dans ce document. D'autres pistes de recherche ou de continuité dans cette avenue y seront exposées.



# CHAPITRE 1

## INTRODUCTION THÉORIQUE

Cette section a pour but d'introduire le lecteur aux notions théoriques nécessaires. La première section présente les différentes notations ainsi que les termes qui seront utilisés tout au long de ce rapport.

La section suivante exposera ainsi les idées et le principe derrière le relais de protection de synchronisme par apprentissage profond.

Finalement, les dernières sections présentent sommairement les techniques d'apprentissage profond employées pour réaliser le relais.

### 1.1 Notions et concepts théoriques

L'algorithme d'apprentissage machine requiert un ensemble d'entraînement pour lui permettre d'apprendre le comportement souhaité. Un ensemble d'entraînement est un ensemble de cas et ses réponses attendues. Un autre ensemble appelé ensemble de test est composé de cas différents de celui de l'ensemble d'apprentissage et sert à valider l'apprentissage d'algorithme. Ces ensembles sont différents puisqu'on souhaite généraliser l'apprentissage de la machine et non simplement les cas appris dans l'ensemble d'apprentissage.

On peut visualiser l'ensemble d'apprentissage et de test par une matrice. À chaque temps, n variables,  $x_1$  à  $x_n$ , sont associées à m états de sorties  $o_1$  à  $o_m$  ce qui correspond à une fonction.

$$X \rightarrow O : \mathbf{R} \times \mathbf{R}^n \rightarrow \mathbf{R} \times \mathbf{R}^m \quad (1.1)$$

t	$x_1(t)$	$x_2(t)$	...	$x_n(t)$
1	$x_1(1)$	$x_2(1)$	...	$x_n(1)$
2	$x_1(2)$	$x_2(2)$	...	$x_n(2)$
3	$x_1(3)$	$x_2(3)$	...	$x_n(3)$
⋮	⋮	⋮	⋮	⋮
T	$x_1(T)$	$x_2(T)$		$x_n(T)$

↓

t	$o_1(t)$	$o_2(t)$	...	$o_m(t)$
1	$o_1(1)$	$o_2(1)$	...	$o_m(1)$
2	$o_1(2)$	$o_2(2)$	...	$o_m(2)$
3	$o_1(3)$	$o_2(3)$	...	$o_m(3)$
⋮	⋮	⋮	⋮	⋮
T	$o_1(T)$	$o_2(T)$		$o_n(T)$

Figure 1.1 N Série temporelle associée à M série temporelle

Lorsque les états de sortie ont tous la valeur 0 à l'exception d'un seul qui prend la valeur 1, on dit que c'est un ensemble d'apprentissage de classification. Dans un problème de classification, l'idée est de classer les valeurs des n différentes variables en m classes.

Si les états de sortie sont des classificateurs pour une série de valeurs d'entrée, l'ensemble d'apprentissage est alors un ensemble de séries de valeurs d'entrée pour un seul état de sortie.

$$X \rightarrow O : \mathbf{R} \times \mathbf{R}^n \rightarrow \mathbf{R}^m \quad (1.2)$$

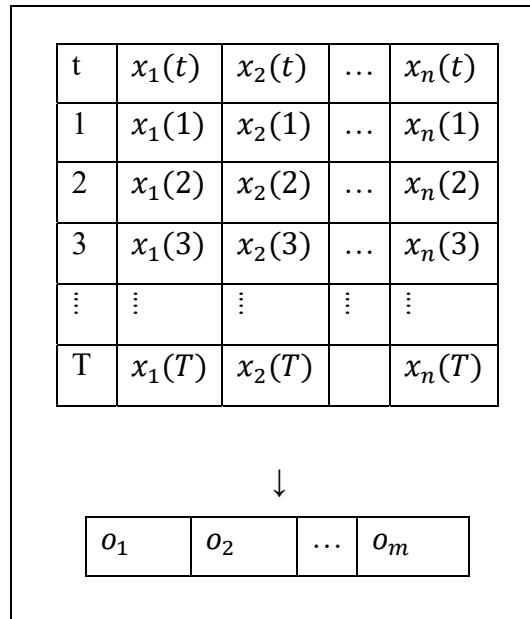


Figure 1.2 N Série temporelle associée à M sorties

Il s'agit donc d'une classification de série de données dans le temps, c'est-à-dire qu'à la figure ci-haut, chaque colonne correspond à une seule valeur de sortie. Dans notre cas particulier, notre classification se fait sur deux états de sortie, l'état stable et l'état instable.

## 1.2 Principe de fonctionnement du relais

Le principe de classement qui sera employé utilisera la méthode de centre d'inertie (Burhan, 2012) ; (Sauer & Pai, 1998) ; (Hashim, Abidin, Yap, Musirin & Zulkepali, 2010). Les simulations effectuées sont au préalable manuellement classées stables et instables. Pour les simulations stables, la variable à prédire de chaque série et pour chaque générateur est définie à 0 alors que pour les cas instables, celle-ci est définie à 1. Le modèle de prédiction est ensuite entraîné par apprentissage supervisé à classifier les cas série après série. Appliquée en temps réel, après un court-circuit, chaque nouvelle donnée échantillonnée est ajoutée à la série et classée par le modèle de prédiction dans un état stable ou instable jusqu'à ce qu'une décision d'ouverture des disjoncteurs soit prise.

### 1.3 Réseau de neurones artificiels classique

Les réseaux de neurones artificiels classique (Artificial Neural Network ANN) sont des systèmes vaguement inspirés des réseaux de neurones biologiques qui constituent le cerveau animal. Ces systèmes « apprenant » exécutent des tâches en considérant des exemples sans être programmés par des règles spécifiques. Par exemple, dans la reconnaissance d’images, ils peuvent apprendre à reconnaître un chien en analysant plusieurs exemples d’images contenant un chien qui ont été préalablement identifiées comme des images contenant un chien ou non. Ils peuvent faire ce genre de tâche sans connaissance à priori de ce qu’est un chien.

#### 1.3.1 Principe général

Un réseau de neurones artificiels est composé de 3 éléments :

- le premier élément du réseau contient des neurones d’entrée du réseau de neurones en rouge dans l’image suivante. Ceux-ci sont joints aux neurones de la couche du milieu en bleu nommée couche cachée. Ces neurones servent d’entrée aux informations de l’extérieur vers le réseau. Ces neurones ne font aucun traitement, ils ne font que diffuser l’information brute de l’extérieur au réseau;
- le deuxième élément du réseau contient les neurones de la couche cachée en bleu dans l’image suivante. Ces neurones font le travail de traiter l’information des différents neurones d’entrée. Les neurones de la couche cachée sont reliés aux neurones de la couche de sortie;
- le troisième élément du réseau contient les neurones de la couche de sortie en vert dans l’image suivante. Ces neurones sont rattachés aux différents neurones de la couche cachée. Ces neurones font un traitement de l’information des neurones de la couche précédente pour produire l’information à la sortie du réseau.

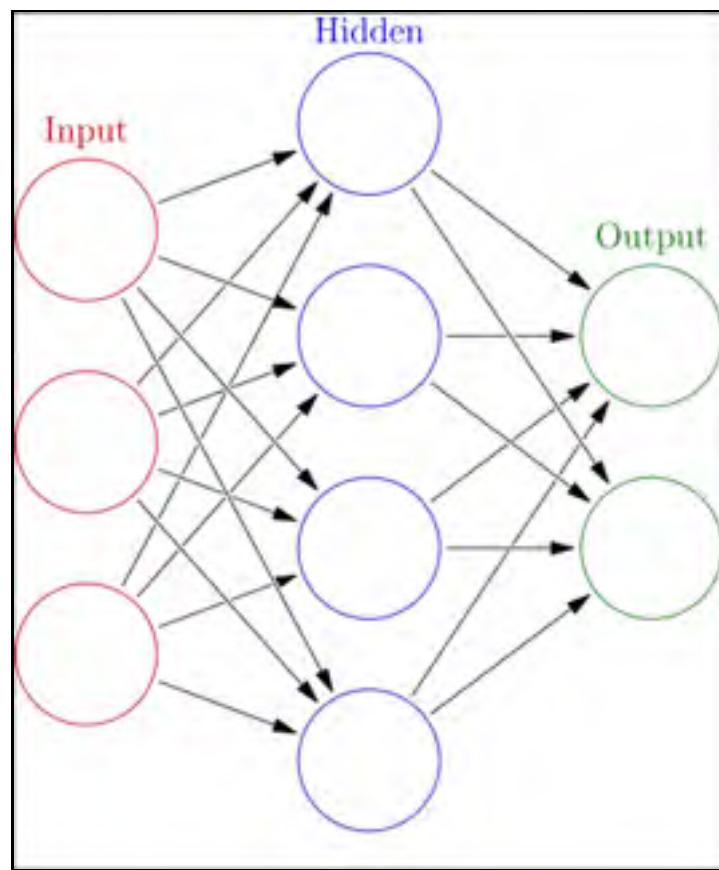


Figure 1.3 Colored neural network  
Tirée de Glosser.ca (2013) Repéré à  
[https://upload.wikimedia.org/wikipedia/commons/4/46/Colored\\_neural\\_network.svg](https://upload.wikimedia.org/wikipedia/commons/4/46/Colored_neural_network.svg)  
(Consulté en février 2020)

Dans la figure 1.3 sont illustrés les trois différents types de neurones d'un réseau de neurones artificiels classique.

### 1.3.2 Composantes

Afin de mieux comprendre le réseau de neurones, voici une description des composantes en détail.

### 1.3.3 Modèle de neurones

Le neurone est l'unité de traitement de base d'un réseau de neurones. Il est un modèle simple de traitement qui représente vaguement un vrai neurone biologique. Le modèle de neurones employé s'appelle le modèle du perceptron développé par Frank Rosenblatt en 1958. Voici le détail du modèle du neurone sur la figure ci-bas. Chaque cercle de la figure 1.3, à l'exception des neurones d'entrée en rouge, représente des neurones modélisés en détail à la figure 1.4.

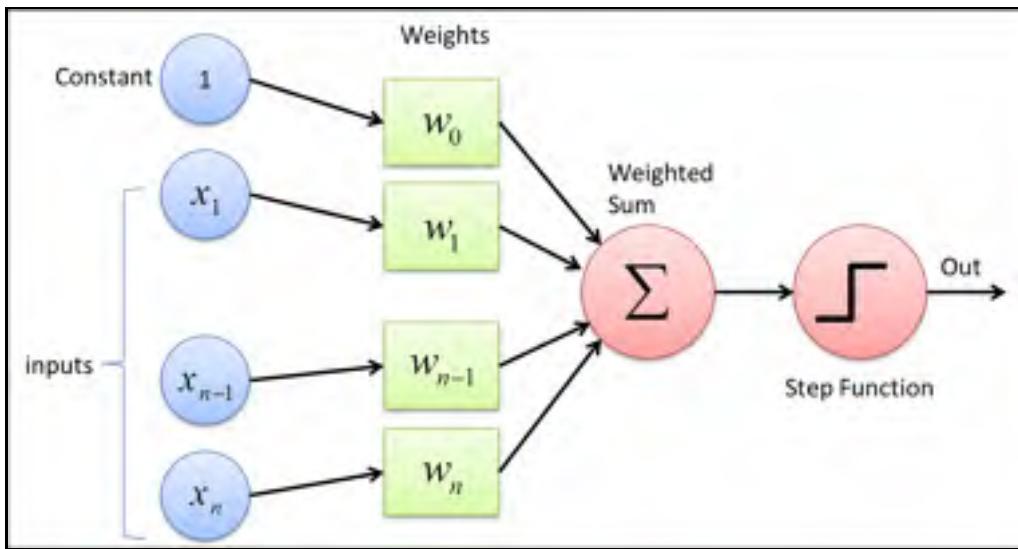


Figure 1.4 Perceptron  
Tirée de Sharma (2017)

Repéré à <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>  
(Consulté en mars 2020)

On peut représenter mathématiquement un neurone par l'équation suivante :

$$o_t = f \left( \sum_{i=1}^n w_i x_{i,t} \right) \quad (1.3)$$

Où  $o_t$  est la sortie du perceptron, les  $x_{i,t}$  sont les entrées, les  $w_i$  sont les poids et  $f$  est ce qu'on appelle la fonction d'activation sur les entrées du neurone. Souvent cette fonction est une

fonction logistique comme la fonction échelon ou tangente hyperbolique. Les symboles avec l'indice rappellent qu'il peut y avoir une valeur distincte pour un temps donné.

Nous allons utiliser la notation matricielle dans ce rapport pour alléger la notation. La même équation s'écrit donc comme suit :

$$o_t = f(\mathbf{w}\mathbf{x}_t) \quad (1.4)$$

Où  $\mathbf{w}$  est le vecteur des poids et  $\mathbf{x}_t$  est le vecteur d'entrée.

Avant l'ère des ordinateurs, le modèle du perceptron était implémenté de façon analogue. Voici le premier perceptron de Rosenblatt en 1958 réalisé entièrement avec des 200 fils pour les entrées et des potentiomètres pour les poids. Cette image n'illustre qu'un seul neurone!

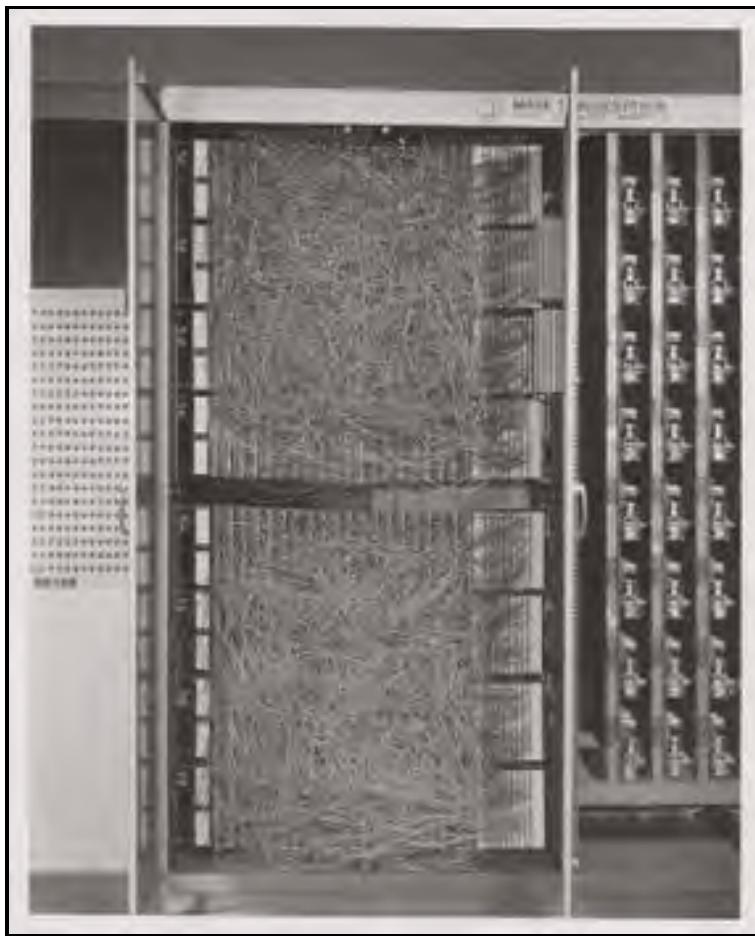


Figure 1.5 The Mark I perceptron, an early pattern recognition system (2018)  
Tirée de [https://upload.wikimedia.org/wikipedia/en/5/52/Mark\\_I\\_perceptron.jpeg](https://upload.wikimedia.org/wikipedia/en/5/52/Mark_I_perceptron.jpeg)  
(Consulté en mars 2020)

#### 1.3.4 Connexions et poids

Les connexions dans le perceptron sont représentées par un poids. Ces poids peuvent prendre des valeurs positives, négatives ou zéro. Lorsqu'un lien a un poids de zéro, on considère que le perceptron n'est pas relié à cette entrée. Chaque perceptron a également une entrée nommée  $x_0$  qui représente le chiffre 1. Cette entrée multipliée par le poids  $w_0$  représente le biais du perceptron. C'est un réglage interne au perceptron qui représente un "offset" des valeurs d'entrée du perceptron. Souvent il est représenté par la valeur  $B = w_0x_0$  et peut prendre une valeur négative.

### 1.3.5 Fonction de propagation

La fonction de propagation est la fonction  $\sigma$  appliquée à la sortie du perceptron sur la somme de toutes les entrées pondérées. Cette fonction est aussi nommée la fonction d'activation du neurone puisqu'elle représente une valeur comprise entre 0 et 1 ou -1 et 1 le plus souvent. Elle représente l'excitation du neurone lorsque les entrées parviennent à activer l'état du neurone en fonction du biais.

### 1.3.6 Apprentissage

Le perceptron apprend. C'est lui qui est à la base de l'apprentissage d'un réseau de neurones. L'apprentissage du perceptron se fait par un signal de rétropropagation (Backpropagation). Ce signal est la différence entre le signal de sortie attendu et celui obtenu. Ce signal sert à moduler les poids du neurone pour corriger la sortie du neurone et atteindre le signal voulu.

Il y a différentes façons de faire le signal de propagation, mais la plus simple est d'utiliser une fonction d'erreur  $E$  pour modifier les poids :

$$\mathbf{w} \rightarrow E(\sigma(\mathbf{w}\mathbf{x}_t), y_t) \quad (1.5)$$

Où  $y$  représente la sortie voulue pour une entrée  $x_t$  donnée.

$E$  pourrait être simplement la différence au carré :

$$E(y, y') = |y - y'|^2 \quad (1.6)$$

### 1.3.7 Modèle de réseau en Deep Learning

Si on prend le modèle de perceptron et on l'applique au réseau de neurones de la figure 1.3, on obtient le modèle suivant :

$$\begin{aligned} \mathbf{h}_t &= f(\mathbf{W}\mathbf{x}_t) \\ \mathbf{y}_t &= f(\mathbf{W}\mathbf{h}_t) \end{aligned} \quad (1.7)$$

$\mathbf{W}$  représente la matrice des vecteurs de poids  $\mathbf{w}$  ajoutés ensembles.  $\mathbf{h}_t$  représente le vecteur des neurones bleus de la couche cachée,  $\mathbf{x}_t$  représente le vecteur des neurones rouges de la couche d'entrée et  $\mathbf{y}_t$  représente le vecteur des neurones de sortie en vert. La deuxième équation est nécessaire pour représenter la couche de sortie.

Lorsqu'on parle de *Deep Learning*, on veut dire que le réseau de neurones a plusieurs couches comme celui de la figure 1.6:

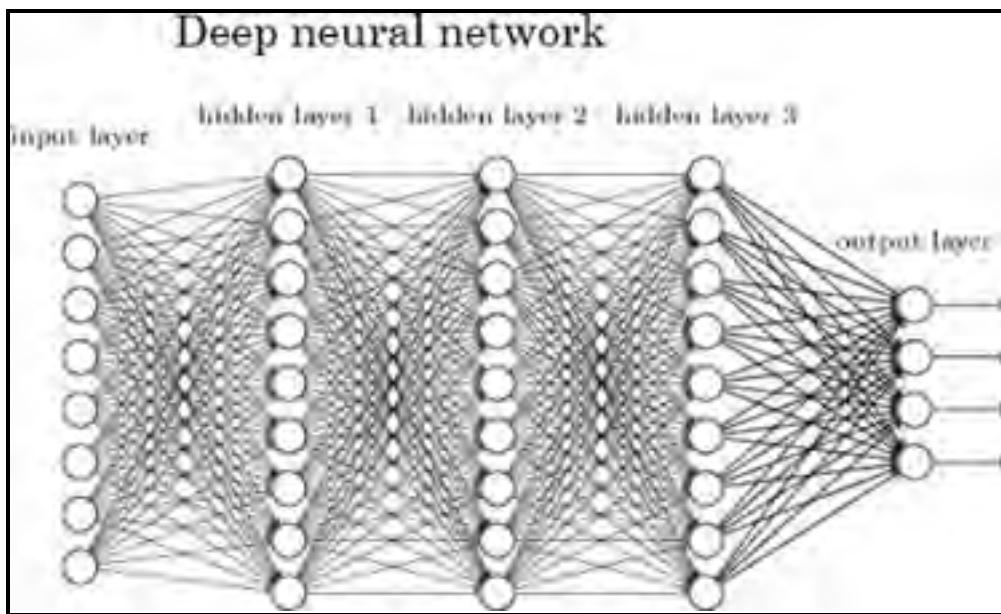


Figure 1.6 Deep neural network  
Tirée de Danish Ali (2016)

Repéré à <https://www.datasciencecentral.com/profiles/blogs/on-going-developments-and-outlook-for-deep-learning>  
(Consulté en février 2020)

On peut représenter ce genre de réseau avec les trois équations suivantes :

$$\begin{aligned}\mathbf{h}_t^1 &= f(\mathbf{W}^1 \mathbf{x}_t) \\ \mathbf{h}_t^l &= f(\mathbf{W}^l \mathbf{h}_t^{l-1}) \\ \mathbf{y}_t &= f(\mathbf{W}^L \mathbf{h}_t^L)\end{aligned}\tag{1.8}$$

Où  $L$  est le nombre de couches cachées. Les trois équations décrivent les trois types de neurones du réseau.

## 1.4 Réseau de neurones récurrent

### 1.4.1 Réseau de neurones “Fully Recurrent”

La variante de réseau de neurones récurrent la plus commune est le réseau de neurones “Fully Recurrent”. Ce type de réseau est très similaire au réseau de neurones classique, c'est-à-dire que les valeurs de sortie du réseau dépendent des valeurs des neurones de la couche cachée. Toutefois, à la différence du réseau de neurones classique, la valeur de sortie des réseaux de neurones récurrent dépend aussi de la valeur précédente des neurones de la couche cachée.

Les réseaux de neurones “Fully recurrent” sont des réseaux de neurones auxquels la valeur des neurones de la couche cachée dépend de la valeur précédente des autres neurones de la couche cachée comme l'illustre la figure 1.7.

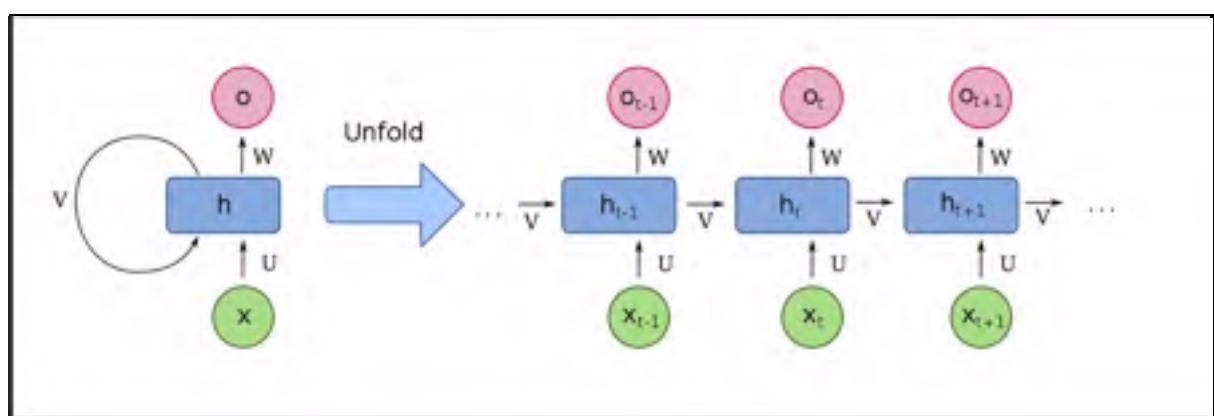


Figure 1.7 Recurrent neural network unfold  
Tirée de Fdeloche (2017)

Repéré à [https://commons.wikimedia.org/wiki/File:Recurrent\\_neural\\_network\\_unfold.svg](https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg)  
(Consulté en février 2020)

Les formules qui représentent ce type de réseau à une seule couche cachée sont les suivantes:

$$\begin{aligned}\mathbf{h}_t &= \tanh(\mathbf{W}\mathbf{h}_{t-1} + \mathbf{I}\mathbf{x}_t) \\ \mathbf{o}_t &= \text{softmax}(\mathbf{W}\mathbf{h}_t)\end{aligned}\tag{1.9}$$

Où la matrice  $\mathbf{I}$  représente la matrice de poids des neurones de la couche d'entrée vers les neurones de la couche cachée et la matrice  $\mathbf{W}$  représente la matrice des poids pour la valeur précédente des neurones de la couche cachée. Dans ce cas particulier, nous avons introduit la fonction tangente hyperbolique pour la couche cachée puisque c'est un exemple de fonction sigmoïdale rencontrée dans ce type de réseau.

Il est commun aussi de rencontrer la fonction *softmax* à la sortie d'un réseau. La fonction *softmax* a pour but de normaliser la sortie de sorte que toutes les valeurs de sortie sont bornées entre 0 et 1 et la somme de toutes les valeurs de sortie doit égaler 1. C'est une fonction très utile pour les problèmes de classification. La fonction *softmax* est définie comme suit :

$$y_t^i = \text{softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}\tag{1.10}$$

$$\mathbf{z} = \mathbf{W}\mathbf{h}_t\tag{1.11}$$

Dans le cas où le réseau de neurones a plusieurs couches cachées, on nomme ce type de réseau un “Deep Recurrent Neural Network” (Réseau de neurones récurrent profond). À la figure 1.8 est présenté un exemple de réseau de neurones récurrent profond :

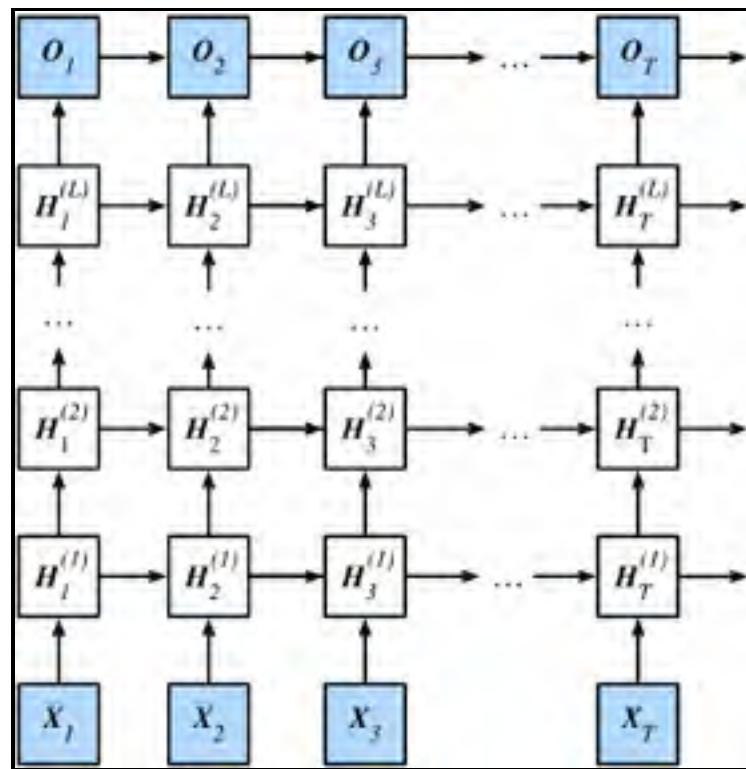


Figure 1.8 Architecture of a deep recurrent neural network  
Tirée de Zhang et al. (2019, p. 360)

Les formules qui représentent ce type de réseau avec plusieurs couches cachées sont les suivantes :

$$\begin{aligned}\mathbf{h}_t^1 &= f(\mathbf{W}^1 \mathbf{h}_{t-1}^1 + \mathbf{I}^1 \mathbf{x}_t) \\ \mathbf{h}_t^l &= \sigma(\mathbf{W}^l \mathbf{h}_{t-1}^l + \mathbf{I}^l \mathbf{h}_{t-1}^{l-1}) \\ \mathbf{o}_t &= \text{softmax}(\mathbf{W}^L \mathbf{h}_t)\end{aligned}\quad (1.12)$$

Où  $\mathbf{h}_t^l$  est la sortie des neurones de la couche no  $l$  et  $\mathbf{h}_{t-1}^l$  est la sortie de la valeur précédente de la couche cachée no  $l$ . Nous avons introduit la fonction  $\sigma$  qui est la fonction logistique sigmoïdale. Cette fonction est largement utilisée comme fonction d'activation des réseaux de neurones.

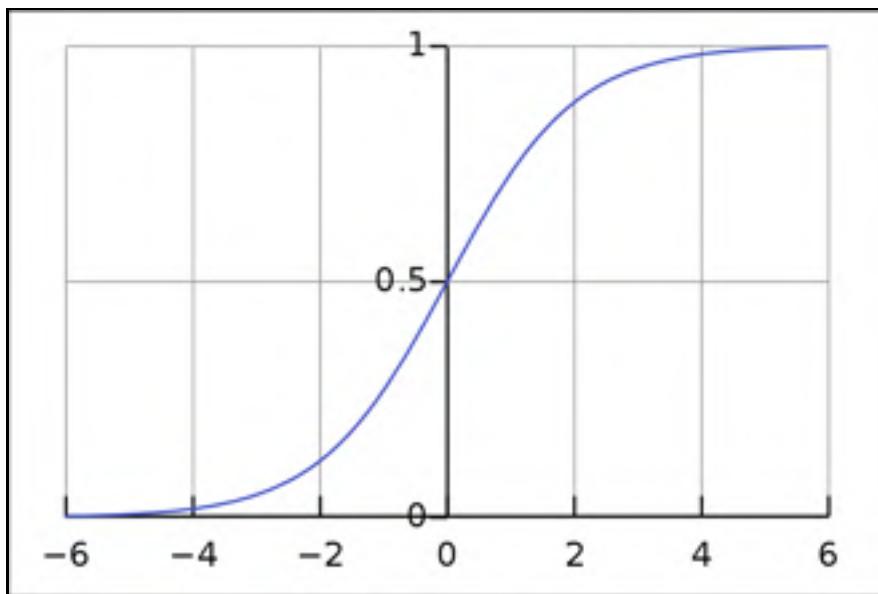


Figure 1.9 The logistic sigmoid function

Tirée de Qef (2008)

Repéré dans [https://en.wikipedia.org/wiki/Sigmoid\\_function#/media/File:Logistic-curve.svg](https://en.wikipedia.org/wiki/Sigmoid_function#/media/File:Logistic-curve.svg)  
(Consulté en février 2020)

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (1.13)$$

Cette fonction est définie entre 0 et 1 et elle est représentée sur la figure 1.9.

Les réseaux de neurones récurrent sont une forme de réseau de neurones qui démontre un comportement dans le temps puisqu'il y a une "mémoire" qui persiste dans le temps grâce à la liaison entre les couches cachées précédentes et suivantes.

### 1.4.2 Long Short-Term Memory RNN

Un des problèmes des réseaux de neurones récurrents est qu'il souffre du “vanishing gradient problem” (problème de gradient atténué). En théorie un réseau de neurones récurrent peut avoir une mémoire qui lui permet d'être influencé par des données très éloignées dans le passé. Par contre en pratique, à cause des représentations des données en mémoire, il y a une limite à cette influence du passé. C'est pour répondre à ce problème que les réseaux de neurones récurrents à longue mémoire à court terme (Long Short-Term Memory (LSTM)) ont été développés. Ces réseaux de neurones remplacent la couche cachée par un neurone plus complexe qui permet de préserver l'information plus longtemps.

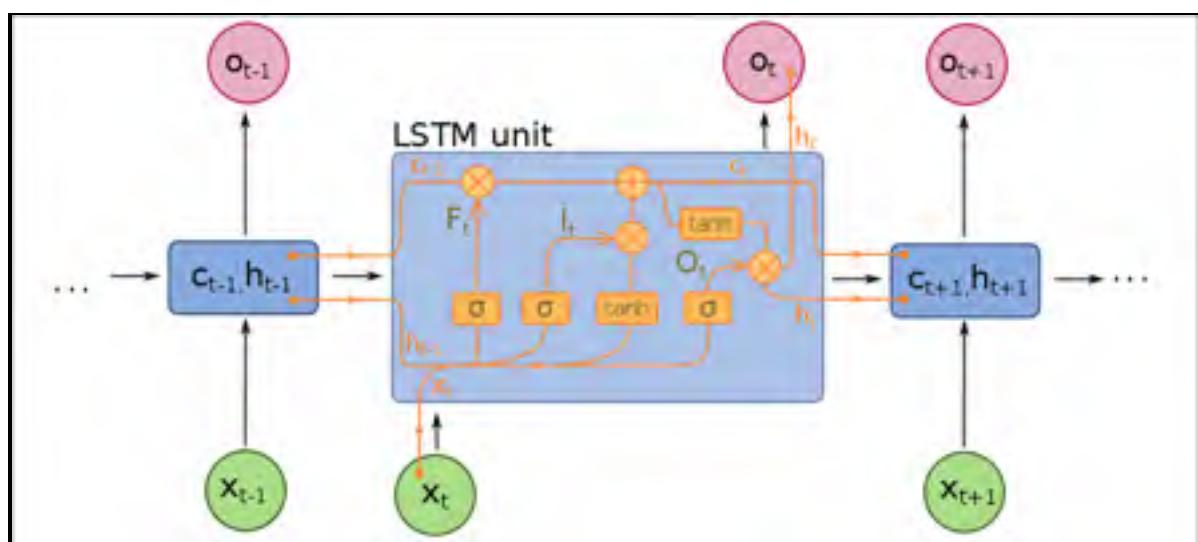


Figure 1.10 A diagram for a one-unit long short-term memory  
Tirée de Fdeloche (2017)

Repéré à <https://commons.wikimedia.org/w/index.php?curid=60149410>  
(Consulté en février 2020)

Voici un neurone avec Long Short-Term Memory illustré ci-dessus.

Les équations qui modélisent ce neurone sont les suivantes :

$$\begin{aligned}
 I_t &= \sigma(W^i h_{t-1} + I^i x_t) \\
 F_t &= \sigma(W^f h_{t-1} + I^f x_t) \\
 O_t &= \sigma(W^o h_{t-1} + I^o x_t)
 \end{aligned} \tag{1.14}$$

$$\mathbf{G}_t = \tanh(\mathbf{W}^g \mathbf{h}_{t-1} + \mathbf{I}^g \mathbf{x}_t)$$

$$\mathbf{C}_t = \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \mathbf{G}_t$$

$$\mathbf{h}_t = \tanh(\mathbf{O}_t \odot \mathbf{C}_t)$$

Où  $\mathbf{I}_t$ ,  $\mathbf{F}_t$ ,  $\mathbf{O}_t$ ,  $\mathbf{G}_t$  sont les vecteurs d'entrée, d'oubli (*forget gate*), de sortie et de la cellule respectivement.  $\mathbf{h}_t$  est le vecteur de la couche cachée de l'unité LSTM. La fonction logistique sigmoïdale est définie par  $\sigma$ , la multiplication élément par élément est représentée par  $\odot$ . La matrice des poids est représentée par la notation  $\mathbf{W}^i$ ,  $\mathbf{W}^f$ ,  $\mathbf{W}^o$ ,  $\mathbf{W}^g$  et les matrices du vecteur d'entrée sont représentées par la notation  $\mathbf{I}^i$ ,  $\mathbf{I}^f$ ,  $\mathbf{I}^o$ ,  $\mathbf{I}^g$ .

Le LSTM peut apprendre des dépendances temporelles plus loin dans le passé qu'un réseau de neurones récurrent classique.

## 1.5 Le réseau multi variable LSTM pleinement convolutionel

### 1.5.1 Architecture réseau

Les réseaux LSTM et ALSTM (Attentional LSTM) ont du succès à classifier des séries temporelles d'une seule variable. Dans le cas de plusieurs variables, (Karim, Majumdar, Darabi, & Hartford, 2019) proposent d'utiliser les réseaux multivariables ALSTM-FCN (MALSTM-FCN). Ce type de réseau augmente la capacité du ALSTM pour plusieurs variables.

Ce modèle est celui employé dans ce travail afin de permettre l'utilisation de plusieurs séries temporelles à l'apprentissage de la classification. Le but étant de classifier un ensemble de séries temporelles.

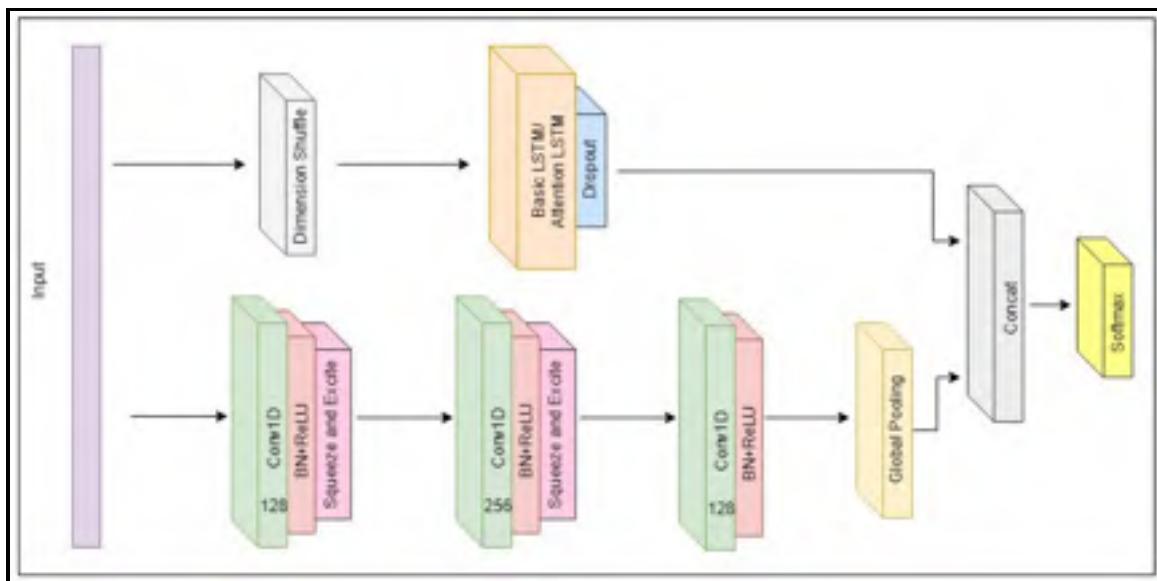


Figure 1.11 Multivariate attentional Long Short Term Memory  
Fully Convolutionel Network  
Tirée de Chothani, Bhalja & Parikh (2014)

Le modèle MALSTM-FCN consiste en un bloc pleinement convolutionel et un bloc LSTM. Chaque bloc convolutionel consiste en une couche convolutionnelle avec une taille de filtre de 128.

Cette méthode possède les étages suivants :

1. Deux étages identiques avec blocs convolutionnels possédant une taille de 128. Chaque bloc possède un étage de « Batch renormalisation » et une activation Relu. Les deux premiers blocs convolutionels se terminent avec un bloc « Squeeze and excite ».

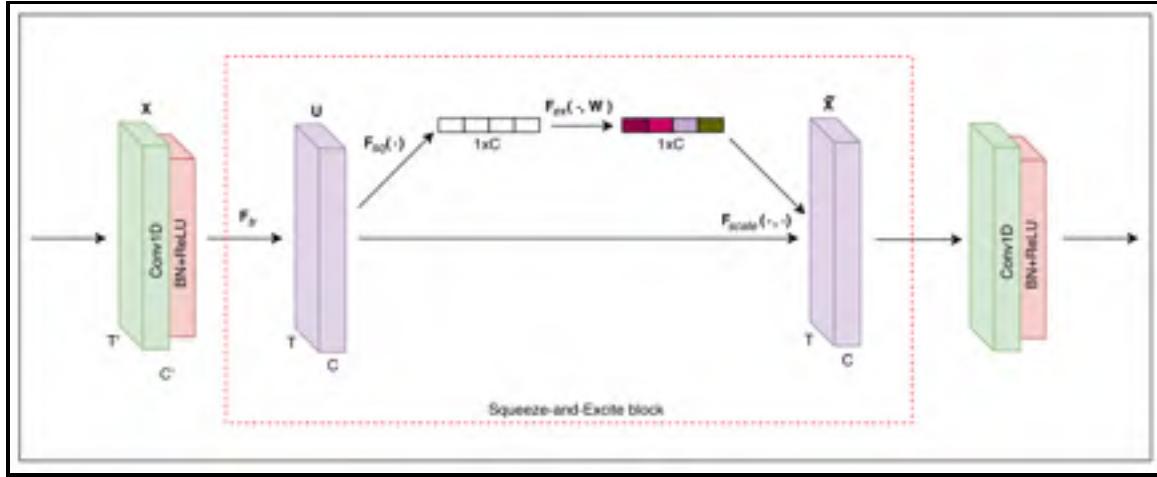


Figure 1.12 Block Squeeze-and-Excite  
Tirée de Chothani, Bhalja & Parikh (2014)

2. Le 3<sup>e</sup> bloc contient la même chose que les deux premiers sauf qu'il n'y a pas d'activation Relu.
3. Au-dessus des blocs convolutionnels se trouve une « dimension Shuffle » qui transpose la dimension temporelle des données d'entrée.
4. Par la suite les données passent par un réseau de neurones de base LSTM ou LSTM-attentionnel avant de passer à la section du « Dropout », qui consiste à ne retenir que la valeur d'activation d'un certain nombre aléatoire de neurones et d'inhiber les autres.
5. Lorsque ces informations sont complétées, le résultat est concaténé avec le profil de groupe.

## 1.6 Évaluation du modèle

### 1.6.1 Matrice de confusion

Un outil utile et efficace pour évaluer les problèmes de classification est la matrice de confusion (Fawaz et al, 2019). Dans une matrice de confusion, les colonnes représentent les prédictions, tandis que les lignes représentent la classification réelle. Dans notre étude, puisqu'il n'y a que deux classes, la matrice de confusion prend la forme d'une matrice deux

par deux. Nous identifions la classe de simulation instable par la valeur « 1 » et la classe de simulation stable par la valeur « 0 ».

Tableau 1.1 Matrice de confusion

		Cas prédit		
		0	1	Total prédit
Cas réel	0	Vrai positif VP	Faux négatif FN	VP+FN
	1	Faux positif FP	Vrai négatif VN	FP+VN
	Total réel	VP+FP	FN+VN	n

La matrice est construite avec les données obtenues de l'ensemble de test. Le modèle évalue l'ensemble de test et produit une prédiction qui est comparée avec la valeur réelle de l'ensemble de test. Ces valeurs sont ensuite récoltées dans la matrice de confusion et permettront d'identifier les cas de faux positif et de faux négatif. Un cas de faux positif serait que le modèle classe une simulation comme instable (1) lorsqu'en réalité elle serait stable (0).

La matrice de confusion est très utile pour obtenir différents taux que l'on peut calculer ainsi :

Tableau 1.2 Terminologie et dérivation de la matrice de confusion

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
$\frac{VP + VN}{n}$	$\frac{VP}{VP + FN}$	$\frac{VN}{VN + FP}$	$\frac{FP + FN}{n}$	$\frac{FP}{FP + VN}$	$\frac{FN}{FN + VP}$

La sensibilité est la probabilité que le modèle prédise correctement un classement positif (instable). La spécificité, quant à elle, donne la probabilité que le modèle prédise correctement un classement négatif (stable). Le taux d'erreur donne la quantité de faux positif ou de faux négatif sur l'ensemble des données. Le taux de faux positif donne la probabilité qu'un cas

classé positif (instable) soit erroné et le taux de faux négatif donne la probabilité qu'un cas classé négatif (stable) soit erroné.

## CHAPITRE 2

### RÉSEAU D'ÉTUDE

Cette section a pour but de détailler la méthode utilisée pour produire des données de simulation de réseau électrique. Dans ce chapitre il sera question du logiciel de simulation employé ainsi que du réseau d'étude pour générer les cas de simulations.

Par la suite, ce chapitre établira les différents paramètres employés pour le réseau ainsi que l'ensemble des simulations utilisées pour entraîner l'algorithme d'apprentissage. Cette section est inspirée des travaux de (Burhan, 2010) [10] pour la construction des données d'entraînement.

#### 2.1 Logiciel utilisé EMTP-RV

Le logiciel EMTP-RV est un logiciel développé par Hydro-Québec en 2003 et distribué par le DCG (Development Coordination Group). Il est employé par Hydro-Québec lors d'études transitoires électromagnétiques pour caractériser les protections et appareillages sur le réseau de transport ainsi que les équipements pour les automatismes de réseau. Ce logiciel est l'outil de choix pour obtenir beaucoup de données de précision sur les transitoires électromagnétiques lors de contingences. Il est donc le meilleur outil pour assembler une base de données d'apprentissage pour le relais de protection de synchronisme.

#### 2.2 Réseau identifié

Le réseau à l'étude est le réseau Anderson tiré de l'ouvrage P.M. Anderson *Power System Protection*. Ce réseau est un petit système de test qui illustre l'essentiel des problèmes de stabilité de puissance. Il permet d'illustrer le comportement typique d'un système sans introduire la complexité et la quantité de données requises pour un grand système interconnecté. Ce système comporte quatre unités de production ce qui résulte en trois modes naturels d'oscillation. Les zones 2 et 3 représentent un large réseau composé de plusieurs

générateurs, charges et lignes de transmission. Ces systèmes sont représentés par un générateur et une charge équivalente pour modéliser ces larges réseaux éloignés. La zone 1 représente le système d'intérêt pour une étude détaillée. Ce système a deux générateurs et trois centres de charges majeurs. Toutes les lignes de transport de la zone sont des lignes de 500 kV avec de la compensation série. La charge la plus importante se retrouve à la barre 7 et la tension à cette barre est contrôlée par une compensation statique de 1500 MVA positive ou négative. L'écoulement de puissance entre les régions représente des achats ou ventes de

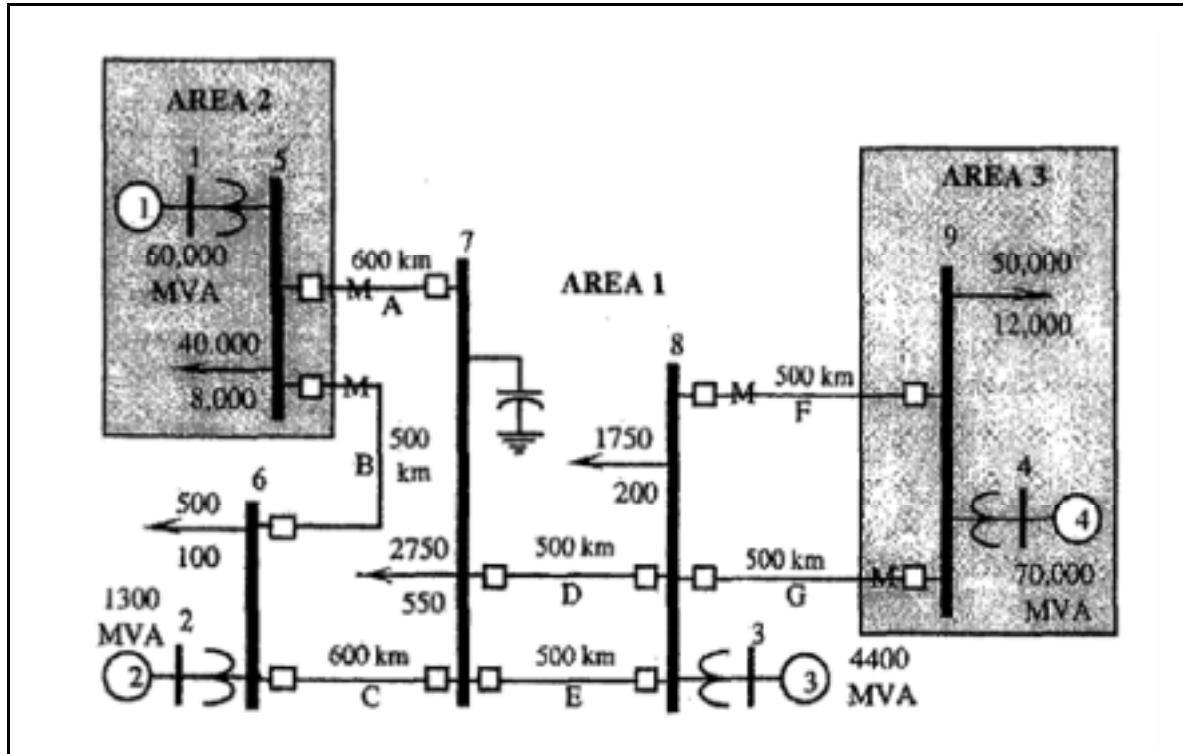


Figure 2.1 Réseau Anderson  
Tirée de Anderson (1998, p. 402)

puissance parmi ces régions. Les points de tension avec la lettre "M" indiquent le point de mesure entre les zones. La sous transmission et distribution sous-jacentes à travers la zone 1 ne sont pas représentées, mais sont considérées comme étant présentes. Par contre les centres de charges de la zone 1 sont suffisamment éloignés pour qu'on puisse assumer qu'il n'y a pas d'interconnexion à plus basse tension. Le système est testé en appliquant un défaut sur une ligne ou une barre de la zone 1.

### 2.3 Représentation du réseau d'étude sur EMTP

Le réseau d'étude a été modélisé dans le logiciel de simulation électromagnétique EMTP. Le réseau d'étude dans EMTP est illustré à la figure suivante. Dans les prochaines sections, il sera question de détailler les éléments de ce réseau. Il y a une seule différence entre le modèle du réseau Anderson et sa représentation sur EMTP. Dans la représentation sur EMTP toutes les lignes sont des lignes de 600 km à l'exception de la ligne b. La raison de ce changement est de permettre au modèle de plus facilement éprouver des problèmes de stabilité avec des lignes de plus grande longueur que sur le réseau Anderson d'origine.

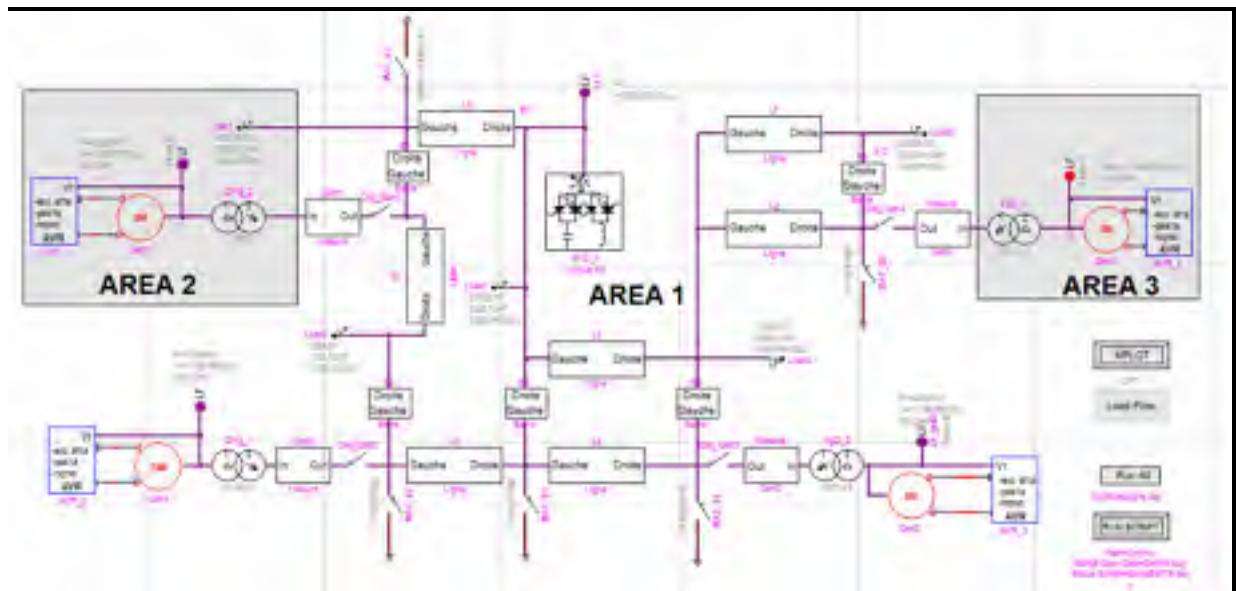


Figure 2.2 Réseau Anderson dans EMTP-RV

### **2.3.1 Description détaillée du réseau d'étude**

### 2.3.2 Générateurs

Cette section détaille les générateurs et leurs paramètres. Dans la zone 1 il y a deux générateurs : l'un de 1300 MVA et l'autre de 4400 MVA. Ils sont reliés entre eux par 1200 km

de lignes de transport. La zone 2 possède un générateur relié à la zone 1 par deux lignes de 600 km. Le dernier générateur se trouve dans la zone 3 et est aussi relié par deux lignes de 600 km. Les générateurs de la zone 2 et 3 sont donc séparés par 1800 km de lignes.

Voici un tableau qui illustre les différents paramètres des générateurs :

Tableau 2.1 Détail des générateurs

	Générateur 1	Générateur 4	Générateur 2	Générateur 3
Puissance[MVA]	60000	70000	1300	4400
Nb de pôles	54		64	
$R_a/X_1/X_0$ [pu]	0.03 / 0.149 / 0.113		0.011 / 0.14 / 0.127	
$X_d/X_d'/X_d''$ [pu]	1.028 / 0.34 / 0.253		1.1 / 0.323 / 0.299	
$X_q/X_q'/X_q''$ [pu]	0.544 / 0.5 / 0.298		0.416 / 0.4 / 0.212	
$T_{d0}'/T_{d0}''$ [s]	7.5 / 0.07		5.6 / 0.09	
$T_{q0}'/T_{q0}''$ [s]	7.5 / 0.09		5.6 / 0.09	
Constante H[pu]	4.1		3.2	

L'ensemble des paramètres représente les valeurs pour un test à circuit ouvert. Les génératrices sont connectées en étoile avec mise à la terre pour le neutre. Dans tous les cas, la tension aux bornes des génératrices est de 13.8 kV RMS ligne-ligne.

### 2.3.3 Transformateurs

Concernant les transformateurs, ils ont une configuration Triangle/étoile avec neutre mis à la terre. La puissance nominale de chaque transformateur est égale à la puissance de la machine correspondante.

Au primaire du transformateur, la tension est de 13.8 kV RMS ligne-ligne et au secondaire la tension est de 500 kV RMS ligne-ligne. Voici les caractéristiques des transformateurs illustrés dans le tableau ci-dessous.

Tableau 2.2 Détail des transformateurs

	<b>Transfo.</b> <b>1</b>	<b>Transfo.</b> <b>2</b>	<b>Transfo.</b> <b>3</b>	<b>Transfo.</b> <b>4</b>
<b>Puissance[MVA]</b>	60000	1300	1400	70000
<b>Résistance au primaire [pu]</b>	0.001	0.001	0.001	0.001
<b>Résistance au secondaire[pu]</b>	0.001	0.001	0.001	0.001
<b>Réactance au primaire [pu]</b>	0.001	0.001	0.001	0.001
<b>Réactance au secondaire[pu]</b>	0.11	0.082	0.082	0.09

### 2.3.4 Lignes

La tension nominale des lignes employées dans le modèle du réseau Anderson est de 500 kV. La longueur des lignes est de 500 km pour la ligne C tandis que la longueur est de 600 km pour les lignes A, B, D, E, F et G. Toutes les lignes sont de type à paramètres distribués. Ce type de modèle de ligne permet une grande rapidité de simulation avec une faible erreur si la fréquence ne varie pas énormément. Les paramètres utilisés pour définir la ligne sont la capacité linéique, la résistance et l'inductance dans leurs composantes homopolaires et directes. Les paramètres utilisés sont illustrés dans le tableau suivant :

Tableau 2.3 Détail des lignes

<b>Séquence</b>	<b>Longueur [km]</b>	<b>R' [Ω]</b>	<b>L'[H]</b>	<b>C'[F]</b>
<b>Homopolaire</b>	300	0.35936	0.00266216	3.37889e-06
<b>Direct</b>	300	0.01613	0.0009218	1.23e-08

Voici le sous-circuit de ligne illustré dans la figure suivante. Il contient les composantes suivantes : deux lignes de type CP, quatre interrupteurs triphasés en série, un compensateur série modélisé par une capacité triphasée, deux interrupteurs triphasés de mise à terre avec en série élément RLC et trois interrupteurs monophasés en série.

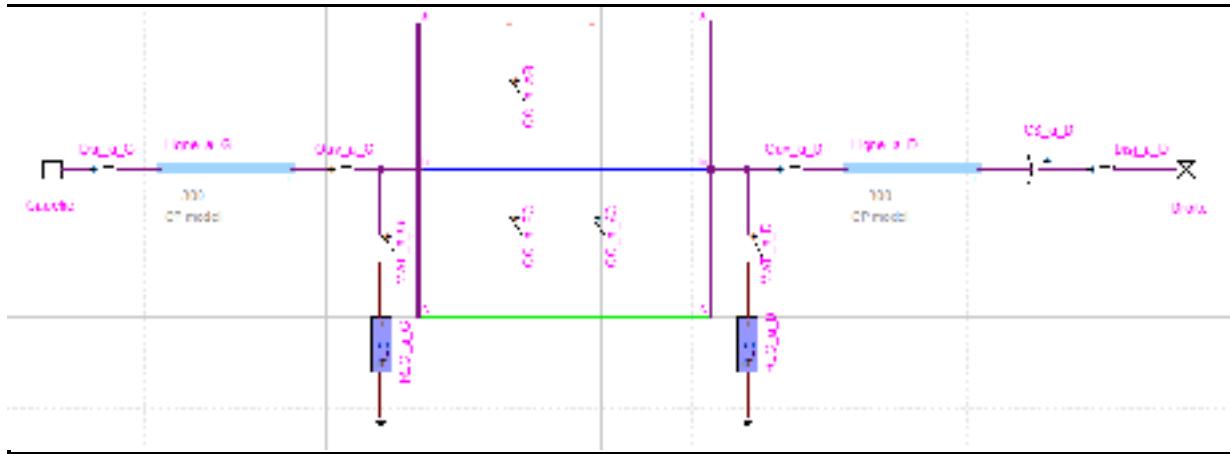


Figure 2.3 Détail d'une ligne sur EMTP-RV

Voici un sous-circuit conservé en librairie et réutilisé pour l'ensemble des lignes :

Ce sous-circuit permet, grâce aux nombreux interrupteurs présents, de produire un vaste choix de courts-circuits possibles, par exemple un court-circuit interphase ou à la terre ou une ouverture de ligne.

Grâce à certains paramètres comme la longueur des lignes aux deux extrémités, il est possible de choisir l'emplacement du court-circuit sur la ligne et en modifier les paramètres. Dans notre étude, les courts-circuits se situeront au milieu de la ligne.

Ce type de sous-circuit comporte des désavantages. Par exemple en segmentant la ligne en deux, nous augmentons le nombre de lignes et donc le nombre de calculs que le simulateur doit exécuter et donc des simulations plus longues et le pas de calcul d'une taille minimale égale à la constante de temps de la ligne. Néanmoins, la grande modularité de ce sous-circuit compense amplement pour ces désavantages.

Dans cette étude, l'acquisition se fait à une fréquence de 3840 Hz, pour un temps d'échantillonnage de 260 us.

Finalement, les lignes incluent une compensation série de 30% avec un condensateur à chaque ligne avec une capacité 0.1824 F et de 0.152 F pour les lignes de 600 km et de 500 km de long respectivement.

### 2.3.5 La compensation statique

Le compensateur statique d'énergie réactive est un élément pouvant servir à injecter ou retirer de l'énergie réactive d'une barre d'un réseau de grande puissance. Cela permet d'avoir un contrôle de tension à la barre à laquelle il est rattaché. Dans cette étude, nous fixons une tension de référence de 1.02 pu et le compensateur statique se charge, grâce à son contrôleur, d'injecter jusqu'à 1000 kVAR ou d'absorber jusqu'à 500 kVAR pour pouvoir maintenir la tension de référence. Les gains proportionnels et intégrateurs du contrôleur sont de 12 et 2800. Les autres paramètres sont les paramètres définis par défaut.

### 2.3.6 Les charges

Pour la simulation, les charges employées sont de type PQ imposant à la charge la puissance active et réactive consommée par celle-ci ainsi que la tension nominale, soit 500 kV RMS ligne-ligne. Le tableau suivant décrit en détail les charges :

Tableau 2.4 Détail des charges

Nom	P [MW]	Q [MVAR]	Zone	Barre connectée
<b>Load 1</b>	40000	8000	2	Barre 1
<b>Load 2</b>	500	100	1	Barre 2
<b>Load 3</b>	2750	550	1	Barre 3
<b>Load 4</b>	1750	200	1	Barre 4
<b>Load 5</b>	50000	12000	3	Barre 5

### 2.3.7 Sous-circuit de mesures

Le sous-circuit présenté dans la figure suivante est en série après le secondaire de chacun des transformateurs des génératrices. Il permet d'obtenir les mesures suivantes :

- les courants et tensions temporelles et RMS;
- les puissances actives et réactives temporelles et RMS;
- les composantes symétriques des courants et tensions temporelles et RMS ;
- les fréquences de chaque phase.

Au total cela représente trente-neuf mesures prises pour chaque machine soit un total de 156 mesures.

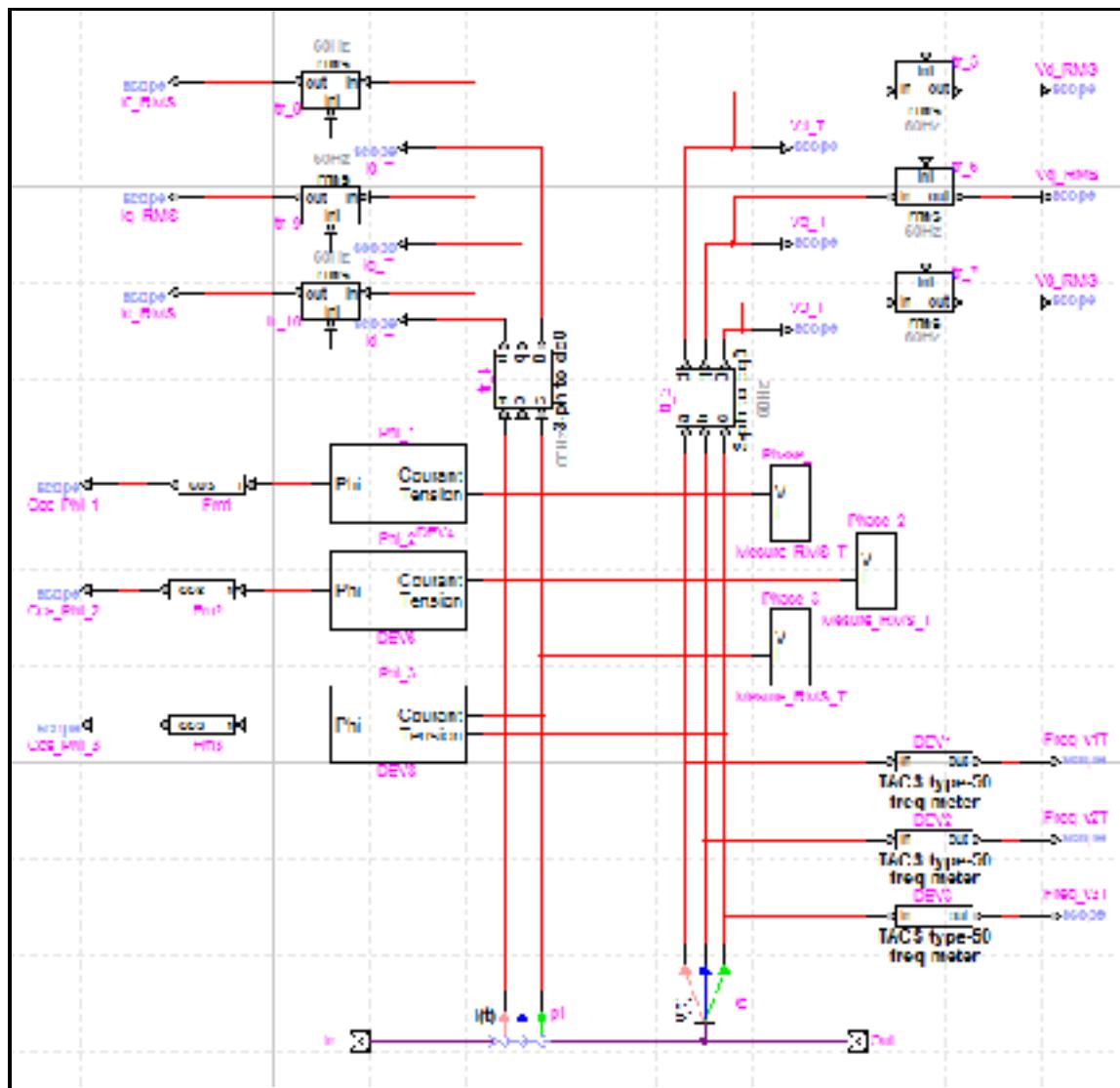


Figure 2.4 Détail du sous-circuit de mesures

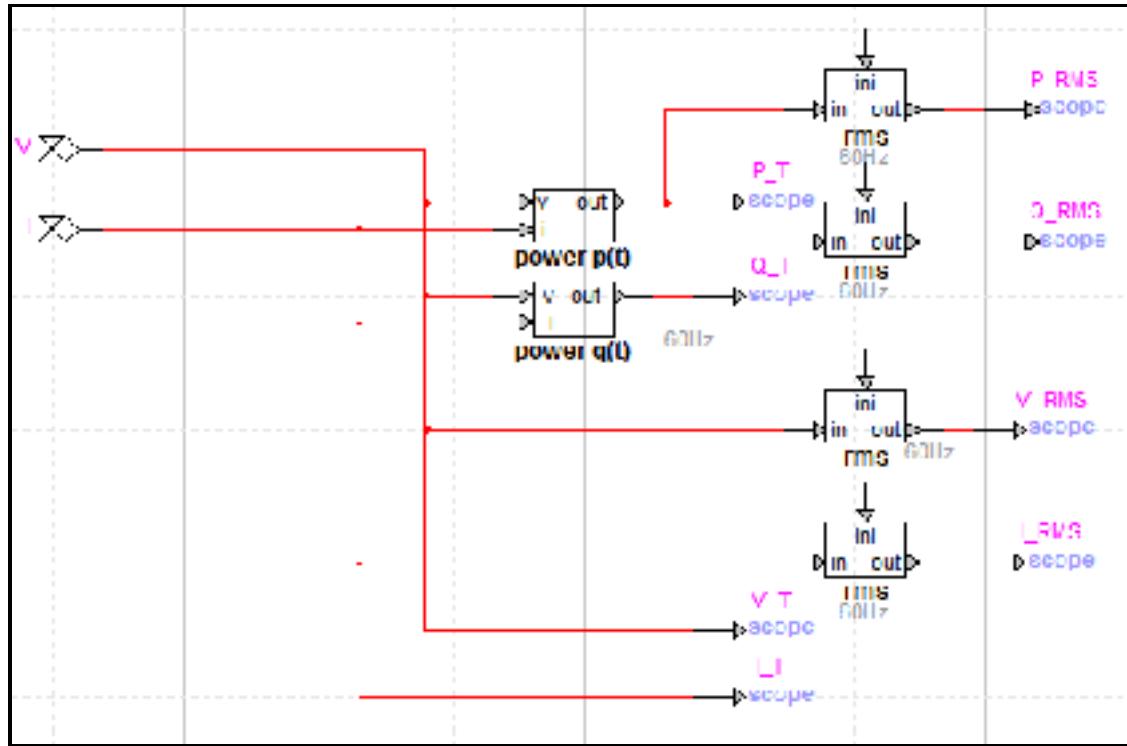


Figure 2.5 Détail du sous-circuit de mesures pour une phase

## 2.4 Simulations

### 2.4.1 Load-Flow

Pour construire la base de données, trois Load-Flow différents ont été simulés. Lorsqu'une machine est sélectionnée comme bus de référence, les autres machines sont situées sur des barres de type PV et sont réglées pour fournir le maximum de puissance active. Le but de fournir le maximum de puissance active est de rapprocher le système vers la limite de la stabilité. L'objectif étant de permettre aux machines de décrocher plus facilement lors d'une perturbation.

### 2.4.2 Zone 1

Dans la zone 1, le bus Slack ou  $V\delta$  se situe aux bornes de la génératrice 4. Tous les autres bus sont de type PV. La puissance est triphasée et les valeurs de tension sont de type RMS ligne-ligne.

Tableau 2.5 Générateur pour le Load-Flow 1

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	PV	PV	$V\delta$	PV
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	40000 [MW]	1000 [MW]	0 [°]	52000 [MW]

Tableau 2.6 Résultat du Load-Flow 1

Device	Type	Vabc (kVRMSSL,deg)		P [W]	Q [VAR]	Eabc (kVRMSLL,deg)		Iabc (A,deg)	
<b>LF_gen3</b>	Slack	1.38E+01	-2.29E-10	5.24E+09	6.04E+08	1.97E+01	3.93E+01	3.12E+05	-6.57E+00
		1.38E+01	-1.2E+02			1.97E+01	-8.07E+01	3.12E+05	-1.27E+02
		1.38E+01	1.2E+02			1.97E+01	1.59E+02	3.12E+05	1.13E+02
<b>LF1_svc</b>	PVbus	5.00E+03	-2.18E+01	3.10E+00	4.46E+08	5.00E+02	-2.18E+01	7.28E+02	-1.12E+02
		5.00E+03	-1.42E+02			5.00E+02	-1.42E+02	7.28E+02	1.28E+02
		5.00E+03	9.82E+01			5.00E+02	9.82E+01	7.28E+02	8.20E+00
<b>LF_gen1</b>	PVbus	1.38E+01	-4.23E+01	4.00E+10	1.05E+10	1.73E+01	-1.76E+01	2.45E+06	-5.70E+01
		1.38E+01	-1.62E+02			1.73E+01	-1.38E+02	2.45E+06	-1.77E+02
		1.38E+01	7.78E+01			1.73E+01	1.02E+02	2.45E+06	6.30E+01
<b>LF_gen2</b>	PVbus	1.38E+01	-2.90E+01	1.30E+09	-3.13E+08	1.54E+01	1.38E+01	7.91E+04	-1.54E+01
		1.38E+01	-1.49E+02			1.54E+01	-1.06E+02	7.91E+04	-1.35E+02
		1.38E+01	9.10E+01			1.54E+01	1.34E+02	7.91E+04	1.05E+02
<b>LF_gen4</b>	PVbus	1.38E+01	-2.61E+01	4.90E+10	1.52E+10	1.78E+01	-9.61E-01	3.04E+06	-4.34E+01
		1.38E+01	-1.46E+02			1.78E+01	-1.21E+02	3.04E+06	-1.63E+02
		1.38E+01	9.39E+01			1.78E+01	1.19E+02	3.04E+06	7.66E+01

On remarque que la génératrice 3 est en dessous de sa puissance nominale et la génératrice 2 consomme de la puissance réactive alimentée par le compensateur statique qui maintient la tension à 1.02 pu.

### 2.4.3 Zone 2

La deuxième simulation de Load-Flow est réalisée en situant la barre Slack aux bornes de la génératrice 1 :

Tableau 2.7 Générateur pour Load-Flow 2

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	V $\delta$	PV	PV	PV
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	0 [°]	1300 [MW]	4400 [°]	49000 [MW]

Tableau 2.8 Résultat pour Load-Flow 2

Device	Type	Vabc (kVRMSSL,deg)		P [W]	Q [VAR]	Eabc (kVRMSLL,deg)		Iabc (A,deg)	
LF_gen1	Slack	1.38E+01	1.24E-11	4.08E+10	1.08E+10	1.74E+01	2.50E+01	2.50E+06	-1.48E+01
		1.38E+01	-1.20E+02			1.74E+01	-9.49E+01	2.50E+06	-1.35E+02
		1.38E+01	1.20E+02			1.74E+01	1.45E+02	2.50E+06	1.05E+02
LF1_svc	PVbus	5.00E+03	-3.67E+00	1.10E-01	9.42E+07	5.00E+02	-3.68E+00	1.54E+02	-9.37E+01
		5.00E+03	-1.24E+02			5.00E+02	-1.24E+02	1.54E+02	1.46E+02
		5.00E+03	1.16E+02			5.00E+02	1.16E+02	1.54E+02	2.63E+01
LF_gen2	PVbus	1.38E+01	6.77E-01	1.30E+09	-2.31E+08	1.58E+01	4.18E+01	7.81E+04	1.07E+01
		1.38E+01	-1.19E+03			1.58E+01	-7.81E+01	7.81E+04	-1.09E+02
		1.38E+01	1.21E+02			1.58E+01	1.62E+02	7.81E+04	1.31E+02
LF_gen3	PVbus	1.38E+01	-2.35E+00	4.40E+09	-8.77E+07	1.71E+01	3.52E+01	2.60E+05	-1.20E+00
		1.38E+01	-1.22E+02			1.71E+01	-8.48E+01	2.60E+05	-1.21E+02
		1.38E+01	1.18E+02			1.71E+01	1.55E+02	2.60E+05	1.19E+02
LF_gen4	PVbus	1.38E+01	-2.71E+01	4.90E+10	1.52E+10	1.78E+01	-2.01E+00	3.04E+06	-4.44E+01
		1.38E+01	-1.47E+02			1.78E+01	-1.22E+02	3.04E+06	-1.64E+02
		1.38E+01	9.29E+01			1.78E+01	1.18E+02	3.04E+06	7.56E+01

On peut constater que les quatre génératrices sont proches de leur puissance nominale.

Nous voyons que toutes les machines sont à la limite de la surcharge.

#### 2.4.4 Zone 3

La dernière simulation de Load-Flow est accomplie en situant la barre Slack aux bornes de la génératrice 4.

Tableau 2.9 Générateur pour Load-Flow 3

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	PV	PV	PV	V $\delta$
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	42000 [MW]	1000 [MW]	4000 [MW]	0 [°]

Tableau 2.10 Résultat pour Load-Flow 3

Device	Type	Vabc (kVRMSSL,deg)		P [W]	Q [VAR]	Eabc (kVRMSSL,deg)		Iabc (A,deg)
<b>LF_gen4</b>	Slack	1.38E+01	-5.41E-12	4.98E+10	1.51E+10	1.79E+01	2.55E+01	3.08E+06 -1.68E+01
		1.38E+01	-1.20E+02			1.79E+01	-9.45E+01	3.08E+06 -1.37E+02
		1.38E+01	1.20E+02			1.79E+01	1.46E+02	3.08E+06 1.03E+02
<b>LF1_svc</b>	PVbus	5.00E+02	-1.21E+01	2.25E+00	4.21E+08	5.00E+02	-1.21E+01	6.87E+02 -1.02E+02
		5.00E+02	-1.32E+02			5.00E+02	-1.32E+02	6.87E+02 1.38E+02
		5.00E+02	1.08E+02			5.00E+02	1.08E+02	6.87E+02 1.79E+01
<b>LF_gen1</b>	PVbus	1.38E+01	-3.26E+01	4.00E+10	1.05E+10	1.73E+01	-7.90E+00	2.45E+06 -4.73E+01
		1.38E+01	-1.53E+02			1.73E+01	-1.28E+02	2.45E+06 -1.67E+02
		1.38E+01	8.74E+01			1.73E+01	1.12E+02	2.45E+06 7.27E+01
<b>LF_gen2</b>	PVbus	1.38E+01	-1.93E+01	1.30E+09	-3.13E+08	1.54E+01	2.35E+01	7.91E+04 -5.75 E+00
		1.38E+01	-1.39E+02			1.54E+01	-9.65E+01	7.91E+04 -1.26E+02
		1.38E+01	-1.01E+02			1.54E+01	1.44E+02	7.91E+04 1.14E+02
<b>LF_gen3</b>	PVbus	1.38E+01	8.42E+00	4.40E+09	2.59E+08	1.78E+01	4.44E+01	2.61E+05 5.06E+00
		1.38E+01	-1.12E+02			1.78E+01	-7.56E+01	2.61E+05 -1.15E+02
		1.38E+01	1.28E+02			1.78E+01	1.64E+02	2.61E+05 1.25E+02

Nous pouvons constater que la compensation statique est à 45% de sa capacité maximale 1000 MVAR et que l'ensemble des génératrices fournissent de la puissance réactive sauf la génératrice 2.

#### **2.4.5 Les types de défauts**

Concernant les défauts, ils sont produits par 60 simulations différentes. Chaque simulation varie avec les paramètres suivants :

- durée du court-circuit de 12 cycles – 6 cycles ou 200 ms à 100 ms;
- type de court-circuit soit, monophasé, biphasé, triphasé;
- lieu du court-circuit aux lignes A, B, C, D, F ou aux barres B1, B2, B3, B4, B5.

Les défauts sont éliminés après 6 à 12 cycles et aucune ligne ou barre demeure ouverte.

Ces 60 défauts différents sont répétés pour chacun des trois Load-Flow réalisés, portant ainsi le nombre de simulations à 180 avec un pas de temps de 260 us, soit un taux d'échantillonnage de 3840 Hz. La constante de temps de la ligne la plus courte impose ce pas de temps. Les simulations sont d'une durée de 20 secondes.

#### **2.4.6 Script des simulations**

Grace à EMTP-RV, nous avons la possibilité de scripter et d'automatiser les simulations à exécuter à partir d'un fichier .DWJ. Cela nous permet de faire des modifications sur des composantes et des attributs de la simulation et d'exécuter un grand nombre de simulations en série. Parmi les options qui s'offrent à nous pour les scripts, nous pouvons lancer une série de simulations en modifiant le temps d'ouverture et de fermeture des disjoncteurs et sectionneurs, modifier entre chaque simulation la longueur des lignes ou modifier la valeur des composantes  $\pi$  des lignes ou des résistances ou réactances des charges. Dans le cas de cette étude, nous utilisons un fichier .DWJ uniquement, mais il est exécuté trois fois puisque les simulations sont exécutées, ayant au préalable changé la barre de référence et calculé le Load-Flow correspondant. Le fichier .DWJ est présenté en annexe de Burhan (2012).

## **CHAPITRE 3**

### **TRAITEMENT DES DONNÉES**

Dans ce chapitre, en premier lieu, nous expliquerons la façon d'importer les simulations depuis EMTP-RV. Dans la section suivante, nous définirons les critères de discrimination entre les simulations stables et instables.

Par la suite, un sous-chapitre expliquera comment les sous-ensembles d'apprentissage et de test sont créés.

Pour terminer, nous décrirons comment les données sont exportées en format .mat vers l'application sur Python.

#### **3.1 Importations des données sous MATLAB**

Grâce à l'outil de simulation EMTP-RV, nous pouvons exporter des données vers MATLAB en différents formats. Les formats possibles incluent le format ASCII ou un format de fichier MATLAB (.mat).

MATLAB est un outil de choix pour la manipulation d'un grand nombre de données. Il est privilégié lors de la manipulation de gros fichiers et c'est la raison pour laquelle il est utilisé.

Les fichiers de simulation d'EMTP sont en format .mda et sont de très grandes tailles. Chaque fichier représente environ 260 Méga-octets et contient 120 724 lignes de long et 156 colonnes de large, ce qui correspond à 20 secondes de simulation. L'ensemble des fichiers générés par EMTP représente 44,5 Gigaoctets de données.

Grâce à un outil intégré dans ETMP, les données en format .mda sont transférées vers le format .mat pour l'importation dans MATLAB. Durant cette étape, seulement un sous-ensemble des données est importé. Ce sous-ensemble comprend :

- courants et tensions temporels par phase (6 mesures par machine);
- puissances actives et réactives temporelles par phase (6 mesures par machine).

À cela nous ajoutons les vitesses de rotation et les angles des machines :

- vitesses de rotation et angles temporels (2 mesures par machine).

Au total cela fait 56 colonnes et toujours 120 724 lignes, ceci réduit la taille des fichiers à 88 mégaoctets. L'ensemble des fichiers est désormais d'une taille de 14,8 gigaoctets.

Lorsque le fichier est converti par EMTP en format (.mat), le fichier est importé sur MATLAB. Voici les éléments du Workspace de MATLAB lorsque le fichier est importé tel qu'illustré à la figure suivante. La variable Y du tableau contient 56 variables et 120 724 lignes. La variable Ylabels contient 56 noms des variables associées aux cellules Y. La variable X contient 56 variables de temps et 120 724 lignes et la variable Xlabels contient les 56 noms de variables de temps.

Workspace	
Name	Value
X	1x56 cell
Xlabels	1x56 cell
Y	1x56 cell
Ylabels	1x56 cell

Figure 3.1 Workspace de MATLAB

Une fois importé dans le Workspace de MATLAB, nous créons une seule variable de temps t et nous supprimons les 55 autres variables X et Xlabels.

Le format des données est converti du format double au format simple. Ensuite les données sont séparées en mesures mécaniques qui contiennent les angles et les vitesses des générateurs

VectMec et quatre matrices qui contiennent les variables de mesures électriques pour chaque générateur : VectElectGen1, VectEletGen2, VectElectGen3 et VectElectGen4.

En terminant, pour permettre d'obtenir une prédition avant les 20 secondes de simulation, nous ne garderons que 1.2 seconde des mesures électriques. Ce temps correspond à la durée du défaut.

Ces nouvelles variables sont sauvegardées et réutilisent l'ancien fichier .mat pouvant ainsi réduire sa taille à 8 MB par fichier, soit au total 1.41 GB pour l'ensemble des simulations.

## **3.2 Classemement des simulations en cas stables et instables**

### **3.2.1 Introduction théorique au centre d'inertie**

Lorsqu'un réseau électrique subit une perturbation, que ce soit lors de la perte d'une ligne ou d'une charge importante ou lors d'un défaut, les génératrices tendent soit à accélérer ou à décélérer. Très souvent les génératrices reviennent à leur point d'équilibre après une perturbation à l'issue d'une période transitoire. Mais lorsque la perturbation est importante et dépendamment de l'état du réseau suivant la contingence, il est possible que ce nouveau point d'équilibre soit très éloigné du régime nominal.

Pour déterminer quelles génératrices vont quitter suffisamment le régime nominal pour les déclarer instables dans nos simulations, nous avons besoin d'un critère de stabilité pour une génératrice donnée.

Suivant les formules suivantes, selon (Burhan, 2012), (Sauer & Pai, 1998), (Hashin et al., 2010), nous pouvons établir une mesure de l'écart des angles et des vitesses des génératrices du système par rapport à l'ensemble des génératrices.

L'angle moyen pondéré par l'inertie des machines se nomme angle COI (Center of Inertia) et est défini comme suit :

$$\delta_{COI} \triangleq \frac{1}{M_T} \sum_{i=1}^m M_i \delta_i \quad (3.1)$$

Où  $\delta_{COI}$  représente l'angle moyen,  $M_i$  est le moment d'inertie d'une génératrice donnée et  $M_T$  est la somme de tous les moments d'inertie des génératrices d'un système défini par :

$$M_T \triangleq \sum_{i=1}^m M_i \quad (3.2)$$

Finalement, l'angle d'une génératrice  $j$  référencé en fonction du centre d'inertie COI des angles est défini par la formule suivante :

$$\widehat{\delta}_j \triangleq \delta_j - \delta_{COI} \quad (3.3)$$

Nous pouvons également définir une mesure similaire avec les vitesses des génératrices. Le COI des vitesses ainsi que la référence des vitesses d'une génératrice  $j$  en fonction du COI sont définis par les trois formules suivantes :

$$\omega_{COI} \triangleq \frac{1}{M_T} \sum_{i=1}^m M_i \omega_i \quad (3.4)$$

$$M_T \triangleq \sum_{i=1}^m M_i \quad (3.5)$$

$$\widehat{\omega}_j \triangleq \omega_j - \omega_{COI} \quad (3.6)$$

### 3.2.2 Critères d'instabilité

Selon Kundur (1994), on peut déterminer si une génératrice est stable ou instable en fonction des critères suivants :

- si une génératrice a un centre d'inertie d'angle inférieur +/- pi et que la génératrice a un centre d'inertie de vitesse lent, alors la génératrice est considérée stable;
- si une génératrice a un centre d'inertie d'angle supérieur +/- pi et que la génératrice a un centre d'inertie de vitesse rapide, alors la génératrice est considérée instable.

Pour appliquer ces critères, il faut définir un seuil minimal de vitesse auquel on considère qu'une génératrice devient instable.

À des fins de comparaison des résultats avec les travaux précédents, nous avons jugé bon de conserver les seuils d'angle et de vitesse de (Burhan, 2012). Ces seuils sont basés sur les simulations instables détectées par le logiciel EMTP-RV. Lorsqu'une simulation est détectée instable, le logiciel arrête la simulation. Ceci offre une méthode pour déterminer quelle simulation peut être déclarée instable. Ceci nous permet de définir un minimum du COI de vitesse des simulations divergentes et de l'utiliser comme seuil de vitesse. Le même principe s'applique pour le COI $\Omega$  d'angle.

La figure suivante est celle qui nous permet de définir le seuil grâce à l'algorithme de (Burhan, 2012). Les simulations marquées d'un X sont celles donc le COIA était supérieur à pi. Les deux barres bleues et vertes correspondent à +/- 3,14 rad.

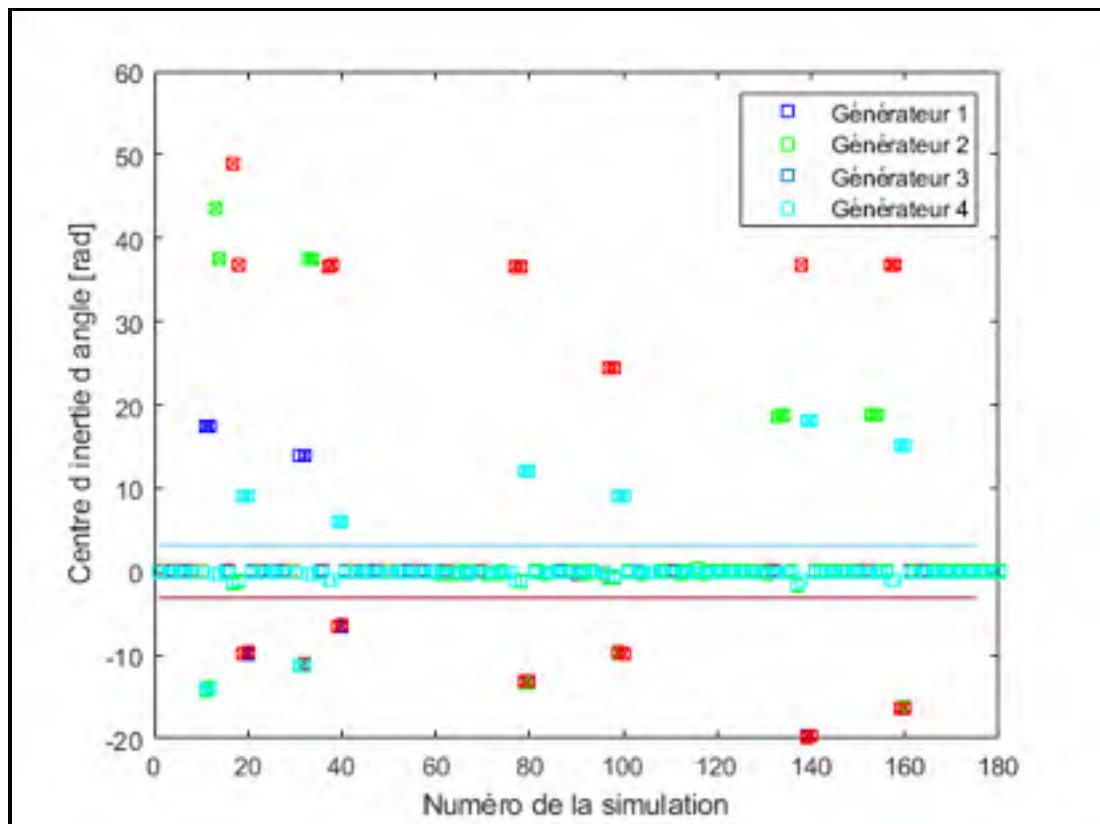


Figure 3.2 Centre d'inertie d'angle des quatre générateurs

Nous avons choisi d'utiliser le seuil minimum de vitesse de Burhan (2012) qui correspond à un seuil de vitesse angulaire de  $\pm 0.0002$  pu. Ce seuil est représenté par la barre bleue et verte à la figure suivante :

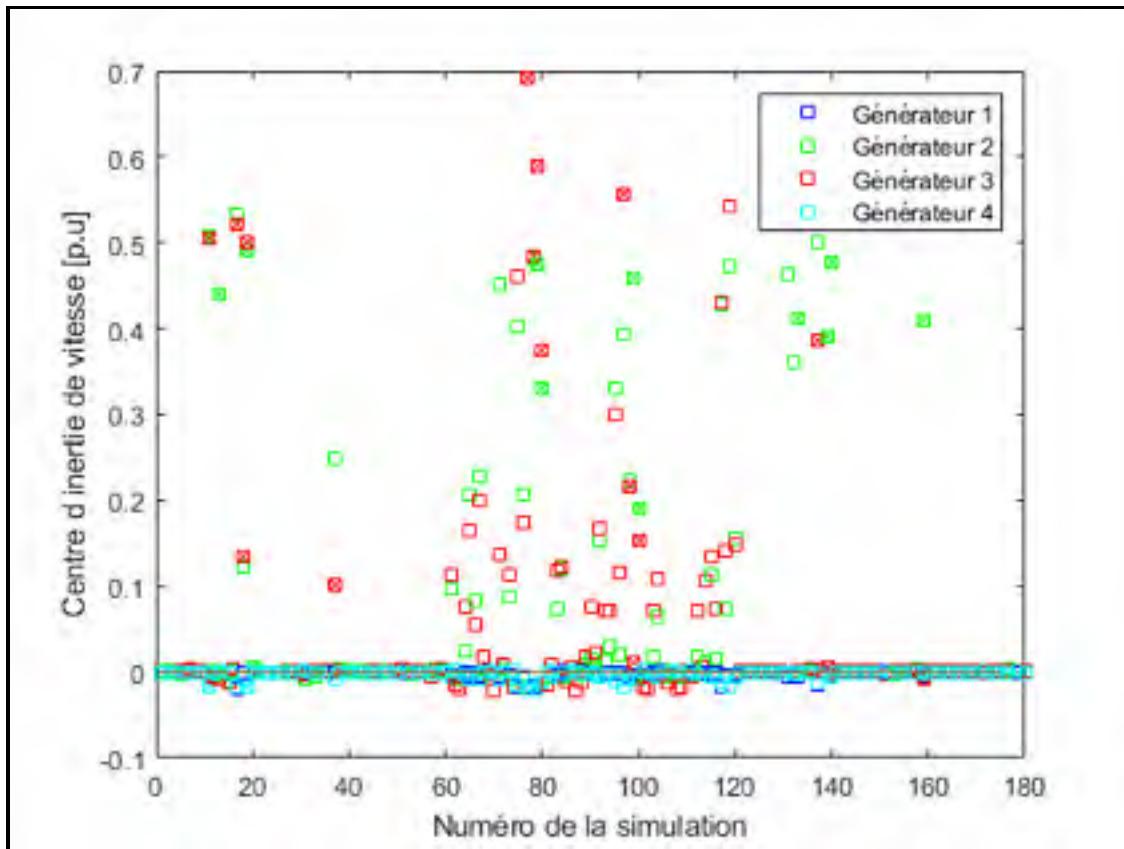


Figure 3.3 Centre d'inertie de vitesse des quatre générateurs

### 3.2.3 Application sur l'ensemble de données

Ayant donc déterminé le centre d'inertie de vitesse à  $\pm 0.0002$  pu et d'angle à  $\pm 3.14$  rad, nous appliquons ces deux conditions sur l'ensemble des simulations. C'est grâce à ces critères que le vecteur d'état sera défini pour permettre un apprentissage.

### 3.2.4 Vecteur d'état

Une fois que nous avons déterminé quelles sont les simulations stables et instables, nous pouvons construire un vecteur qui contient l'information sur la stabilité pour chaque simulation. Ce vecteur est un seul chiffre qui caractérise si la simulation est stable avec une valeur de 0 et instable si la valeur est 1. Pour chaque simulation, chaque générateur possède sa

propre valeur de stabilité. La détermination de la stabilité d'une simulation se fait avec les critères de stabilité du centre d'inertie d'angle et de vitesse définis à la section 3.2.2.

### 3.3 Crédation des différents ensembles

Afin d'entraîner un algorithme d'apprentissage, nous employons 75% de tout l'ensemble des simulations à l'ensemble d'apprentissage. Le 25 % restant des simulations va servir comme ensemble de test pour évaluer les performances de l'apprentissage de l'algorithme.

La formation de ces ensembles se fait aléatoirement. Une simulation donnée aura une probabilité de 75 % de faire partie de l'ensemble d'apprentissage et 25 % de faire partie de l'ensemble de test. Ceci permet d'avoir une répartition uniforme des cas de simulations stables et instables et d'avoir deux ensembles avec la même proportion. Lorsque les ensembles sont préparés, le nom des simulations est préservé pour permettre de les retracer une fois que les simulations seront partagées entre l'ensemble d'apprentissage et de test.

Le graphique ci-dessous montre la précision de la prédiction d'un ensemble de test en fonction du pourcentage de données attribué à l'ensemble d'entraînement. On peut constater que si l'ensemble d'entraînement est petit par rapport à l'ensemble de test, il y a sous-entraînement qui donne peu de performance de prédiction. À l'inverse, si l'ensemble d'entraînement occupe une grande partie de l'ensemble total de données, il y a un risque de surentraînement et une mauvaise capacité de généralisation sur l'ensemble de test ce qui résulte en des performances réduites.

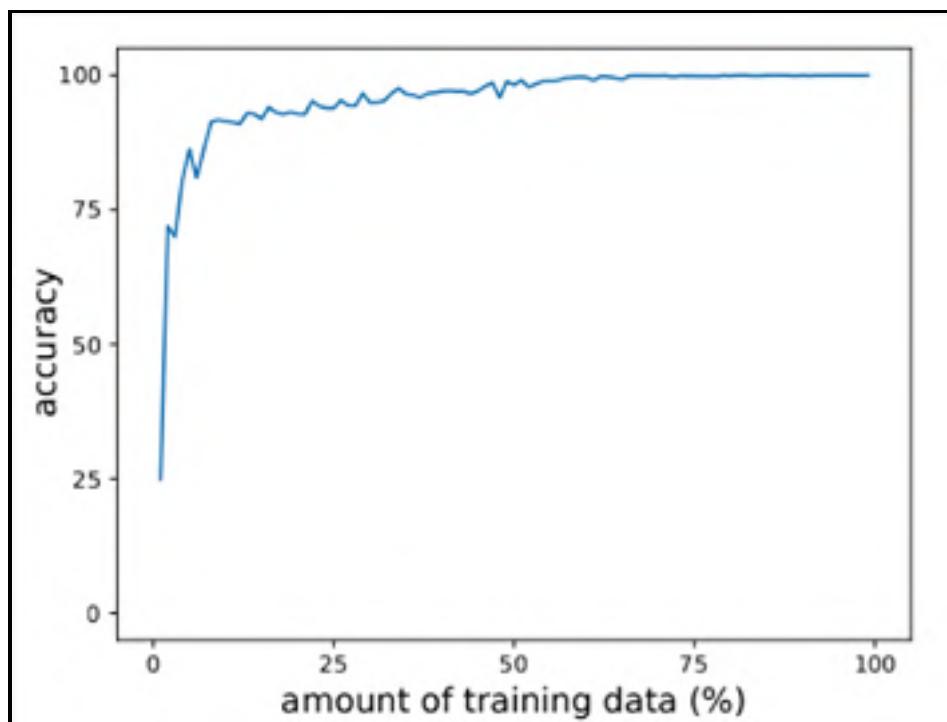


Figure 3.4 Précision de la prédiction en fonction de la taille du groupe d’entraînement

En résumé, dans notre étude, 75% des simulations sont dédiées à l’ensemble d’apprentissage et le 25% des simulations restantes sont dédiées à l’ensemble de test.

### 3.4 Exportation des données

Dans le but d’utiliser Python pour l’utilisation de l’algorithme d’apprentissage profond et la prédiction des résultats, nous devons réexporter les données en format .MAT.

Chaque fichier à exporter contient un ensemble d’apprentissage et de test dans lesquels ont été répartis aléatoirement les cas de simulations. Un deuxième fichier est associé à ce premier fichier. Il contient le nom des simulations pour chacun de ces ensembles afin de pouvoir garder la traçabilité des résultats. Ce deuxième fichier est ajouté en annexe.

Au préalable, nous avons retiré toutes les données jusqu'à l'apparition du court-circuit et nous n'avons gardé que les 200 millisecondes après la contingence, car la prédiction de l'état du système se fera après que le court-circuit soit arrêté.

À la fin du processus, il y a 4 fichiers .mat exportés de MATLAB. Un fichier contient l'information pour une seule machine incluant les tensions et courants triphasés, les puissances actives et réactives et il contient également son vecteur statut. Le fichier par contre ne contient pas l'information sur les angles et vitesses angulaires des machines pour éviter que l'algorithme apprenne de ces données directement.

## **CHAPITRE 4**

### **APPRENTISSAGE DES DONNÉES**

Tout d'abord dans ce chapitre, il sera question du choix du logiciel utilisé pour l'apprentissage machine avec des réseaux de neurones.

Par la suite nous allons établir le nombre d'époques optimal pour l'apprentissage machine en fonction du pouvoir de prédiction des modèles produits. Un résumé sera fait des raisons derrière le choix du nombre d'époques choisies.

Et finalement la dernière section traitera des résultats de l'apprentissage machine pour chacun des générateurs au nombre d'époques choisies.

#### **4.1       TensorFlow**

Il existe plusieurs choix pour développer des algorithmes de réseaux de neurones, mais de loin le plus connu est la librairie TensorFlow de Python qui offre une immense librairie pour modéliser un grand éventail de fonctions prédéterminées. MATLAB permet de faire des réseaux de neurones, mais les librairies disponibles sont beaucoup plus limitées sur MATLAB pour faire des réseaux de neurones. De plus, il existe toute une communauté derrière TensorFlow pour aider les débutants dans l'utilisation d'algorithmes de Deep Learning. C'est pour ces raisons que TensorFlow et Python ont été choisis pour effectuer les calculs des algorithmes des réseaux de neurones en Deep Learning.

#### **4.2       Détermination du nombre d'époques optimal**

Le nombre d'époques correspond au nombre de fois que l'algorithme d'apprentissage passe à travers l'ensemble d'entraînement. Il est important de bien calibrer ce nombre d'époques puisque trop peu d'époques d'entraînement et le réseau de neurones n'aura pas eu le temps de bien converger vers un modèle qui minimise le taux d'erreur. À l'opposé, trop d'époques et le

réseau de neurones est à risque de s'exposer au phénomène de surapprentissage. Le surapprentissage d'un réseau de neurones se réalise lorsque le modèle d'apprentissage est trop ciblé sur l'ensemble d'entraînement de sorte que le réseau perd sa capacité de généraliser son comportement sur d'autres ensembles. Le surapprentissage cause donc un plus grand taux d'erreurs sur un ensemble de test malgré d'excellents résultats sur l'ensemble d'apprentissage.

Pour trouver le bon nombre d'époques pour entraîner le réseau, nous avons choisi d'utiliser le générateur 3 comme baromètre puisqu'il est très sensible à des perturbations compte tenu de sa faible inertie. En utilisant ce générateur, nous avons réalisé une série d'entraînement avec différents nombres d'époques. Le nombre d'époques varie de 100 à 2000. Le barème pour mesurer le succès d'une détection est si la probabilité d'une détection de cas instable est supérieure à 50 %. Voici le résultat :

#### 4.2.1 100 époques.

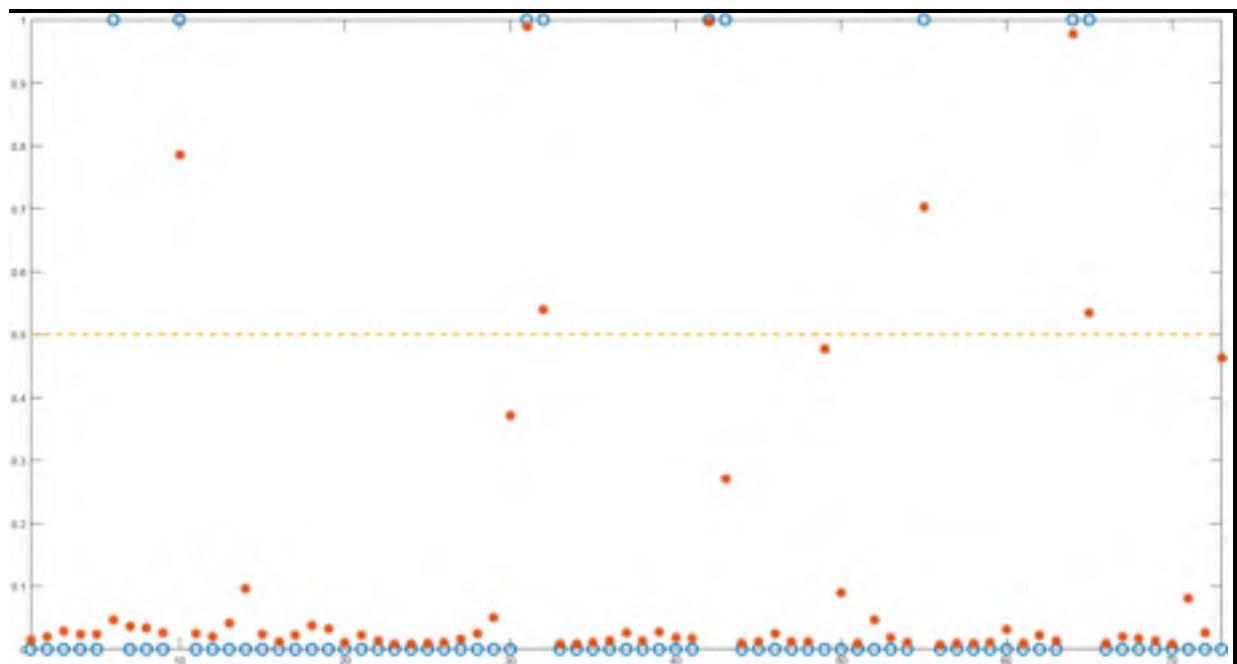


Figure 4.1 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 100 époques

Tableau 4.1 Matrice de confusion du générateur 3 avec 100 époques

		Cas prédit		
		0	1	Total prédit
Cas réel	0	171	0	171
	1	2	7	9
	Total réel	173	7	180

Tableau 4.2 Taux mesurés pour la matrice de confusion du générateur 3 avec 100 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
98.89%	100.00%	77.78%	1.11%	22.22%	0.00%

#### 4.2.2 200 époques

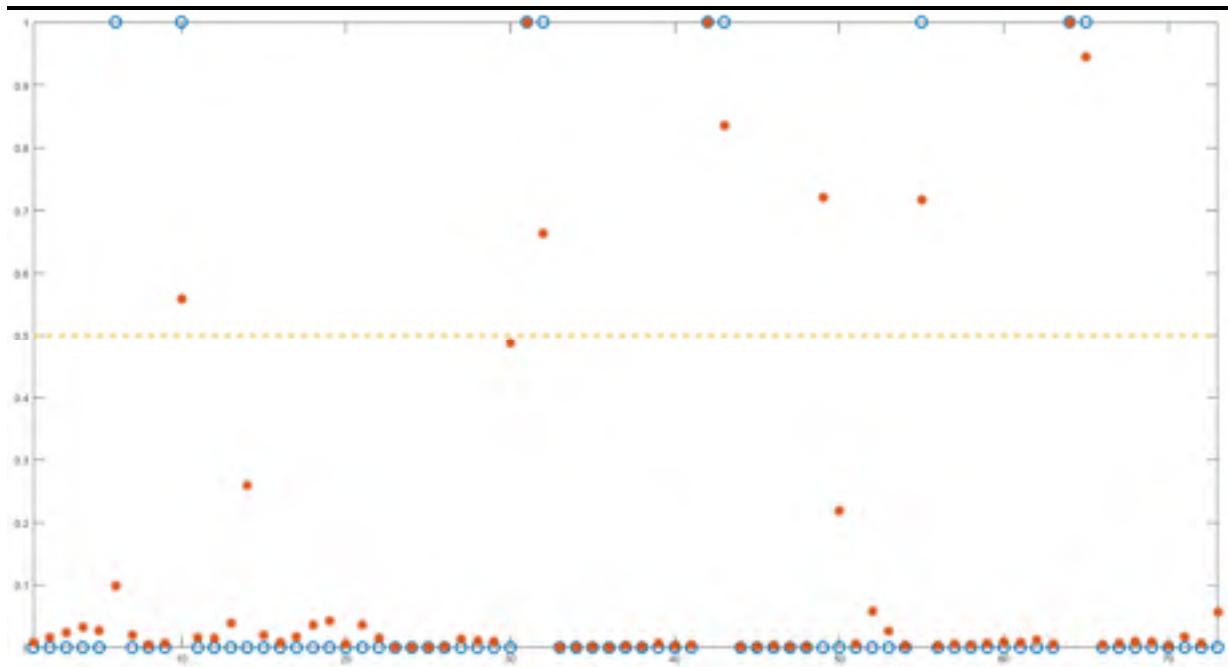


Figure 4.2 Prédiction (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 200 époques

Tableau 4.3 Matrice de confusion du générateur 3 avec 200 époques

		Cas prédict		
		0	1	Total prédict
Cas réel	0	171	1	172
	1	1	8	9
	Total réel	172	9	181

Tableau 4.4 Taux mesurés pour la matrice de confusion du générateur 3 avec 200 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
98.89%	99.42%	88.89%	1.11%	11.11%	0.58%

#### 4.2.3 300 époques

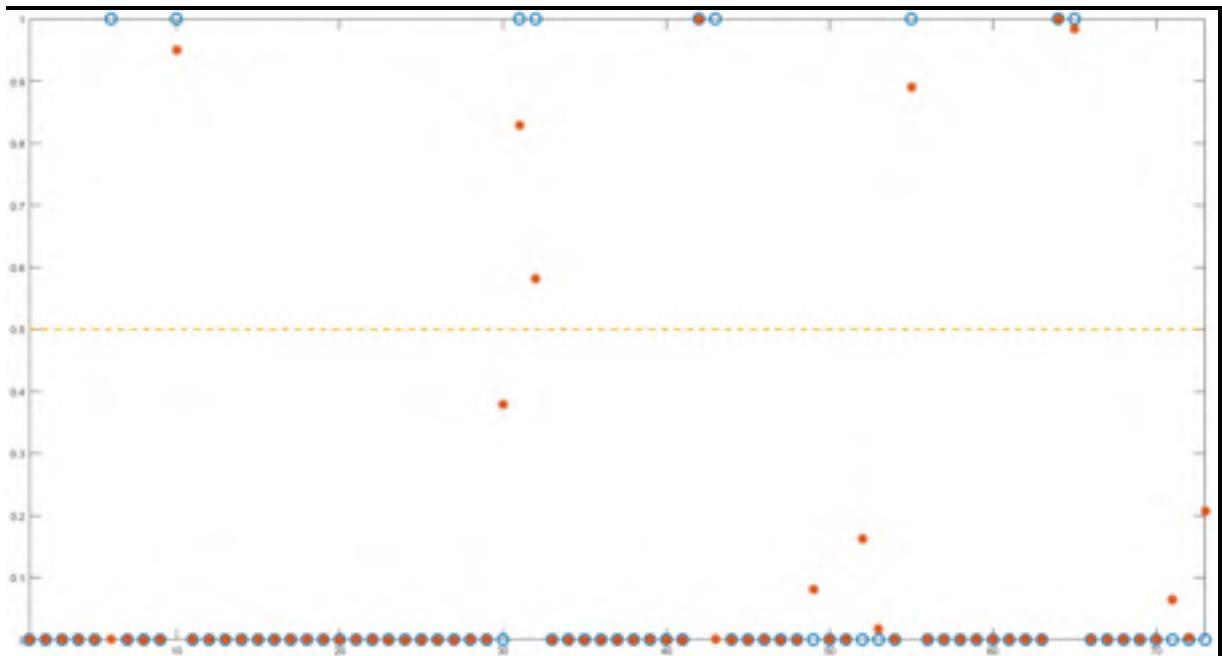


Figure 4.3 Prédiction (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 300 époques

Tableau 4.5 Matrice de confusion du générateur 3 avec 300 époques

		Cas prédit		
		0	1	Total prédit
Cas réel	0	171	0	171
	1	2	7	9
	Total réel	173	7	180

Tableau 4.6 Taux mesurés pour la matrice de confusion du générateur 3 avec 300 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif

98.89%	100.00%	77.78%	1.11%	22.22%	0.00%
--------	---------	--------	-------	--------	-------

#### 4.2.4 400 époques

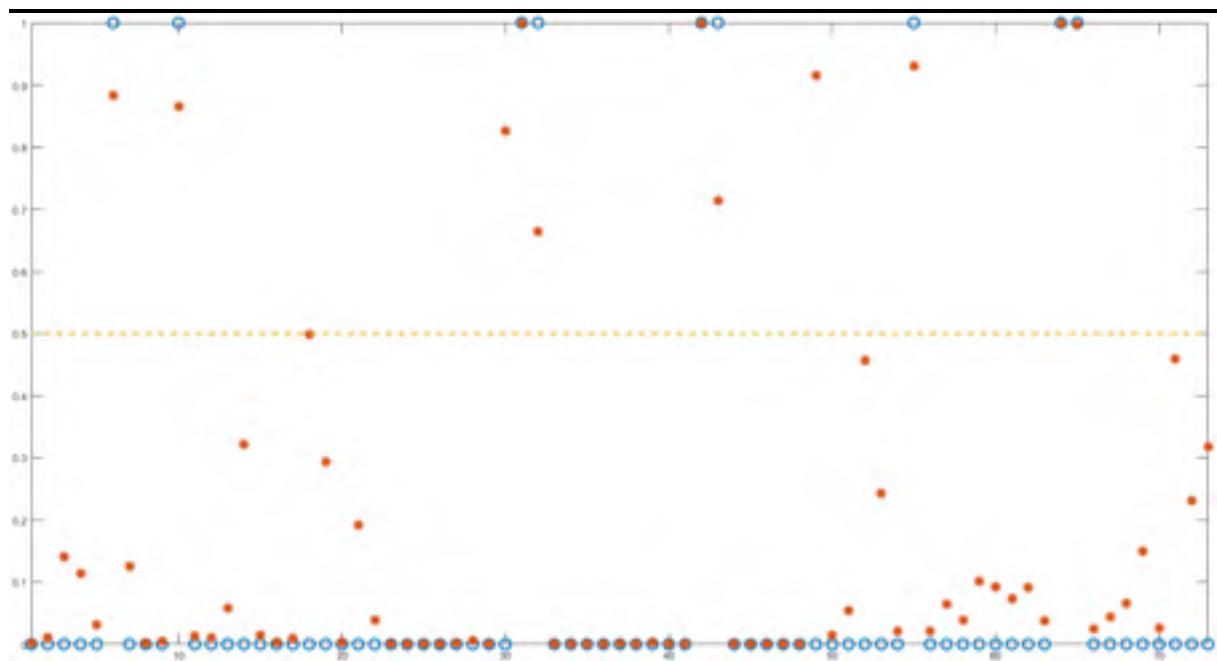


Figure 4.4 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 400 époques

Tableau 4.7 Matrice de confusion du générateur 3 avec 400 époques

		Cas prédit		Total prédit
		0	1	
Cas réel	0	169	2	171
	1	0	9	9
	Total réel	169	11	180

Tableau 4.8 Taux mesurés pour la matrice de confusion du générateur 3 avec 400 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
-----------	-------------	-------------	---------------	----------------------	----------------------

98.89%	98.83%	100.00%	1.11%	0.00%	1.17%
<b>4.2.5 500 époques</b>					

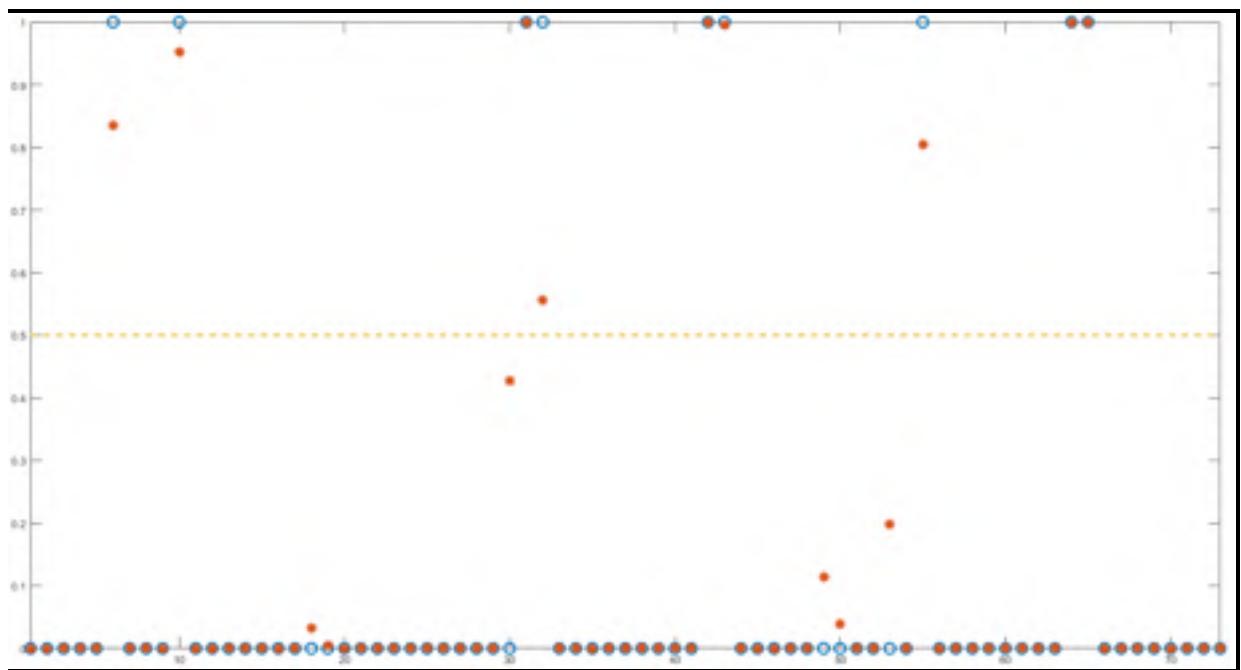


Figure 4.5 Prédiction (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 500 époques

Tableau 4.9 Matrice de confusion du générateur 3 avec 500 époques

		Cas prédict		Total prédict
		0	1	
Cas réelle	0	171	0	171
	1	0	9	9
	Total réel	171	9	180

Tableau 4.10 Taux mesurés pour la matrice de confusion du générateur 3 avec 500 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif

100.00%	100.00%	100.00%	0.00%	0.00%	0.00%
---------	---------	---------	-------	-------	-------

#### 4.2.6 600 époques

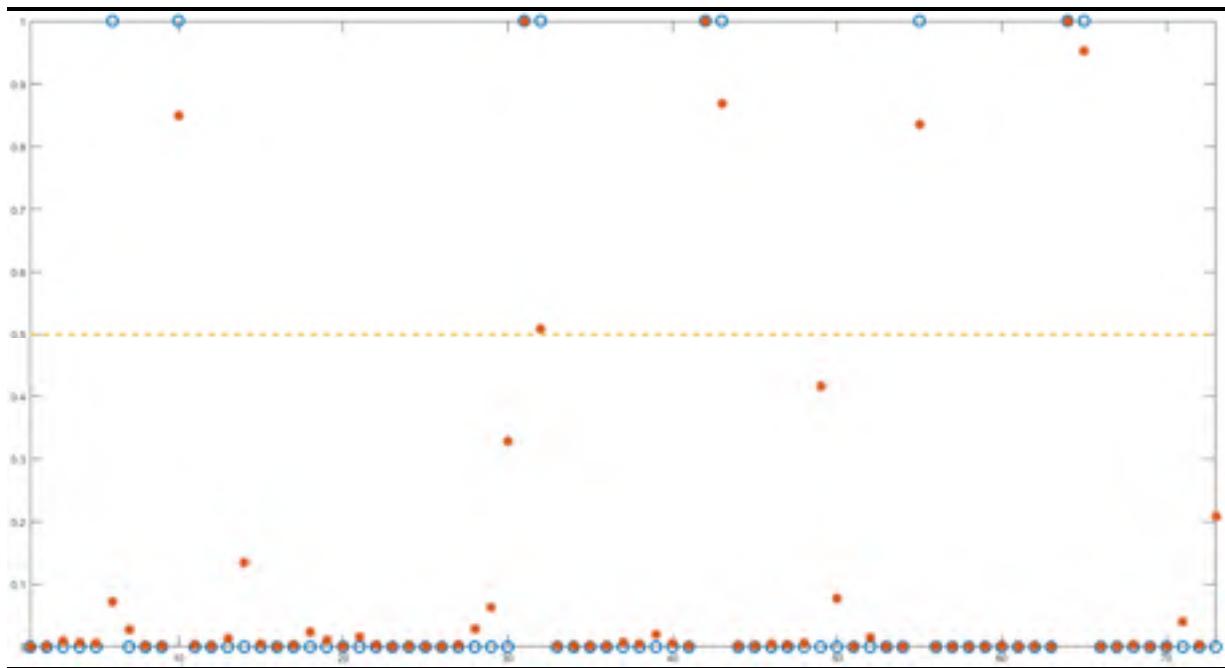


Figure 4.6 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 600 époques

Tableau 4.11 Matrice de confusion du générateur 3 avec 600 époques

		Cas prédit		
		0	1	Total prédict
Cas réelle	0	171	0	171
	1	1	8	9
	Total réel	172	8	180

Tableau 4.12 Taux mesurés pour la matrice de confusion du générateur 3 avec 600 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
-----------	-------------	-------------	---------------	----------------------	----------------------

99.44%	100.00%	88.89%	0.56%	11.11%	0.00%
--------	---------	--------	-------	--------	-------

#### 4.2.7 700 époques

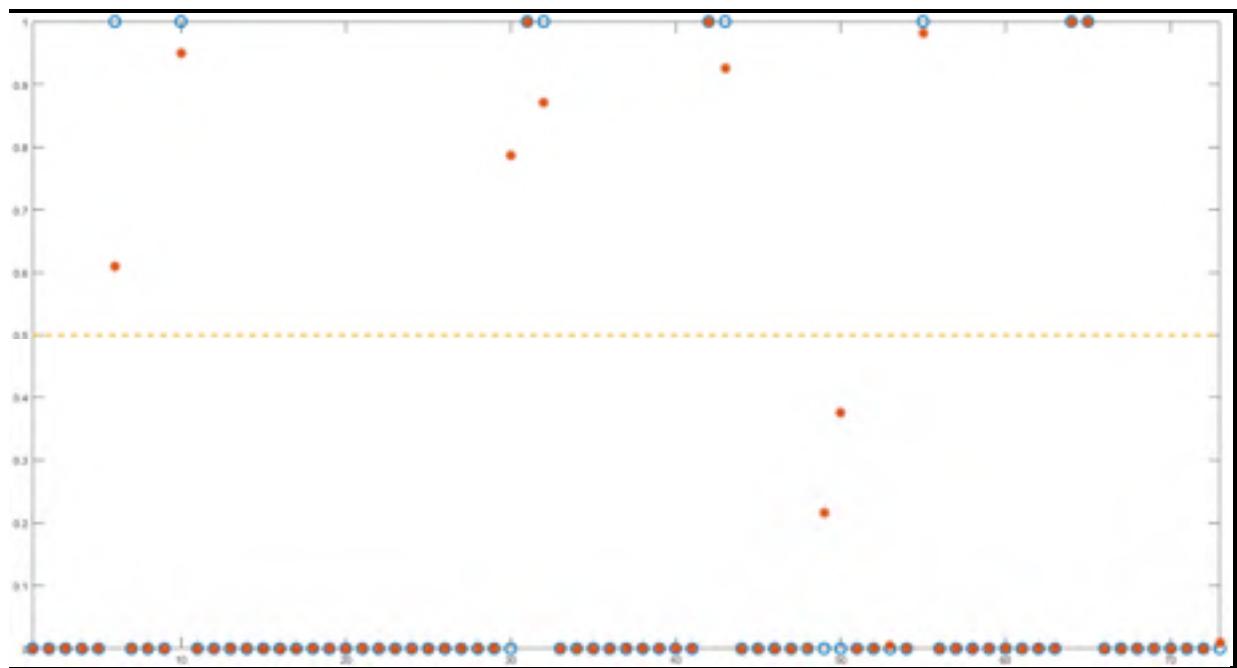


Figure 4.7 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 700 époques

Tableau 4.13 Matrice de confusion du générateur 3 avec 700 époques

		Cas prédit			Total prédit
		0	1		
Cas réel	0	170	1	171	
	1	0	9	9	
	Total réel	170	10	180	

Tableau 4.14 Taux mesurés pour la matrice de confusion du générateur 3 avec 700 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
99.44%	99.42%	100.00%	0.56%	0.00%	0.58%

#### 4.2.8 800 époques

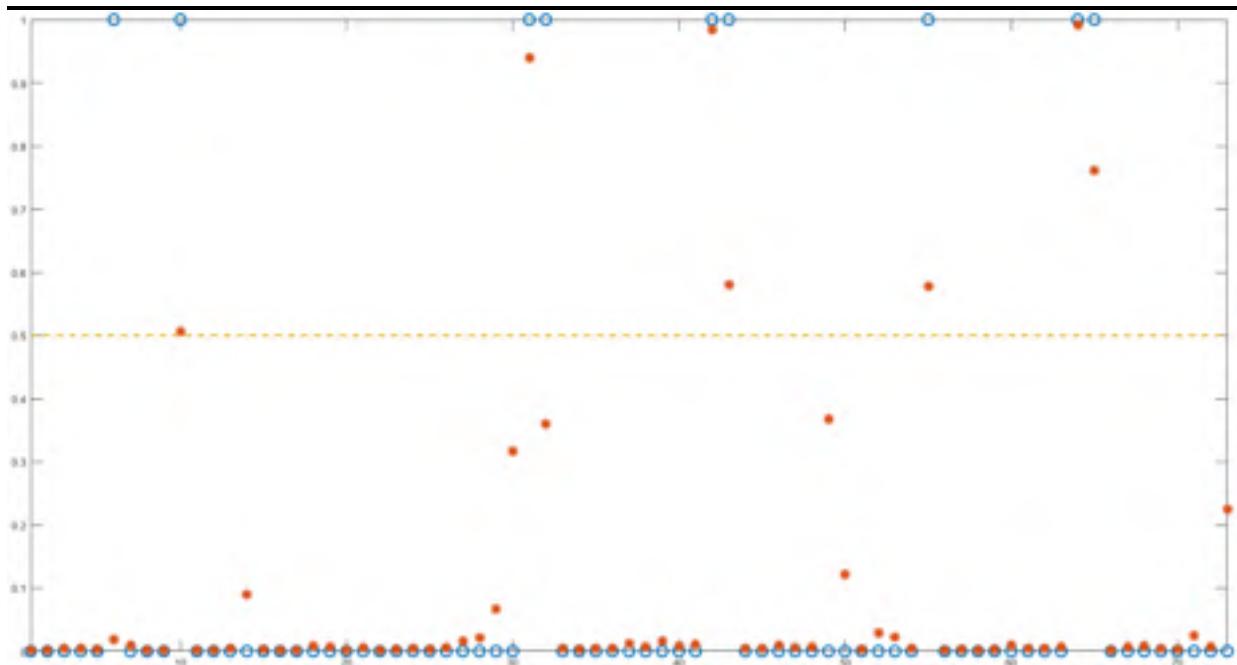


Figure 4.8 Prédiction (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 800 époques

Tableau 4.15 Matrice de confusion du générateur 3 avec 800 époques

		Cas prédict		
		0	1	Total prédict
Cas réel	0	171	0	171
	1	2	7	9
	Total réel	173	7	180

Tableau 4.16 Taux mesurés pour la matrice de confusion du générateur 3 avec 800 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
98.89%	100.00%	77.78%	1.11%	22.22%	0.00%

#### 4.2.9 900 époques

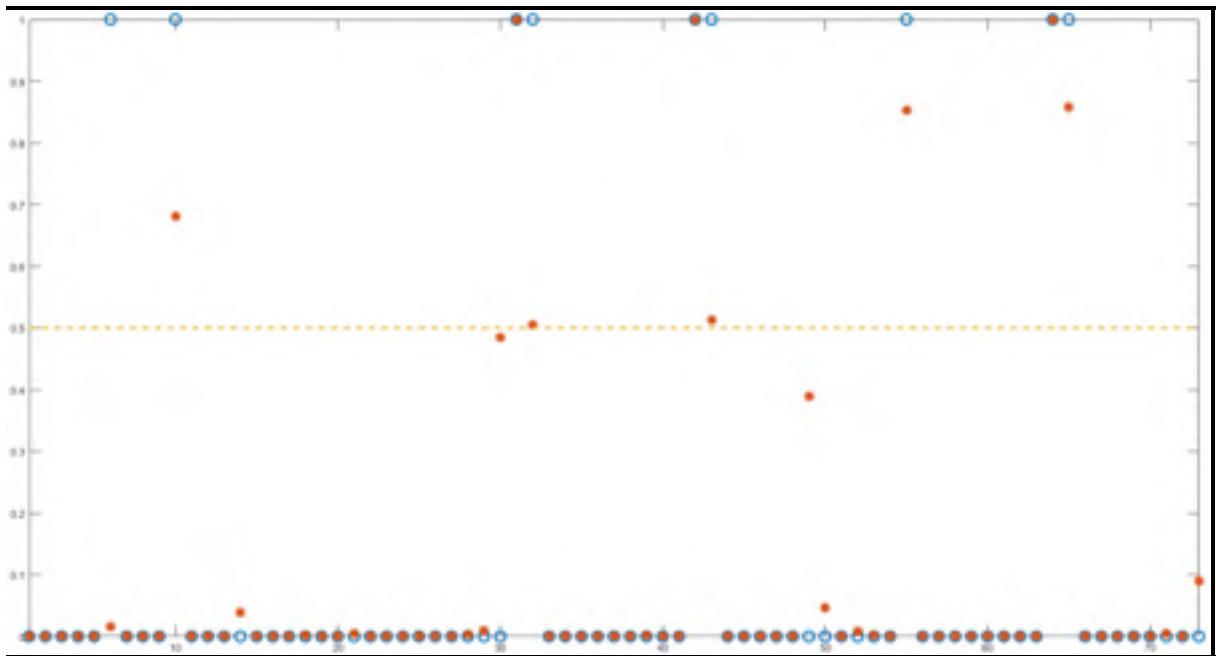


Figure 4.9 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 900 époques

Tableau 4.17 Matrice de confusion du générateur 3 avec 900 époques

		Cas prédit		
		0	1	Total prédit
Cas réel	0	171	0	171
	1	1	8	9
	Total réel	172	8	180

Tableau 4.18 Taux mesurés pour la matrice de confusion du générateur 3 avec 900 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
99.44%	100.00%	88.89%	0.56%	11.11%	0.00%

#### 4.2.10 1000 époques

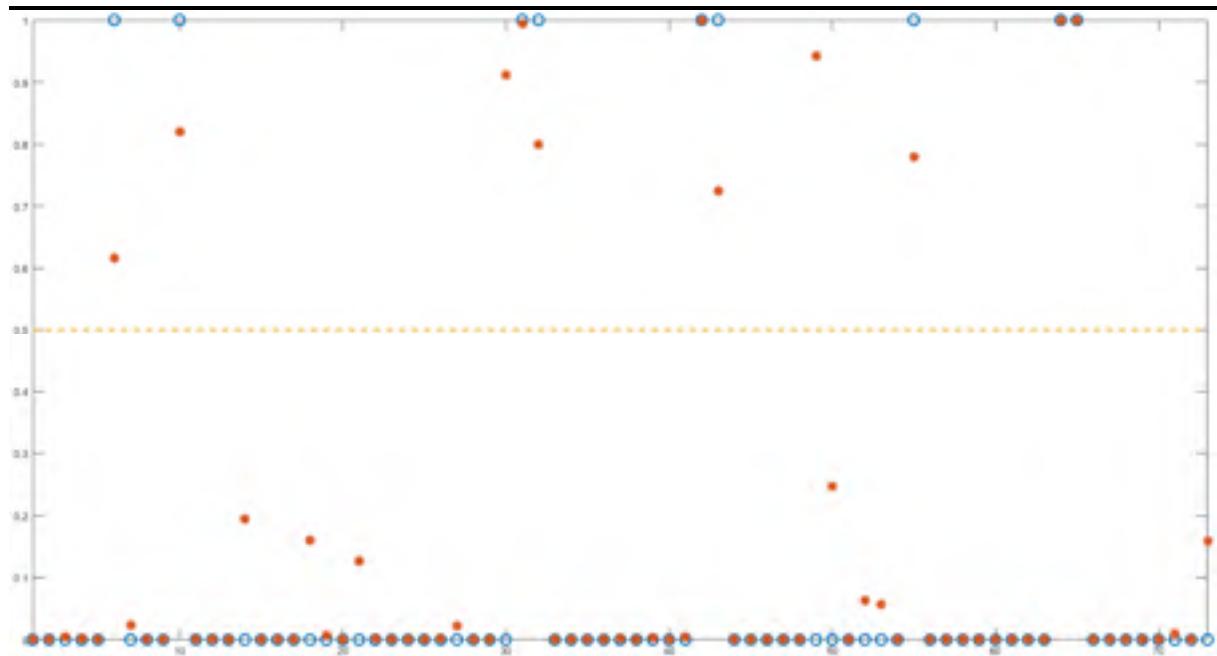


Figure 4.10 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 1000 époques

Tableau 4.19 Matrice de confusion du générateur 3 avec 1000 époques

		Cas prédit		
		0	1	Total prédit
Cas réel	0	169	2	171
	1	0	9	9
	Total réel	169	11	180

Tableau 4.20 Taux mesurés pour la matrice de confusion du générateur 3 avec 1000 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
98.89%	98.83%	100.00%	1.11%	0.00%	1.17%

#### 4.2.11 2000 époques

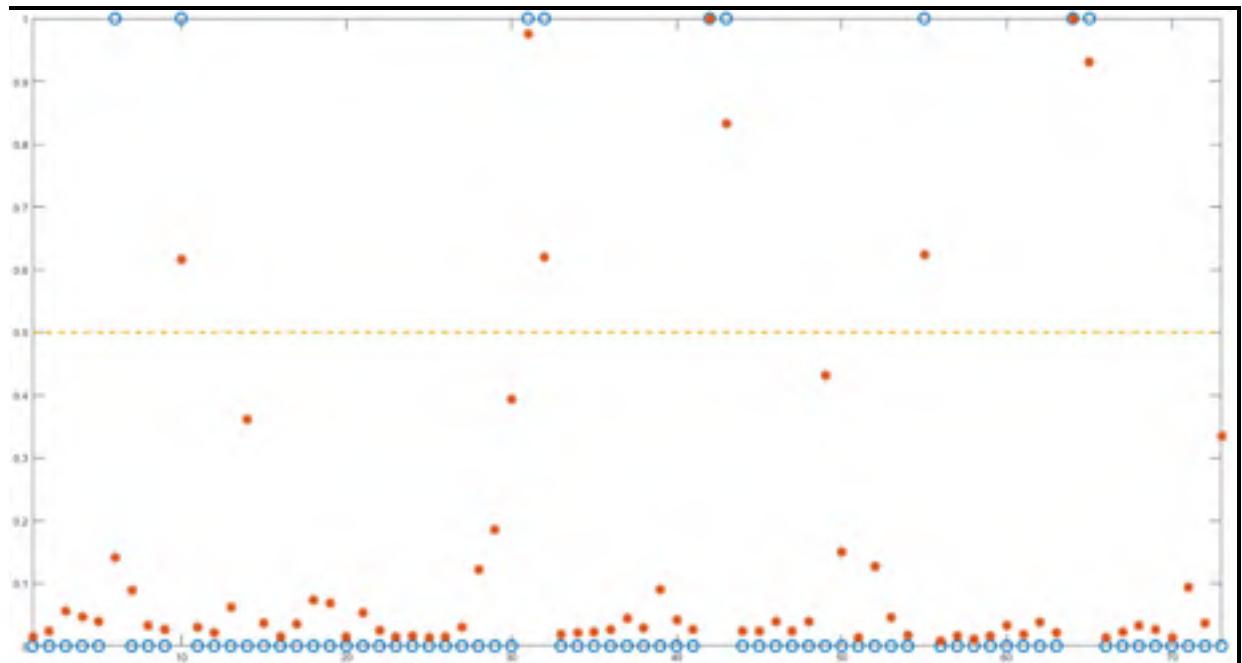


Figure 4.11 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations, cas réels en bleu pour 2000 époques

Tableau 4.21 Matrice de confusion du générateur 3 avec 2000 époques

		Cas prédit		
		0	1	Total prédict
Cas réel	0	171	0	171
	1	1	8	9
	Total réel	172	8	180

Tableau 4.22 Taux mesurés pour la matrice de confusion du générateur 3 avec 2000 époques

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif

99.44%	100.00%	88.89%	0.56%	11.11%	0.00%
--------	---------	--------	-------	--------	-------

#### 4.2.12 Résumé et analyse

Nous pouvons constater que le nombre d'époques n'affecte pas de façon significative la précision des résultats. Les prédictions semblent essentiellement similaires.

Ceci dit, nous constatons une augmentation très importante de la certitude des prédictions. La certitude des prédictions se mesure par le nombre de prédictions proche du 100 % de probabilité d'instabilité dans une simulation et par le nombre de prédictions proche du 0 % pour le cas des simulations stables.

En effet, on constate une augmentation de la certitude des prédictions de 100 époques à 400 époques. Cette augmentation semble s'arrêter à un maximum à 500 époques. Puis elle se stabilise jusqu'à 1000 époques. Par la suite on voit qu'à 2000 époques la certitude des prédictions diminue drastiquement, ce qui indique que le surentraînement est bien amorcé.

On remarque également que l'entraînement à 500 époques produit des résultats exempts de faux positif ou faux négatif. Ce qui semble suggérer que nous avons atteint un optimum sur la performance d'apprentissage du réseau de neurones MALSTM-FCN pour cet ensemble de données.

Nous pouvons constater que l'augmentation du nombre d'époques sur l'ensemble d'apprentissage réduit de façon importante le nombre de faux positif et augmente la certitude des prédictions. Nous voyons également qu'après un certain nombre d'époques, la performance décline substantiellement due au phénomène de surapprentissage des réseaux de neurones qui perdent leur capacité à généraliser sur d'autres ensembles ce qu'ils ont appris dans l'ensemble d'apprentissage. Nous voyons également que les performances entre 500 et 1000 époques sont sensiblement pareilles, mais le temps d'entraînement passe de 1 h. à 2 h. pour peu de gains. Nous pouvons prendre la décision suivante concernant les autres générateurs à entraîner :

- une période d’entraînement de 500 époques est largement suffisante et permet d’obtenir les meilleurs résultats avec le moins grand nombre d’époques;
- cette période d’entraînement de 500 époques nous permet également d’éviter le surentraînement qui survient lorsque le nombre d’époques dépasse 1000 époques.

### 4.3 Phase de test du réseau de neurones

Une fois le nombre d’époques optimal déterminé, nous pouvons enfin réaliser la validation des ensembles de tests pour les quatre générateurs. Chaque entraînement des générateurs prend environ de une demi-heure à une heure et les ensembles d’entraînement sont presque toujours appris avec un taux de précision de 100 %.

Après avoir entraîné le réseau de neurones, nous utilisons l’ensemble de test afin de voir la qualité de la prédiction sur les quatre générateurs. Les étapes pour parvenir à ces résultats sont les suivantes : nous exportons les valeurs attendues de l’état de stabilité et les valeurs prédites et nous les comparons graphiquement avec le logiciel MATLAB.

Pour tous les graphiques, la valeur de l’axis correspond au no de simulation de 1 à 75 pour l’ensemble de test qui représente environ 40 % des cas de simulations. La valeur de l’ordonnée donne la prédiction qui est une valeur entre 0 et 1. Plus la prédiction est proche de 1, plus l’algorithme est certain de la stabilité de cette simulation donnée. Les valeurs attendues sont également illustrées. Elles prennent soit la valeur 0 lorsque la simulation est stable ou 1 si la simulation est instable. Au milieu du graphique à 0.5 il y a une ligne rouge qui représente une certitude de 50 %. Nous considérons qu’au-dessus de ce 0.5 l’algorithme détermine que la simulation est instable. Voici un résumé des cas stables et instables des quatre générateurs :

Tableau 4.23 Ensemble des simulations stables et instables

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
<b>Stables</b>	166	156	152	164
<b>Instables</b>	7.78 %	13.34 %	15.56 %	9.76 %

La section suivante présente les résultats pour l’ensemble des quatre générateurs.

### 4.3.1 Générateur 1

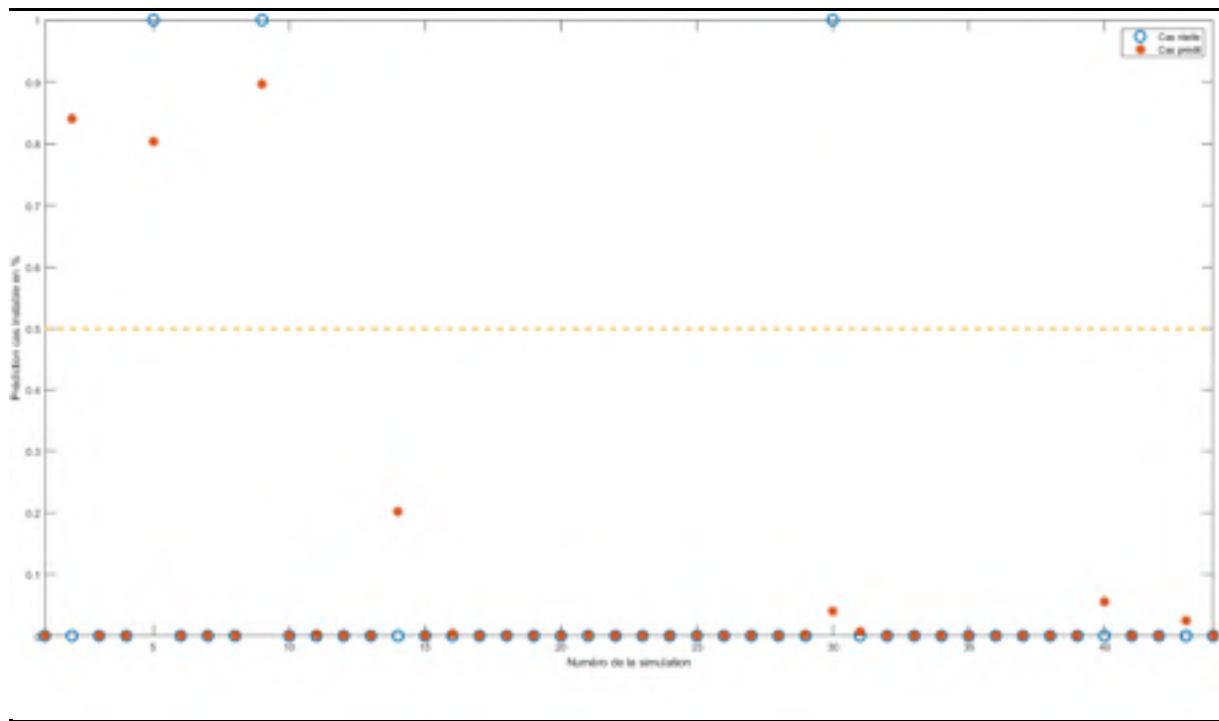


Figure 4.12 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 1

Tableau 4.24 Matrice de confusion du générateur 1

		Cas prédit		
		0	1	Total prédit
Cas réel	0	176	1	177
	1	1	2	3
	Total réel	177	3	180

Tableau 4.25 Taux mesurés pour la matrice de confusion du générateur 1

<b>Précision</b>	<b>Sensibilité</b>	<b>Spécificité</b>	<b>Taux d'erreur</b>	<b>Taux de faux positif</b>	<b>Taux de faux négatif</b>
98.89%	99.44%	66.67%	1.11%	33.33%	0.56%

Voici les résultats pour le premier générateur. On peut voir que le réseau de neurones est bien capable de discerner les cas stables des cas instables. Pour le premier générateur, l'ensemble des simulations non apprises ont correctement été prédites avec une probabilité de 80.32 % de l'instabilité à la simulation no 5 ainsi que la simulation no 9 à 89.64 %. L'ensemble présente un faux positif pour la simulation 2 à 84.00 % et un faux négatif pour la simulation 30 à 4.02 % d'instabilité.

À la figure suivante, nous pouvons mieux analyser le cas du faux positif en rouge de la simulation no 2 en comparaison avec un cas stable en bleu de la simulation no 25.

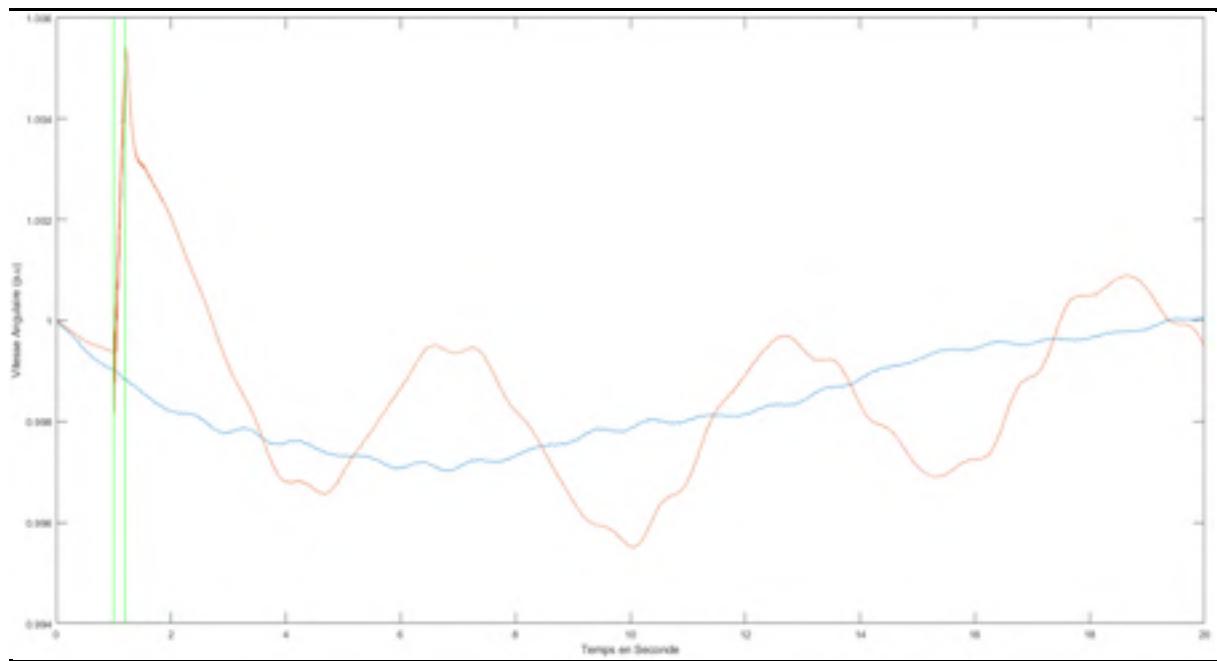


Figure 4.13 Vitesse Angulaire (pu) de la simulation faux positif no 2 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

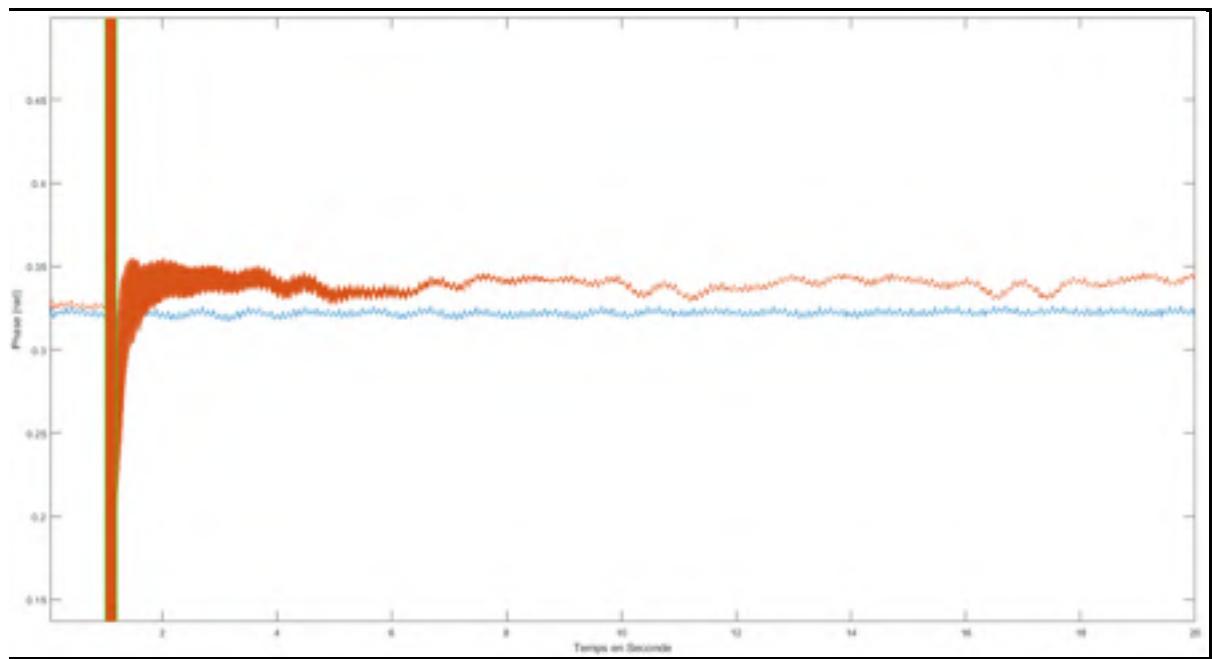


Figure 4.14 Phase (radian) de la simulation faux positif no 2 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

On peut constater qu'il y a une plus grande oscillation de la vitesse angulaire et de la phase mais que dans les deux cas la machine no 1 se stabilise. En comparaison voici un cas classé instable bien prédit par l'algorithme. A la figure suivante, voici la simulation no 9 en rouge comparé avec la simulation no 25 en bleu.

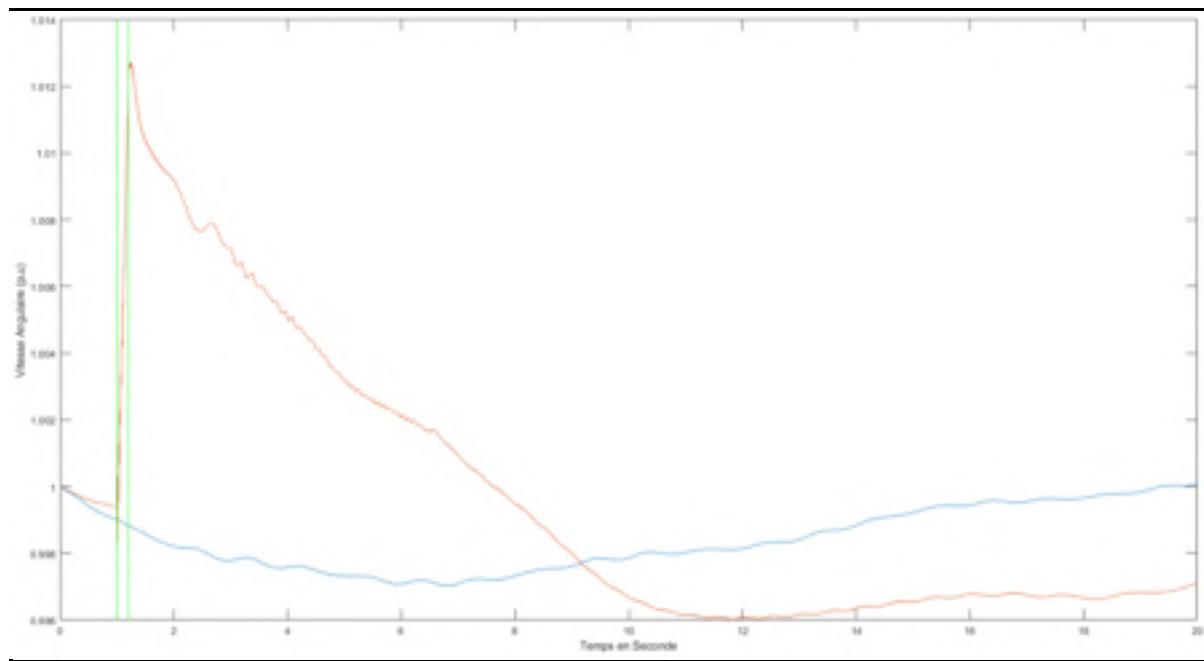


Figure 4.15 Vitesse Angulaire (pu) de la simulation instable no 9 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert

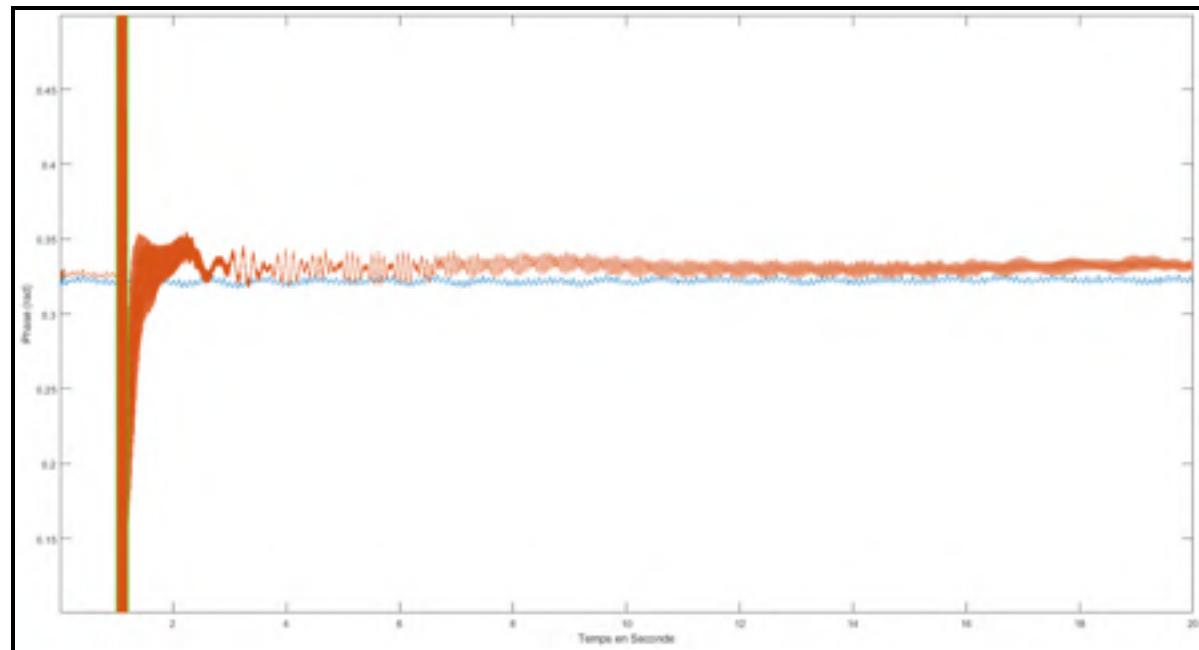


Figure 4.16 Phase (radian) de la simulation instable no 9 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert

Nous pouvons constater que la machine no 1 semble réagir de la même façon que dans le cas no 2 mais avec une plus grande amplitude. Comme les deux comportements sont très similaires, il est possible que l'algorithme ait associé ce genre de comportement à un classement instable et ceci pourrait expliquer le faux positif.

Il y a aussi un cas de faux négatif analysé à la figure suivante. Dans les deux prochaines figures, on peut voir la comparaison entre la vitesse angulaire et la phase de la simulation du cas no 30 en rouge et 25 en bleu (le cas stable témoin).

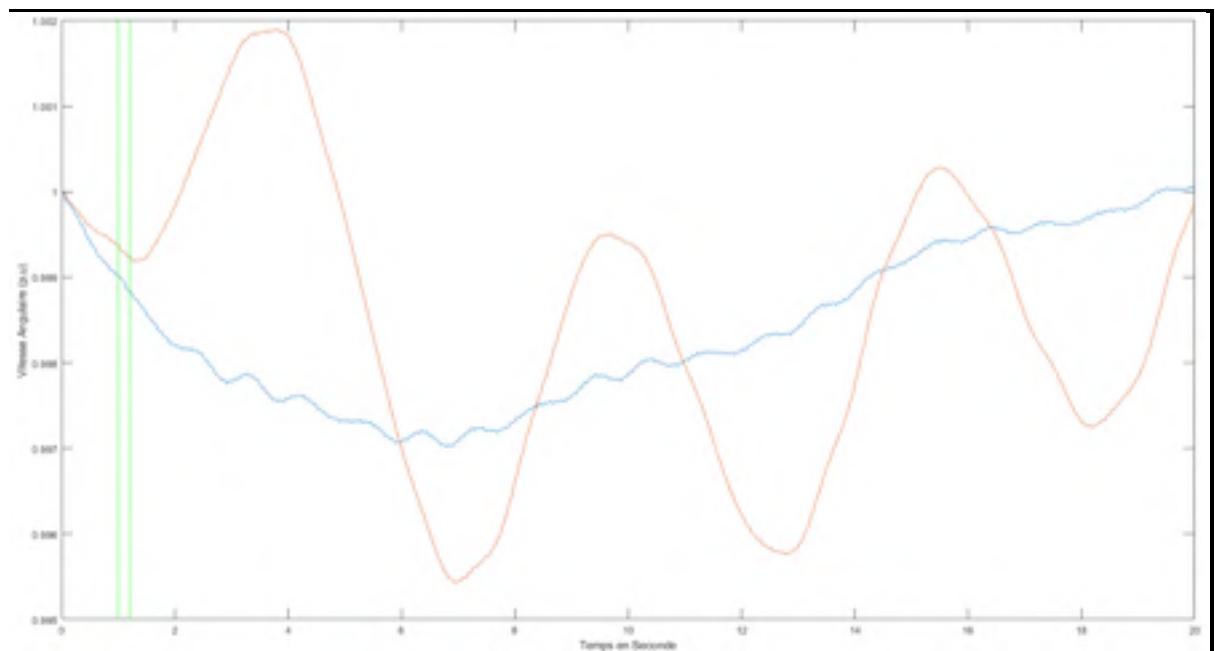


Figure 4.17 Vitesse Angulaire (pu) de la simulation faux négatif no 30 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

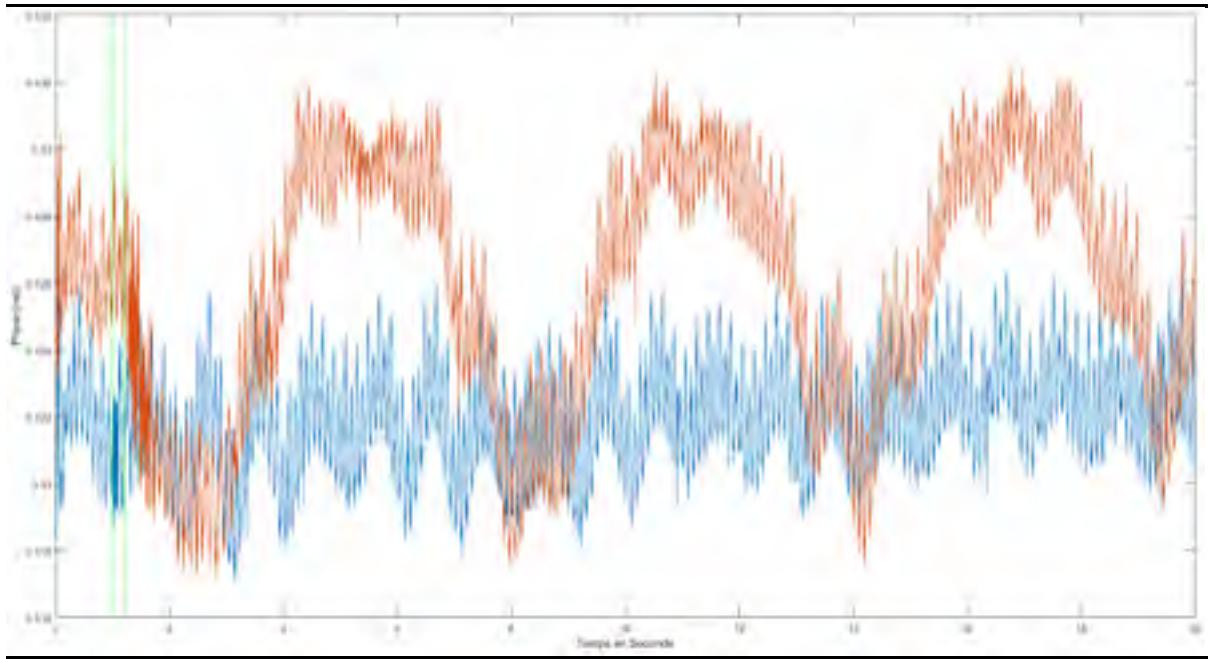


Figure 4.18 Phase (radian) de la simulation faux négatif no 30 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

On constate que la simulation no 30 en rouge a une forte amplitude de variation de vitesse angulaire et de phase ce qui a été considéré comme une instabilité selon l'apprentissage de l'algorithme mais pas selon les critères de classement de stabilité.

Au total, ceci donne un taux de spécificité de 66,67 %. Ce qui correspond à un apprentissage qui détecte 3/4 des cas. Le générateur 1 possède une grande inertie et donc il est plus difficile d'obtenir beaucoup de cas d'instabilité dans son ensemble d'entraînement. Ce qui se reflète en peu de cas d'instabilité dans l'ensemble de test, soit seulement 4. Ce qui n'est peut-être pas significatif statistiquement.

Pour les autres cas de simulation, nous voyons que les simulations sont détectées dès les premiers individus et les erreurs pendant l'instabilité sont faibles comme nous le montre la matrice de confusion établie avec l'ensemble de validation.

Nous voyons que le taux d'erreur est faible, mais le taux de faux positif (1 classé comme 0 et 0 classé comme 1) est grand.

Les fichiers de simulation correspondant aux numéros de simulation ainsi que les résultats des prédictions sont fournis en annexe.

#### 4.3.2 Générateur 2

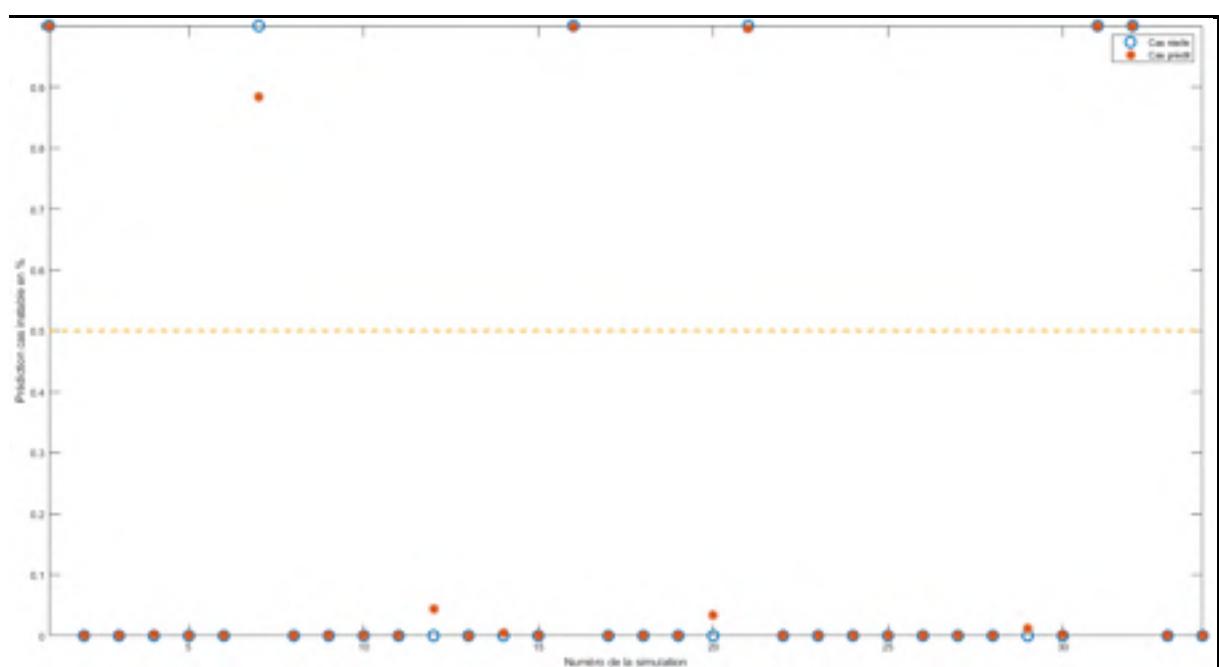


Figure 4.19 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 2

Tableau 4.26 Matrice de confusion du générateur 2

		Cas prédit		
		0	1	Total prédit
Cas réel	0	174	0	174
	1	0	6	6
	Total réel	174	6	180

Tableau 4.27 Taux mesurés pour la matrice de confusion du générateur 2

<b>Précision</b>	<b>Sensibilité</b>	<b>Spécificité</b>	<b>Taux d'erreur</b>	<b>Taux de faux positif</b>	<b>Taux de faux négatif</b>
100.00%	100.00%	100.00%	0.00%	0.00%	0.00%

La prédiction du générateur 2 semble différente. On constate que les simulations 7, 12, 20, 21 et 29 sont des cas de vrai positif et vrai négatif avec une probabilité d'instabilité de 88.28 %, 4.42 %, 3.43 %, 99.60 % et 1,20 % respectivement. Ces simulations sont celles avec la valeur d'incertitude la plus élevée de l'échantillon de test. Il n'y a pas de cas de faux positif ou de cas de faux négatif.

En comparaison avec le prédicteur du générateur 1, nous voyons qu'il y a une plus grande certitude sur les prédictions.

Nous voyons que nous avons un taux de vrai positif plus important que pour le générateur 1, ce qui explique que le prédicteur est sûr lors des détections d'instabilité. Le taux de faux négatif et de faux positif s'est réduit à 0.0%, ce qui est souhaitable. Il est préférable d'avoir plus de faux positif que de faux négatif très bas. Dans cet ensemble de test, le nombre de simulations instables est bien plus important que le générateur 1. Ce qui donne une performance plus statistiquement significative que le générateur précédent.

Dans l'ensemble, le réseau de neurones a très bien appris dans ce cas à distinguer les cas stables et il semble avoir eu moins de difficulté à généraliser les cas instables de son ensemble d'entraînement. Il est possible que ce résultat soit dû à un ensemble d'entraînement statistiquement représentatif de l'ensemble de test et que la majorité des ensembles donneraient des résultats semblables. Pour en avoir le cœur net, il faudrait refaire l'entraînement de ce générateur avec un autre ensemble d'entraînement et de test.

Voici à la figure ci-dessous la comparaison entre les vitesses angulaires et les phases de deux simulations de la machine no 3. La simulation no 21 en rouge qui est prédicté instable et la simulation no 25 en bleu qui est prédicté stable.

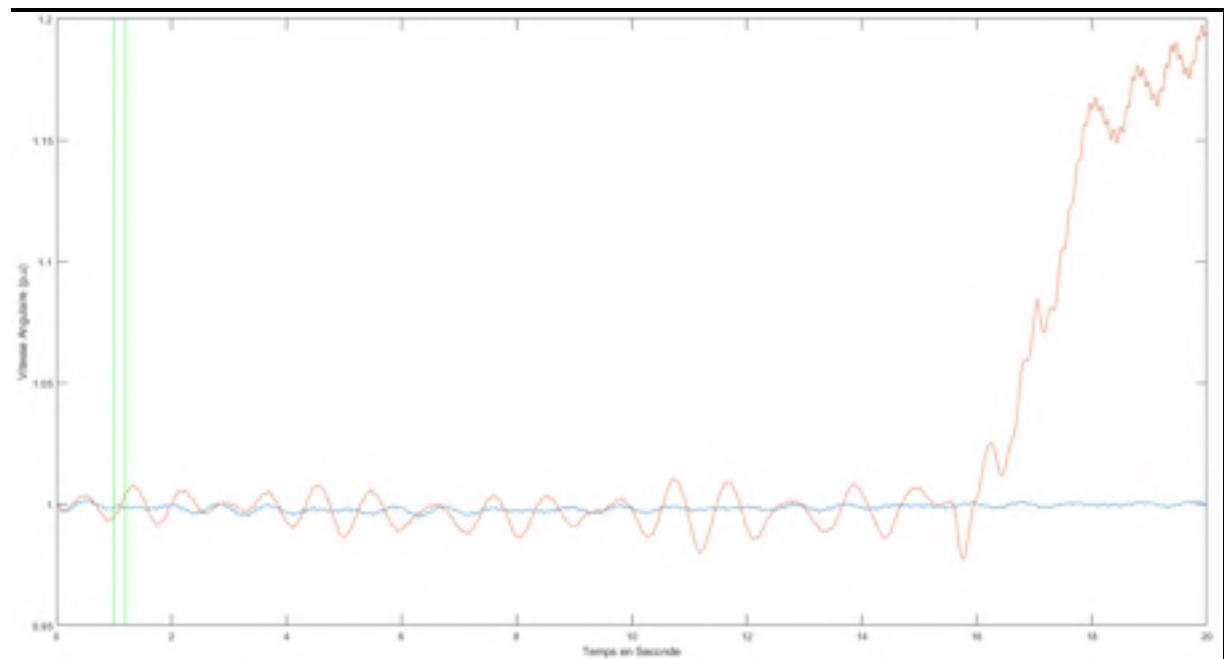


Figure 4.20 Vitesse Angulaire (pu) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

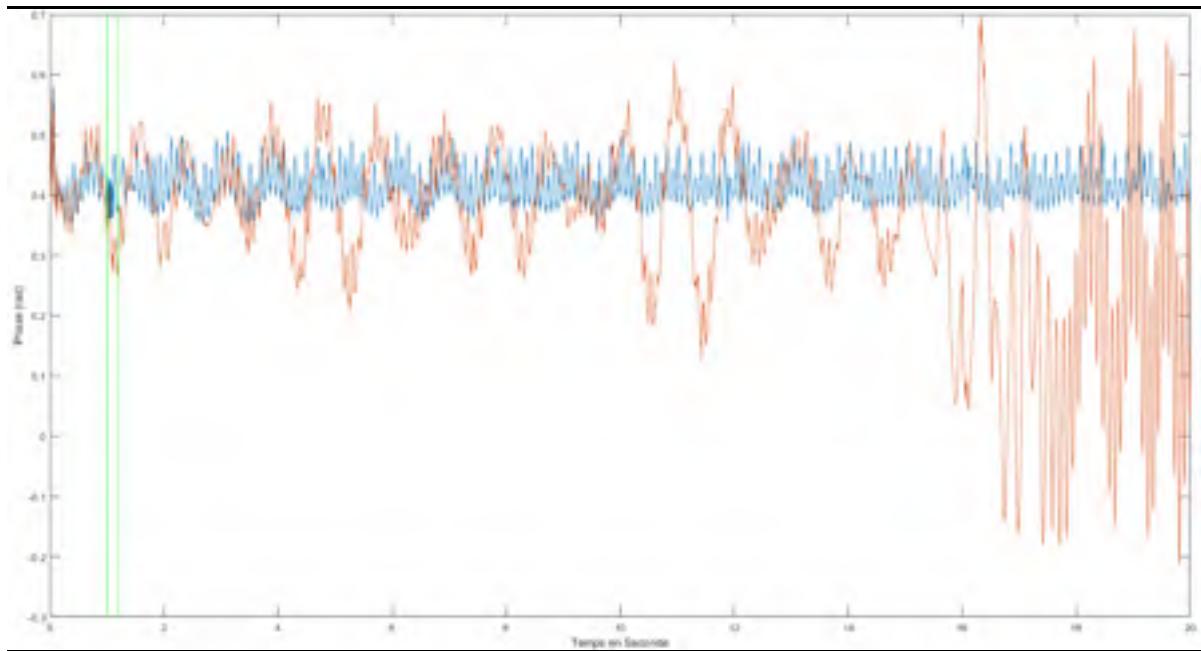


Figure 4.21 Phase (radian) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu. Portion de la courbe utilisée pour la prédition encadrée en vert

On constate que l'algorithme a su prédire l'instabilité du système au temps de 1.2 secondes alors que le décrochage survient vers 16 secondes, donc environ 15 secondes plus tard.

A la figure ci-dessous, voici un des intrants de l'algorithme de prédition. Il s'agit de la tension sur la première phase pour les deux simulations. C'est avec ce genre d'information que l'algorithme fait son analyse de la stabilité du système.

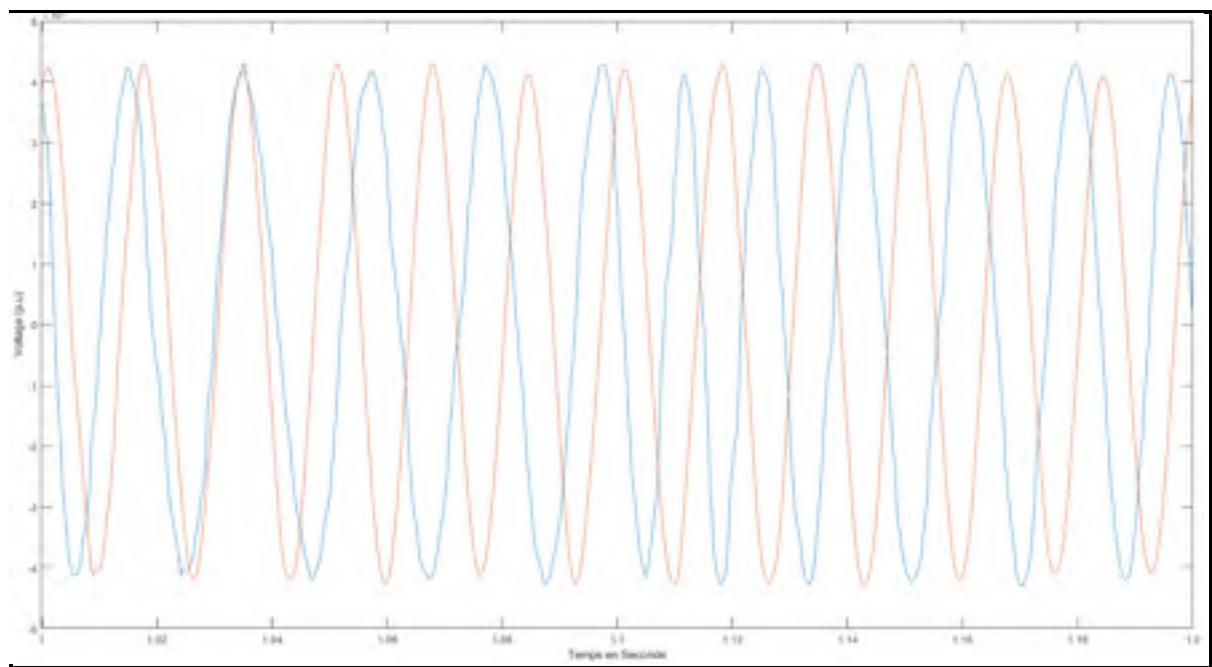


Figure 4.22 Tension (volt) de la simulation instable no 21 en rouge et la simulation stable no 25 en bleu

Les fichiers de simulation correspondant aux numéros de simulation ainsi que les résultats des prédictions sont fournis en annexe.

### 4.3.3 Générateur 3

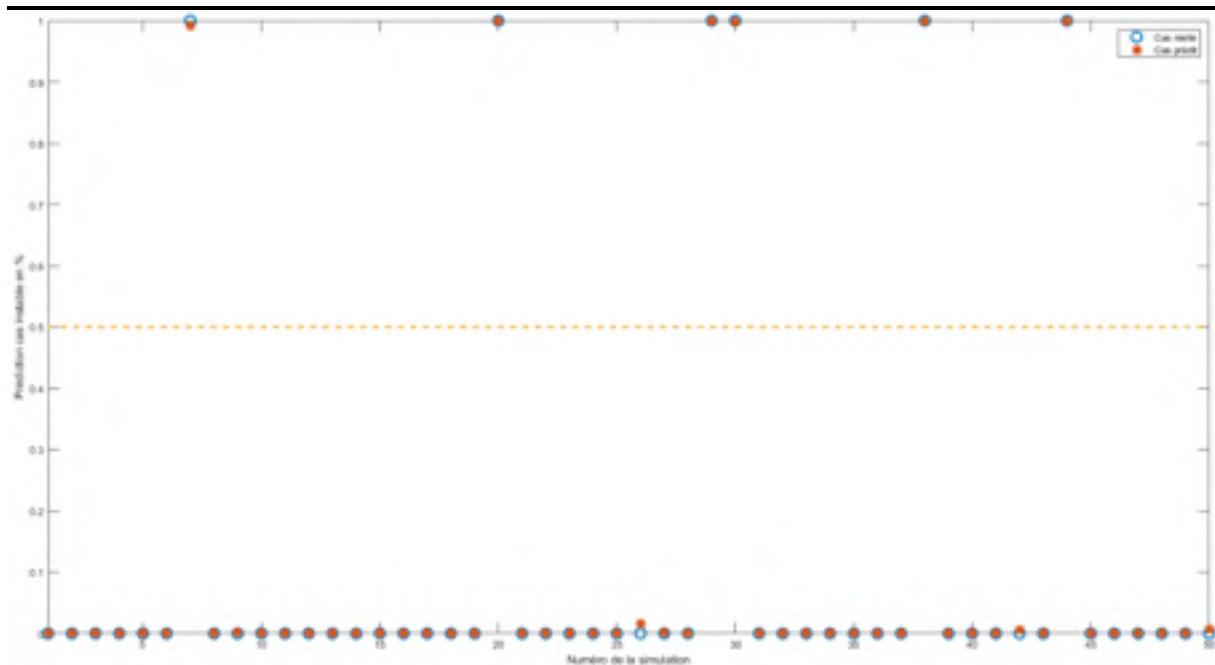


Figure 4.23 Prédictions (en rouge) en pourcentage sur l'ensemble des simulations pour le générateur 3

Tableau 4.28 Matrice de confusion du générateur 3

		Cas prédict		
		0	1	Total prédict
Cas réel	0	174	0	174
	1	0	6	6
	Total réel	174	6	180

Tableau 4.29 Taux mesurés pour la matrice de confusion du générateur 3

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
100.00%	100.00%	100.00%	0.00%	0.00%	0.00%

Concernant le générateur 3, nous pouvons voir qu'il présente les meilleures performances des quatre. Nous pouvons constater que la simulation no 7 est à un seuil de 99.17 % de détection d'instabilité. C'est le score le plus faible dans cet échantillon. Également nous pouvons voir que la simulation 30 a un taux d'instabilité de 1.58 %.

Une des raisons probables est possiblement que c'est une machine avec le plus faible taux d'inertie. Puisque pour une même perturbation sur le réseau cette machine tend à être plus instable, cela permet d'obtenir un plus grand échantillon de cas instables pour cette machine. L'algorithme d'apprentissage profond a donc plus de cas d'instabilité pour son ensemble d'apprentissage.

En considérant plus en détail la matrice de confusion, on peut observer qu'il n'y a pas de cas de faux positif dans un échantillon de 50 cas de validation. Ceci est un excellent score. Par contre avec un échantillon plus grand, il est certain que ce taux de faux positif ne restera pas à 0. Ceci dit, cela demeure un très bon résultat pour un apprentissage machine avec ce nombre d'échantillons.

L'apprentissage du générateur 3 démontre qu'il y a eu une excellente généralisation de l'ensemble d'entraînement à l'ensemble de test. On peut constater que les taux de prédiction sont très élevés autant pour les cas de détection de simulations stables que les cas de simulations instables. Sur les 50 simulations dans l'ensemble de test, tous les cas déclarés stables se situent en dessous de 2 % et tous les cas déclarés instables se situent en dessous de 99 % d'instabilité. Ceci témoigne de la spécificité de l'algorithme d'apprentissage profond MALSTM-FCNs.

Voici à la figure suivante la comparaison entre les vitesses angulaires et les phases de deux simulations de la machine no 3. La simulation no 7 en rouge qui est prédicté instable et la simulation no 40 en bleu qui est prédicté stable.

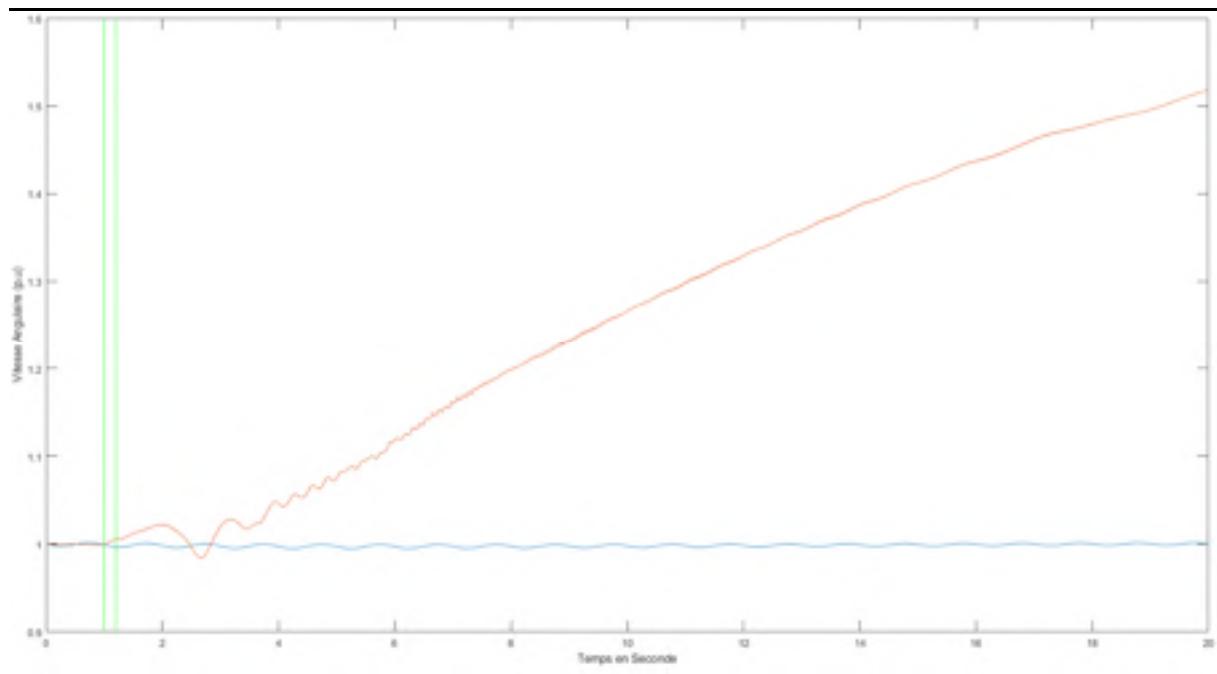


Figure 4.24 Vitesse Angulaire (pu) de la simulation prédictive instable no 7 en rouge et la simulation prédictive stable no 40 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

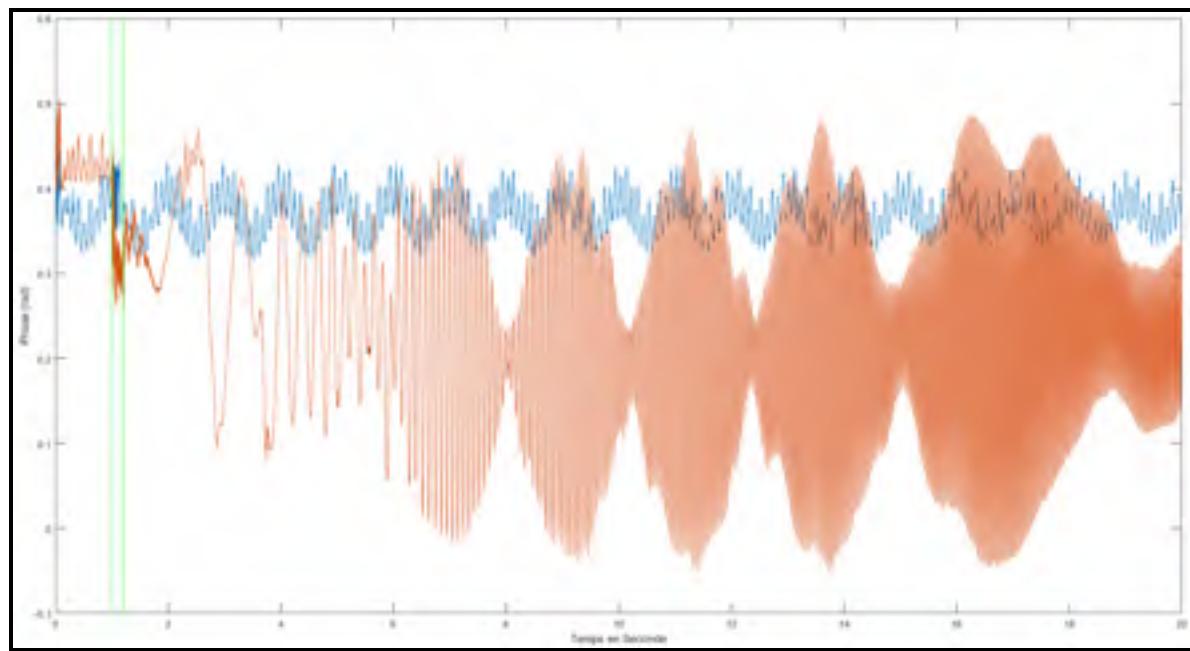


Figure 4.25 Vitesse Angulaire (pu) de la simulation prédictée instable no 7 en rouge et la simulation prédictée stable no 40 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

Les fichiers de simulation correspondant aux numéros de simulation ainsi que les résultats des prédictions sont fournis en annexe.

#### 4.3.4 Générateur 4

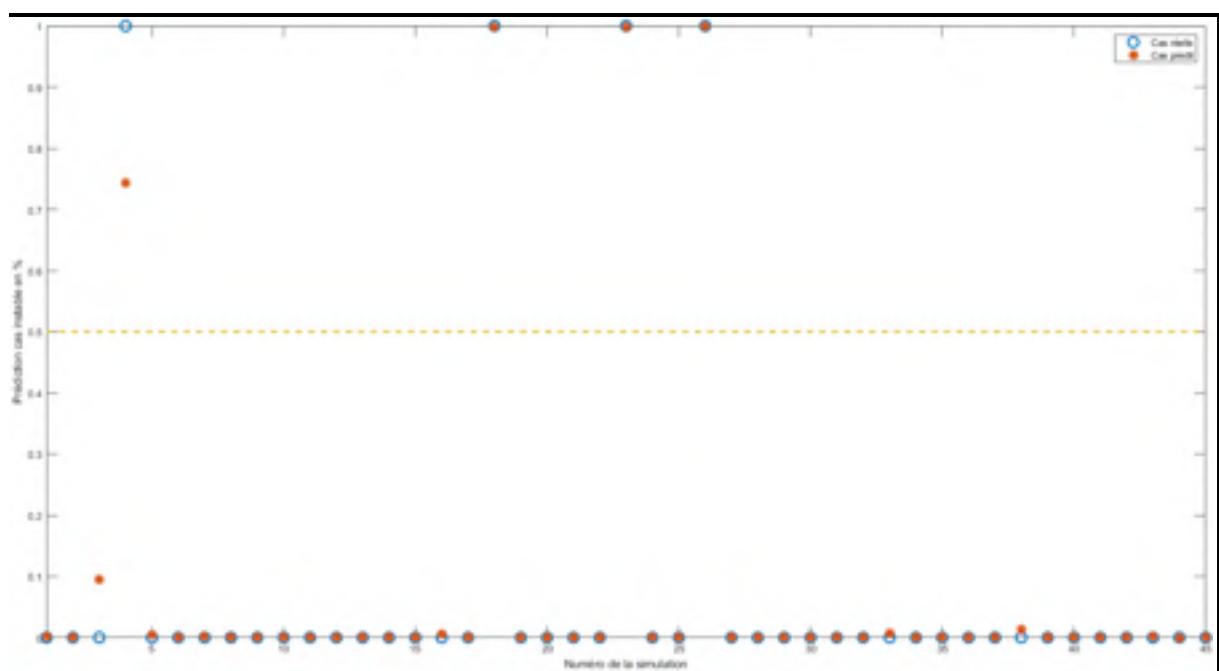


Tableau 4.31 Taux mesurés pour la matrice de confusion du générateur 4

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de faux positif	Taux de faux négatif
100.00%	100.00%	100.00%	0.00%	0.00%	0.00%

Le générateur 4 obtient de bons résultats par rapport à la sensibilité. Tous les cas de vrais négatifs des simulations instables sont détectés et reflétés par un taux de sensibilité de 100 % ce qui est tout à fait remarquable. Cependant le pourcentage de détection d'instabilité pour le cas no 4 est de 74.33 %. En analysant la figure suivante, nous pouvons mieux comprendre pourquoi. La figure ci-dessous est la comparaison entre les vitesses angulaires de la simulation no 4 en rouge qui est prédictée instable et la simulation no 20 en bleu qui est prédictée stable.

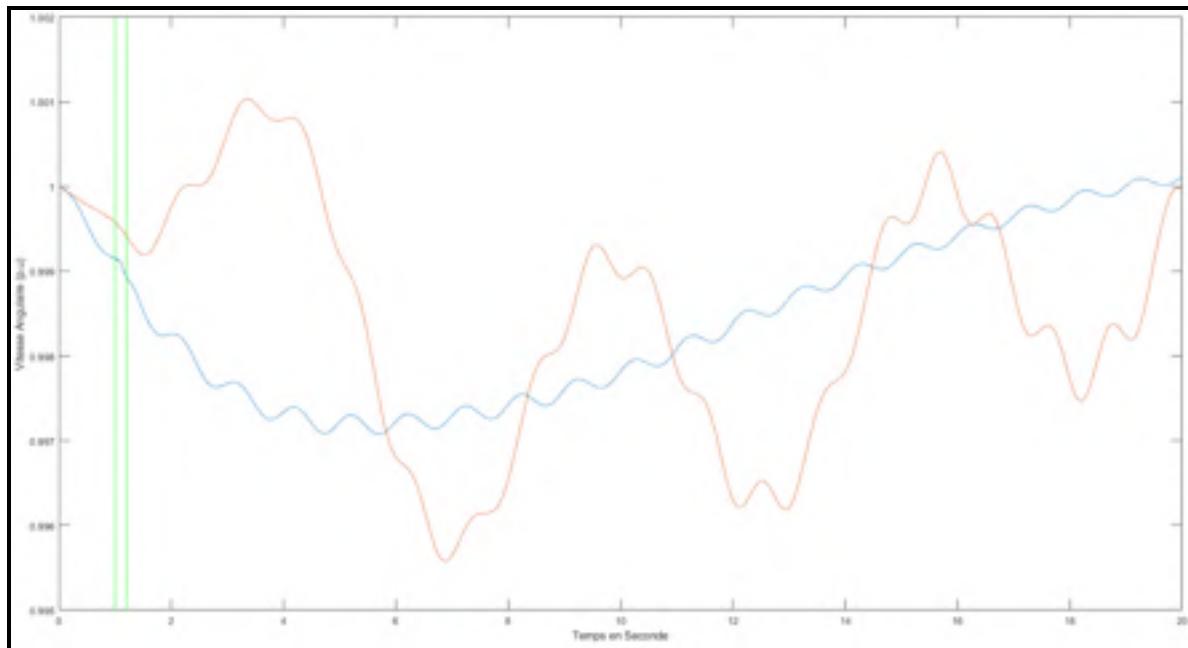


Figure 4.27 Vitesse Angulaire (pu) de la simulation prédictée instable no 4 en rouge et la simulation prédictée stable no 20 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

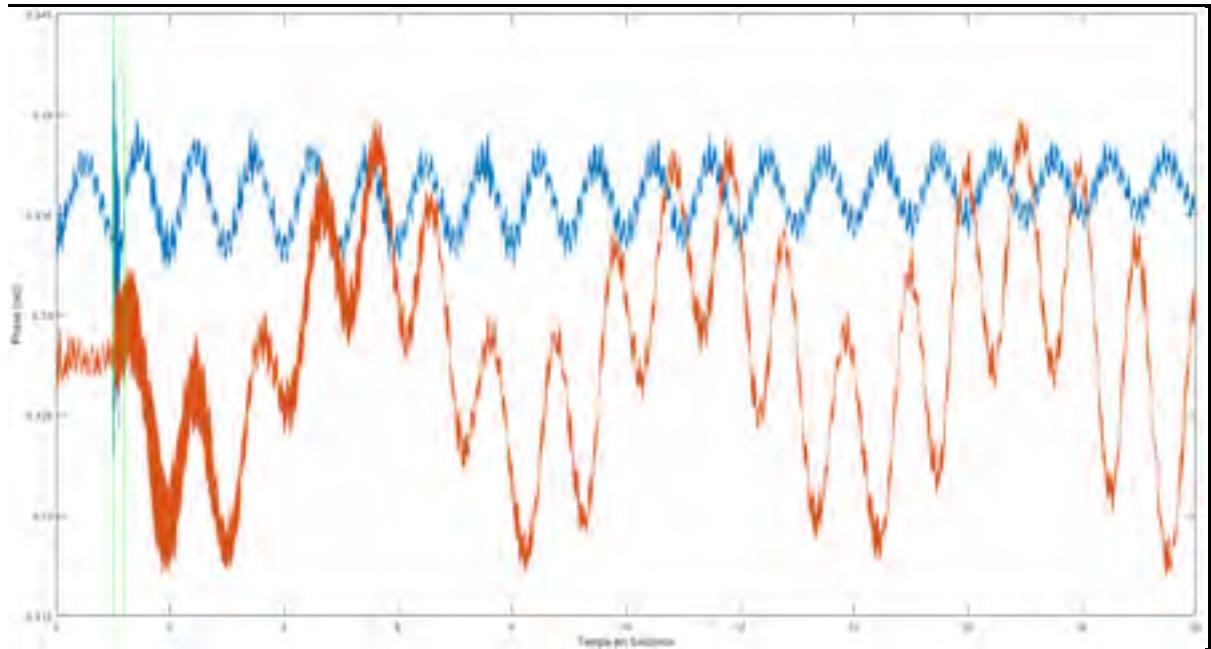


Figure 4.28 Phase (radian) de la simulation prédictive instable no 4 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prévision encadrée en vert

En analysant les deux figures précédentes, on constate la grande similarité des deux types de comportement de la machine no 4. La simulation no 4 a été classée à la limite de l'instabilité. Malgré la grande similarité, l'algorithme a su les classer de façon appropriée mais avec une certitude moindre pour la simulation no 4.

Pour le cas en rouge de la figure suivante, la prévision est sûre de la stabilité et de l'instabilité.

À titre de comparaison, voici à la figure suivante la comparaison entre les vitesses angulaires de la simulation no 18 en rouge qui est prédictive instable et la simulation no 20 de nouveau en bleu qui est prédictive stable.

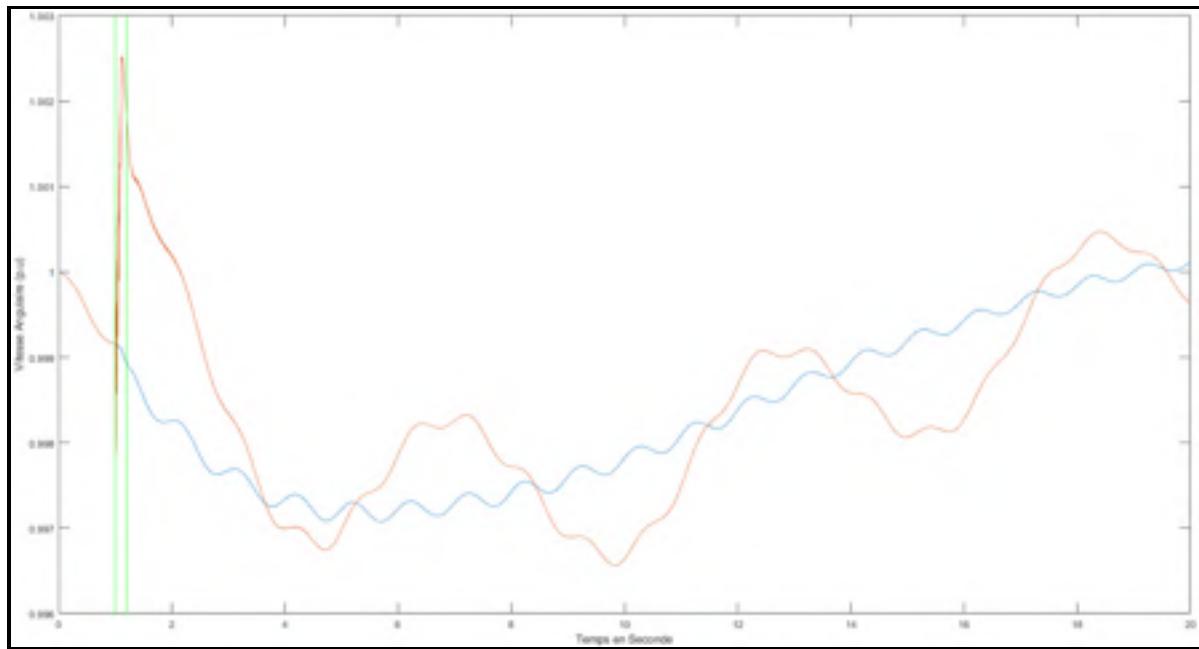


Figure 4.29 Vitesse Angulaire (pu) de la simulation prédictive instable no 18 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

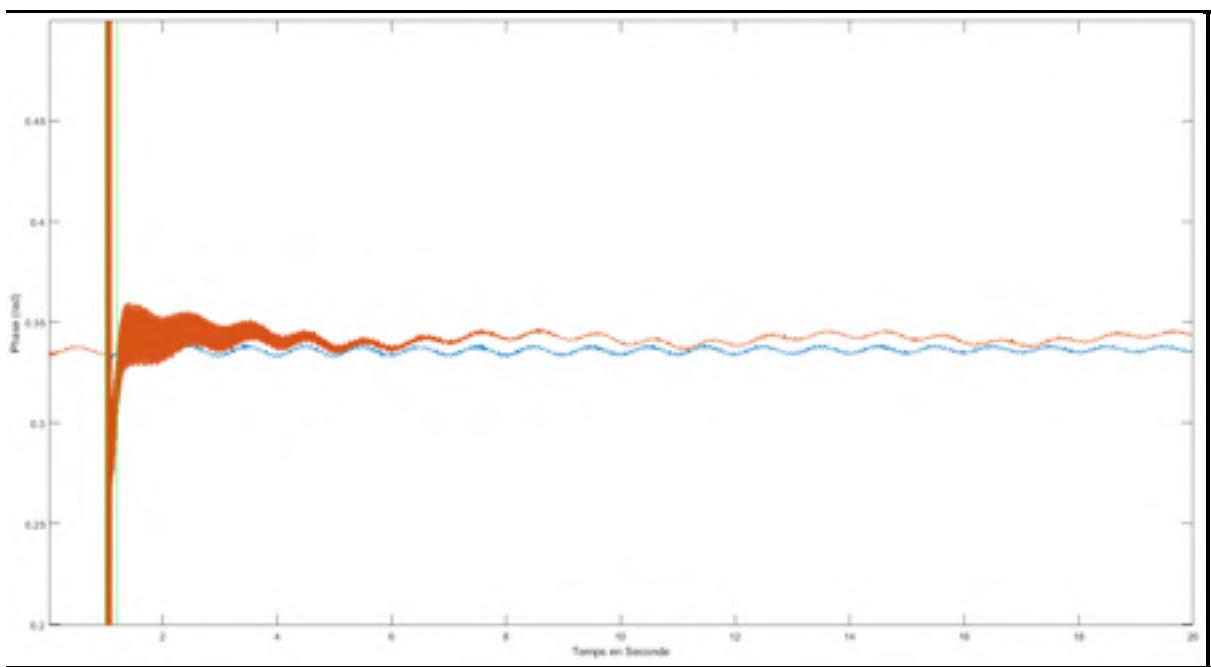


Figure 4.30 Phase (radian) de la simulation prédictive instable no 18 en rouge et la simulation prédictive stable no 20 en bleu. Portion de la courbe utilisée pour la prédiction encadrée en vert

La différence entre ces deux courbes est bien plus grande de même que la certitude de l'algorithme du cas d'instabilité à 99.84 %.

Finalement, après avoir analysé en détail les quatre générateurs, nous pouvons voir que les instabilités sont mieux détectées pour les générateurs avec une faible inertie plutôt que les générateurs avec une forte inertie.

Tous ces résultats sont représentatifs de la capacité de généralisation du réseau de neurones, car ces ensembles de test représentent 25% des simulations.

Les fichiers de simulation correspondant aux numéros de simulation ainsi que les résultats des prédictions sont fournis en annexe.



## CONCLUSION

Le but de cette étude était de savoir si c'est possible de faire un relais de protection de synchronisme en temps réel pour les problèmes de stabilité des réseaux électriques grâce à l'apprentissage machine et le Deep Learning.

La nouveauté de cette approche provient de l'utilisation de données brutes sans filtrage au préalable et du faible nombre de variables d'entrée. Ces données sont les valeurs temporelles des tensions et courants triphasés ainsi que les puissances actives et réactives triphasées, ce qui donne au total 12 variables d'entrées. Cette approche permet de faire un relais avec une très faible demande en ressources informatiques puisqu'il suffit d'évaluer un réseau de neurones en place. Une autre nouveauté par rapport aux autres méthodes d'apprentissage traditionnel est qu'il n'y a pas de «Feature Extraction» (extraction d'attributs) qui permet de modifier les variables d'entrée pour en extraire des informations, comme la fréquence ou un estimé de l'état des génératrices.

Afin de parvenir à une réponse à cette question, nous avons effectué 180 simulations grâce au logiciel de simulation de transitoire électromagnétique EMTP-RV. Nous avons opté pour utiliser un réseau d'étude tiré de l'ouvrage de P. M. Anderson (1998). Les simulations sont d'une durée de 20 secondes et un échantillonnage de 3840 Hz.

Un total de 180 simulations ont été réalisées sur le réseau d'étude. Au préalable, nous avons effectué trois Load-Flow pour mettre le réseau dans trois conditions initiales distinctes. Pour chaque simulation, il y a quatre points de mesures situés à des endroits différents sur le réseau qui représentent l'emplacement des relais de perte de synchronisme.

Dans ce réseau d'étude, il y a trois zones. Une zone centrale qui représente le réseau sous exploitation avec les quatre relais de perte de synchronisme. Deux relais sont placés en amont de deux centres de production dans cette région directement. Les deux autres relais sont placés aux extrémités de la zone d'étude à l'interconnexion entre les deux autres zones. Ces dernières

représentent des réseaux voisins et sont modélisées par une grande charge à un grand générateur. Les relais aux interconnexions jouent le rôle d'assurer la stabilité à l'interface entre différents réseaux pour isoler le réseau d'étude si la fréquence diverge entre les réseaux voisins et le réseau d'étude.

Dans les quatre emplacements des relais de perte de synchronisme, nous prenons des mesures de tension triphasée temporelle, de courant triphasé temporel et de puissance active et réactive temporelle. Ces mesures sont acquisitionnées sur une durée de 20 secondes. Au début de chaque simulation, il y a un défaut appliqué à différents endroits dans le réseau d'étude à partir d'une seconde après le début de la simulation. Différents types de défauts sont présents soit des défauts monophasés, triphasés et de durées de 6 cycles à 12 cycles.

Une fois les simulations réalisées, elles sont importées avec EMTP en format .MAT dans le logiciel MATLAB. Ces simulations sont ensuite séparées par générateur et classées en fonction de leur stabilité grâce aux critères de centre d'inertie d'angle et de vitesse. Le tableau ci-dessous résume l'ensemble des classements effectués par générateur.

Tableau 4.32 Ensemble des simulations stables et instables

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	166	156	152	164
Instables	14	24	28	16

Suite à ce classement, l'ensemble des simulations est divisé en sous-ensembles. Le premier sous-ensemble est l'ensemble d'entraînement qui contient 75 % des simulations pour un générateur donné, soit environ 135 simulations. L'ensemble d'entraînement sert à l'apprentissage de l'algorithme d'apprentissage profond MALSTM-FCN. L'ensemble d'entraînement contient également 75 % des cas instables répartis aléatoirement. L'autre sous-ensemble est l'ensemble de test qui sert à évaluer l'apprentissage et la capacité de

généralisation de l'algorithme d'apprentissage profond. Cet ensemble contient des cas de simulations inconnues pour l'algorithme.

Suite à l'entraînement et l'évaluation de l'apprentissage de l'algorithme, on remarque une meilleure performance à classer les simulations instables des deux générateurs à l'intérieur de la zone d'étude. La raison probable de cette meilleure performance est que ces deux générateurs sont de plus petites tailles et sont donc plus faciles à déstabiliser. On peut constater au précédent que ces deux générateurs, le générateur 2 et 3, possèdent plus de cas instables, ce qui ajoute plus de cas instables à l'ensemble d'apprentissage et de test. Ceci donne l'opportunité à l'algorithme d'apprentissage profond à voir une plus grande variété de cas instables et donc à mieux les classer. Les deux autres générateurs étaient plus difficiles à déstabiliser et leur instabilité était moins grande donc plus subtile à détecter, ce qui explique une moins bonne performance de détection.

L'avantage de cette méthode est qu'on peut entraîner le réseau de neurones jusqu'à la précision de détection voulue puisque l'exécution d'un réseau de neurones qui a appris est très rapide et facile à exécuter dans une plateforme en temps réel. Elle exige très peu de ressources à exécuter une fois apprise. En contrepartie, la plupart du travail vient dans la phase d'apprentissage où un grand nombre de simulations très lourdes en espace mémoire doivent être produites pour un emplacement réseau spécifique. Le relais doit apprendre, et cela peut prendre des heures ou des jours d'apprentissage. Mais ce temps d'apprentissage est plus court que le temps de développement d'un relais conventionnel. Un autre avantage est que si le relais a bien appris à généraliser le comportement de perte de synchronisme, il est assez flexible pour s'adapter à des changements dans le réseau à travers le temps ou à différents emplacements réseau. Un autre avantage provient du fait qu'il n'y a pas de subjectivité humaine pour prédéfinir ou catégoriser les cas stables et instables puisque le classement des cas se fait par la méthode du centre d'inertie de vitesse et d'angle.

Il y a une différence entre la méthode étudiée dans ce travail pour la réalisation d'un relais de perte de synchronisme et les relais conventionnels. Dans les méthodes conventionnelles, il y a

un fondement théorique aux mesures et algorithmes qui déterminent s'il y a perte de synchronisme. Pour l'algorithme d'apprentissage profond, il n'y a pas de transparence sur la méthode employée pour détecter le synchronisme. La méthode est déterminée lors de l'entraînement et n'est pas connue. Cela a l'avantage d'utiliser toutes les informations les plus pertinentes pour arriver à faire une détermination en fonction de la topologie du réseau cible mais cela a également l'inconvénient que le relais agit de manière inconnue et particulièrement sur des topologies de réseaux différentes.

Suivant un changement topologique important comme par exemple l'ajout d'une ligne ou d'une charge, il est nécessaire de refaire la phase d'entraînement afin que le relais ait la meilleure probabilité de succès à détecter une perte de synchronisme. Ceci peut occasionner la nécessité de mettre à jour le relais lorsque de nouvelles sources d'énergie distribuée s'ajoutent sur le réseau ou lorsque la charge change dans le temps. Par contre, du point de vue d'un réseau de transport, ceci peut représenter beaucoup de temps et il est concevable qu'une mise à jour annuelle des poids, suite à de nouveaux entraînements, soit suffisante pour conserver la performance optimale du relais.

Cet algorithme pourrait bien se prêter à une implémentation dans un automate programmable de type RTAC de SEL. Cet automate aurait les performances pour exécuter rapidement l'algorithme d'apprentissage profond préalablement entraîné. De plus, il est possible de télécharger dans le RTAC les mises à jour des poids du réseau de neurones dans le cas où la topologie du réseau aurait changé. La plus grande limitation du RTAC serait la période d'échantillonnage du signal qui est présentement limitée à 1000 Hz. Il resterait à déterminer si l'algorithme préserve ces performances sur des signaux échantillonnés à 1000 Hz.

Idéalement, dans un travail de plus longue haleine, il faudrait comparer les performances sur quelques cas typiques de ce nouveau relais de perte de synchronisme avec les meilleurs relais disponibles actuellement, par exemple le relais de perte de synchronisme par logique floue d'Hydro-Québec. Ces cas devraient inclure des faux positifs et négatifs générés pour chaque type de technologie de détection de perte de synchronisme afin de voir si ces technologies

peuvent se complémenter. Si ces technologies se complémentent, elles ouvrent la porte à une implantation d'un relais à technologie hybride qui pourrait permettre la réalisation de relais hautement fiables.

En résumé, l'apprentissage profond offre beaucoup de promesses pour la détection de perte de synchronisme et pour développer des relais en temps réel. Toutefois cette technologie a davantage besoin d'études pour découvrir son potentiel. Avec l'accroissement de la puissance des ressources informatiques et l'avancement des méthodes d'apprentissage machine, cette technologie offre un réel potentiel pour les problèmes de stabilité de réseau.



## **RECOMMANDATIONS**

Une avenue de recherche intéressante serait de voir si avec une base d'apprentissage composée de 50 % de cas stables et 50% de cas instables, on parviendrait à de meilleurs résultats sur l'ensemble de test. Puisque si peu d'événements instables sont présents dans l'ensemble d'apprentissage, il est possible que l'algorithme ait moins l'occasion d'apprendre à discriminer les cas instables. Une autre avenue serait d'explorer l'utilisation des valeurs RMS au lieu d'avoir des valeurs temporelles de données. Il se pourrait qu'on parvienne à des résultats semblables avec des valeurs RMS si les informations apprises sur la stabilité sont préservées. Ceci pourrait être une avenue de recherche intéressante. Une autre possibilité de recherche est d'utiliser uniquement des valeurs de courant et de tension. Puisqu'en théorie, les informations sur la puissance active et réactive sont déjà contenues dans la tension et le courant, on pourrait réduire la taille de l'échantillon si on se limite à la tension et au courant. Une autre avenue possible serait d'utiliser une méthode de classement manuelle des catégories plutôt qu'une méthode automatique.

L'apprentissage machine est reconnu pour son habileté à modéliser les capacités humaines de classer certaines données lorsque les cas de simulation sont au préalable classés stables ou instables par un être humain. L'algorithme de Deep Learning peut apprendre à partir de cette capacité de classement manuel. Enfin, il serait utile de voir si on peut reproduire ces données avec moins d'information dans le temps, c'est-à-dire d'avoir un temps de réponse plus rapide et un temps de traitement plus court. Il est possible que l'algorithme ait déjà la plupart des informations pour prendre une décision bien avant le temps établi dans notre étude.



## ANNEXE I

### LISTE DES SIMULATIONS EN FONCTION DES FICHIERS DES GÉNÉRATEURS

Tableau-A I-1 L'ensemble de test pour le générateur 1

Générateur 1	
1	AndersonParametric_1167_Monophase_LigneD_Long.mat
2	AndersonParametric_1031_Biphasse_BarreB1_Long.mat
3	AndersonParametric_1147_Biphasse_LigneD_Long.mat
4	AndersonParametric_1176_Monophase_BarreB3_Court.mat
<b>5</b>	<b>AndersonParametric_1079_Tripahse_BarreB5_Long.mat</b>
6	AndersonParametric_1158_Biphasse_BarreB4_Court.mat
7	AndersonParametric_1134_Tripahse_BarreB2_Court.mat
8	AndersonParametric_1143_Biphasse_LigneB_Long.mat
<b>9</b>	<b>AndersonParametric_1011_Tripahse_BarreB1_Long.mat</b>
10	AndersonParametric_1078_Tripahse_BarreB4_Court.mat
11	AndersonParametric_1091_Biphasse_BarreB1_Long.mat
12	AndersonParametric_1082_Biphasse_LigneA_Court.mat
13	AndersonParametric_1035_Biphasse_BarreB3_Long.mat
14	AndersonParametric_1180_Monophase_BarreB5_Court.mat
15	AndersonParametric_1055_Monophase_BarreB3_Long.mat
16	AndersonParametric_1060_Monophase_BarreB5_Court.mat
17	AndersonParametric_1146_Biphasse_LigneC_Court.mat
18	AndersonParametric_1088_Biphasse_LigneD_Court.mat
19	AndersonParametric_1133_Tripahse_BarreB2_Long.mat
20	AndersonParametric_1104_Monophase_LigneB_Court.mat
21	AndersonParametric_1005_Tripahse_LigneC_Long.mat
22	AndersonParametric_1093_Biphasse_BarreB2_Long.mat
23	AndersonParametric_1121_Tripahse_LigneA_Long.mat
24	AndersonParametric_1106_Monophase_LigneC_Court.mat
25	AndersonParametric_1128_Tripahse_LigneD_Court.mat

<b>Générateur 1</b>	
26	AndersonParametric_1173_Monophase_BarreB2_Long.mat
27	AndersonParametric_1130_Tripahse_LigneF_Court.mat
28	AndersonParametric_1114_Monophase_BarreB2_Court.mat
29	AndersonParametric_1150_Biphase_LigneF_Court.mat
<b>30</b>	<b>AndersonParametric_1039_Biphase_BarreB5_Long.mat</b>
31	AndersonParametric_1178_Monophase_BarreB4_Court.mat
32	AndersonParametric_1004_Tripahse_LigneB_Court.mat
33	AndersonParametric_1118_Monophase_BarreB4_Court.mat
34	AndersonParametric_1170_Monophase_LigneF_Court.mat
35	AndersonParametric_1083_Biphase_LigneB_Long.mat
36	AndersonParametric_1086_Biphase_LigneC_Court.mat
37	AndersonParametric_1102_Monophase_LigneA_Court.mat
38	AndersonParametric_1142_Biphase_LigneA_Court.mat
39	AndersonParametric_1064_Tripahse_LigneB_Court.mat
40	AndersonParametric_1071_Tripahse_BarreB1_Long.mat
41	AndersonParametric_1168_Monophase_LigneD_Court.mat
42	AndersonParametric_1165_Monophase_LigneC_Long.mat
43	AndersonParametric_1138_Tripahse_BarreB4_Court.mat
44	AndersonParametric_1025_Biphase_LigneC_Long.mat

Tableau-A I-2 L'ensemble de test pour le générateur 2

<b>Générateur 2</b>	
<b>1</b>	<b>AndersonParametric_1013_Tripahse_BarreB2_Long.mat</b>
2	AndersonParametric_1016_Tripahse_BarreB3_Court.mat
3	AndersonParametric_1027_Biphase_LigneD_Long.mat
4	AndersonParametric_1029_Biphase_LigneF_Long.mat
5	AndersonParametric_1030_Biphase_LigneF_Court.mat
6	AndersonParametric_1037_Biphase_BarreB4_Long.mat
<b>7</b>	<b>AndersonParametric_1040_Biphase_BarreB5_Court.mat</b>
8	AndersonParametric_1045_Monophase_LigneC_Long.mat
9	AndersonParametric_1046_Monophase_LigneC_Court.mat
10	AndersonParametric_1051_Monophase_BarreB1_Long.mat
11	AndersonParametric_1056_Monophase_BarreB3_Court.mat
12	AndersonParametric_1059_Monophase_BarreB5_Long.mat
13	AndersonParametric_1067_Tripahse_LigneD_Long.mat
14	AndersonParametric_1069_Tripahse_LigneF_Long.mat
15	AndersonParametric_1075_Tripahse_BarreB3_Long.mat
<b>16</b>	<b>AndersonParametric_1079_Tripahse_BarreB5_Long.mat</b>
17	AndersonParametric_1081_Biphase_LigneA_Long.mat
18	AndersonParametric_1083_Biphase_LigneB_Long.mat
19	AndersonParametric_1087_Biphase_LigneD_Long.mat
20	AndersonParametric_1098_Biphase_BarreB4_Court.mat
<b>21</b>	<b>AndersonParametric_1100_Biphase_BarreB5_Court.mat</b>
22	AndersonParametric_1104_Monophase_LigneB_Court.mat
23	AndersonParametric_1110_Monophase_LigneF_Court.mat
24	AndersonParametric_1116_Monophase_BarreB3_Court.mat
25	AndersonParametric_1122_Tripahse_LigneA_Court.mat
26	AndersonParametric_1141_Biphase_LigneA_Long.mat
27	AndersonParametric_1144_Biphase_LigneB_Court.mat
28	AndersonParametric_1145_Biphase_LigneC_Long.mat

<b>Générateur 2</b>	
29	AndersonParametric_1149_BiphasE_LigneF_Long.mat
30	AndersonParametric_1150_BiphasE_LigneF_Court.mat
<b>31</b>	<b>AndersonParametric_1154_BiphasE_BarreB2_Court.mat</b>
<b>32</b>	<b>AndersonParametric_1160_BiphasE_BarreB5_Court.mat</b>
33	AndersonParametric_1163_Monophase_LigneB_Long.mat
34	AndersonParametric_1174_Monophase_BarreB2_Court.mat

Tableau-A I-3 L'ensemble de test pour le générateur 3

<b>Générateur 3</b>	
1	AndersonParametric_1001_Tripahse_LigneA_Long.mat
2	AndersonParametric_1005_Tripahse_LigneC_Long.mat
3	AndersonParametric_1006_Tripahse_LigneC_Court.mat
4	AndersonParametric_1013_Tripahse_BarreB2_Long.mat
5	AndersonParametric_1015_Tripahse_BarreB3_Long.mat
6	AndersonParametric_1016_Tripahse_BarreB3_Court.mat
7	<b>AndersonParametric_1019_Tripahse_BarreB5_Long.mat</b>
8	AndersonParametric_1021_Biphase_LigneA_Long.mat
9	AndersonParametric_1029_Biphase_LigneF_Long.mat
10	AndersonParametric_1034_Biphase_BarreB2_Court.mat
11	AndersonParametric_1036_Biphase_BarreB3_Court.mat
12	AndersonParametric_1041_Monophase_LigneA_Long.mat
13	AndersonParametric_1051_Monophase_BarreB1_Long.mat
14	AndersonParametric_1053_Monophase_BarreB2_Long.mat
15	AndersonParametric_1055_Monophase_BarreB3_Long.mat
16	AndersonParametric_1061_Tripahse_LigneA_Long.mat
17	AndersonParametric_1062_Tripahse_LigneA_Court.mat
18	AndersonParametric_1069_Tripahse_LigneF_Long.mat
19	AndersonParametric_1074_Tripahse_BarreB2_Court.mat
20	<b>AndersonParametric_1077_Tripahse_BarreB4_Long.mat</b>
21	AndersonParametric_1082_Biphase_LigneA_Court.mat
22	AndersonParametric_1084_Biphase_LigneB_Court.mat
23	AndersonParametric_1085_Biphase_LigneC_Long.mat
24	AndersonParametric_1086_Biphase_LigneC_Court.mat
25	AndersonParametric_1088_Biphase_LigneD_Court.mat
26	AndersonParametric_1089_Biphase_LigneF_Long.mat
27	AndersonParametric_1090_Biphase_LigneF_Court.mat
28	AndersonParametric_1095_Biphase_BarreB3_Long.mat

<b>Générateur 3</b>	
<b>29</b>	<b>AndersonParametric_1097_Biphase_BarreB4_Long.mat</b>
<b>30</b>	<b>AndersonParametric_1098_Biphase_BarreB4_Court.mat</b>
31	AndersonParametric_1103_Monophase_LigneB_Long.mat
32	AndersonParametric_1105_Monophase_LigneC_Long.mat
33	AndersonParametric_1108_Monophase_LigneD_Court.mat
34	AndersonParametric_1110_Monophase_LigneF_Court.mat
35	AndersonParametric_1124_Tripahse_LigneB_Court.mat
36	AndersonParametric_1135_Tripahse_BarreB3_Long.mat
37	AndersonParametric_1136_Tripahse_BarreB3_Court.mat
<b>38</b>	<b>AndersonParametric_1139_Tripahse_BarreB5_Long.mat</b>
39	AndersonParametric_1143_Biphase_LigneB_Long.mat
40	AndersonParametric_1146_Biphase_LigneC_Court.mat
41	AndersonParametric_1147_Biphase_LigneD_Long.mat
42	AndersonParametric_1150_Biphase_LigneF_Court.mat
43	AndersonParametric_1152_Biphase_BarreB1_Court.mat
<b>44</b>	<b>AndersonParametric_1157_Biphase_BarreB4_Long.mat</b>
45	AndersonParametric_1162_Monophase_LigneA_Court.mat
46	AndersonParametric_1169_Monophase_LigneF_Long.mat
47	AndersonParametric_1173_Monophase_BarreB2_Long.mat
48	AndersonParametric_1176_Monophase_BarreB3_Court.mat
49	AndersonParametric_1177_Monophase_BarreB4_Long.mat
50	AndersonParametric_1179_Monophase_BarreB5_Long.mat

Tableau-A I-4 L'ensemble de test pour le générateur 4

<b>Générateur 4</b>	
1	AndersonParametric_1054_Monophase_BarreB2_Court.mat
2	AndersonParametric_1174_Monophase_BarreB2_Court.mat
3	AndersonParametric_1033_Biphasse_BarreB2_Long.mat
<b>4</b>	<b>AndersonParametric_1031_Biphasse_BarreB1_Long.mat</b>
5	AndersonParametric_1045_Monophase_LigneC_Long.mat
6	AndersonParametric_1120_Monophase_BarreB5_Court.mat
7	AndersonParametric_1021_Biphasse_LigneA_Long.mat
8	AndersonParametric_1111_Monophase_BarreB1_Long.mat
9	AndersonParametric_1117_Monophase_BarreB4_Long.mat
10	AndersonParametric_1085_Biphasse_LigneC_Long.mat
11	AndersonParametric_1087_Biphasse_LigneD_Long.mat
12	AndersonParametric_1015_Tripahse_BarreB3_Long.mat
13	AndersonParametric_1083_Biphasse_LigneB_Long.mat
14	AndersonParametric_1105_Monophase_LigneC_Long.mat
15	AndersonParametric_1071_Tripahse_BarreB1_Long.mat
16	AndersonParametric_1013_Tripahse_BarreB2_Long.mat
17	AndersonParametric_1106_Monophase_LigneC_Court.mat
<b>18</b>	<b>AndersonParametric_1160_Biphasse_BarreB5_Court.mat</b>
19	AndersonParametric_1115_Monophase_BarreB3_Long.mat
20	AndersonParametric_1130_Tripahse_LigneF_Court.mat
21	AndersonParametric_1102_Monophase_LigneA_Court.mat
22	AndersonParametric_1004_Tripahse_LigneB_Court.mat
<b>23</b>	<b>AndersonParametric_1159_Biphasse_BarreB5_Long.mat</b>
24	AndersonParametric_1065_Tripahse_LigneC_Long.mat
25	AndersonParametric_1147_Biphasse_LigneD_Long.mat
<b>26</b>	<b>AndersonParametric_1140_Tripahse_BarreB5_Court.mat</b>
27	AndersonParametric_1024_Biphasse_LigneB_Court.mat
28	AndersonParametric_1150_Biphasse_LigneF_Court.mat

<b>Générateur 4</b>	
29	AndersonParametric_1072_Tripahse_BarreB1_Court.mat
30	AndersonParametric_1123_Tripahse_LigneB_Long.mat
31	AndersonParametric_1007_Tripahse_LigneD_Long.mat
32	AndersonParametric_1107_Monophase_LigneD_Long.mat
33	AndersonParametric_1023_Biphase_LigneB_Long.mat
34	AndersonParametric_1074_Tripahse_BarreB2_Court.mat
35	AndersonParametric_1058_Monophase_BarreB4_Court.mat
36	AndersonParametric_1047_Monophase_LigneD_Long.mat
37	AndersonParametric_1091_Biphase_BarreB1_Long.mat
38	AndersonParametric_1038_Biphase_BarreB4_Court.mat
39	AndersonParametric_1158_Biphase_BarreB4_Court.mat
40	AndersonParametric_1148_Biphase_LigneD_Court.mat
41	AndersonParametric_1096_Biphase_BarreB3_Court.mat
42	AndersonParametric_1146_Biphase_LigneC_Court.mat
43	AndersonParametric_1028_Biphase_LigneD_Court.mat
44	AndersonParametric_1137_Tripahse_BarreB4_Long.mat
45	AndersonParametric_1151_Biphase_BarreB1_Long.mat

## ANNEXE II

### VALEUR DE PRÉDICTION DE L'ENSEMBLE DE TEST

Tableau-A II-1 Valeur de prédiction pour le générateur 1

Générateur 1		
No. de simulation	Stabilité	Prédiction d'instabilité sur 1
1	0	0.00003194
2	0	0.84008050
3	0	0.00003678
4	0	0.00005010
<b>5</b>	<b>1</b>	<b>0.80320275</b>
6	0	0.00092663
7	0	0.00018177
8	0	0.00032077
<b>9</b>	<b>1</b>	<b>0.89639241</b>
10	0	0.00078430
11	0	0.00163644
12	0	0.00007154
13	0	0.00025716
14	0	0.20233147
15	0	0.00001012
16	0	0.00323231
17	0	0.00004969
18	0	0.00009815
19	0	0.00003291
20	0	0.00011353
21	0	0.00067777
22	0	0.00007417
23	0	0.00009617

<b>Générateur 1</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédition d'instabilité sur 1</b>
24	0	0.00026308
25	0	0.00105792
26	0	0.00002494
27	0	0.00045129
28	0	0.00024123
29	0	0.00155668
<b>30</b>	<b>1</b>	<b>0.04026381</b>
31	0	0.00656540
32	0	0.00100715
33	0	0.00012301
34	0	0.00015985
35	0	0.00022556
36	0	0.00015643
37	0	0.00009702
38	0	0.00003532
39	0	0.00011974
40	0	0.05586246
41	0	0.00009007
42	0	0.00008609
43	0	0.02430527
44	0	0.00071654

Tableau-A II-2 Valeur de prédiction pour le générateur 2

<b>Générateur 2</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédiction d'instabilité sur 1</b>
1	1	<b>0.99998701</b>
2	0	0.00020536
3	0	0.00002866
4	0	0.00178998
5	0	0.00041871
6	0	0.00000920
7	1	<b>0.88284737</b>
8	0	0.00000241
9	0	0.00002118
10	0	0.00000488
11	0	0.00038602
12	0	0.04423462
13	0	0.00000009
14	0	0.00430420
15	0	0.00000086
16	1	<b>0.99824095</b>
17	0	0.00000056
18	0	0.00123347
19	0	0.00000057
20	0	0.03430237
21	1	<b>0.99600071</b>
22	0	0.00001184
23	0	0.00016532
24	0	0.00000954
25	0	0.00003360

<b>Générateur 2</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédition d'instabilité sur 1</b>
26	0	0.00002116
27	0	0.00022522
28	0	0.00000319
29	0	0.01203375
30	0	0.00166578
<b>31</b>	<b>1</b>	<b>0.99882179</b>
<b>32</b>	<b>1</b>	<b>0.99932516</b>
33	0	0.00002235
34	0	0.00067793

Tableau-A II-3 Valeur de prédition pour le générateur 3

<b>Générateur 3</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédition d'instabilité sur 1</b>
1	0	0.00002442
2	0	0.00001676
3	0	0.00006547
4	0	0.00037611
5	0	0.00024509
6	0	0.00048107
<b>7</b>	<b>1</b>	<b>0.99166662</b>
8	0	0.00007708
9	0	0.00226585
10	0	0.00059558
11	0	0.00024934
12	0	0.00002205
13	0	0.00040078

Générateur 3		
No. de simulation	Stabilité	Prédition d'instabilité sur 1
14	0	0.00013136
15	0	0.00000492
16	0	0.00000748
17	0	0.00001592
18	0	0.00020895
19	0	0.00007591
<b>20</b>	<b>1</b>	<b>1.00000000</b>
21	0	0.00002432
22	0	0.00003622
23	0	0.00002248
24	0	0.00003256
25	0	0.00040041
26	0	0.01580828
27	0	0.00149749
28	0	0.00046369
<b>29</b>	<b>1</b>	<b>0.99999928</b>
<b>30</b>	<b>1</b>	<b>0.99870718</b>
31	0	0.00003131
32	0	0.00001009
33	0	0.00003257
34	0	0.00005669
35	0	0.00010137
36	0	0.00104657
37	0	0.00103035
<b>38</b>	<b>1</b>	<b>0.99998999</b>
39	0	0.00014623
40	0	0.00006397

<b>Générateur 3</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédition d'instabilité sur 1</b>
41	0	0.00079877
42	0	0.00611725
43	0	0.00021288
<b>44</b>	<b>1</b>	<b>0.99999774</b>
45	0	0.00005567
46	0	0.00033373
47	0	0.00005388
48	0	0.00003800
49	0	0.00001097
50	0	0.00688792

Tableau-A II-4 Valeur de prédition pour le générateur 4

<b>Générateur 4</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédition d'instabilité sur 1</b>
1	0	0.00172370
2	0	0.00006274
3	0	0.09456631
<b>4</b>	<b>1</b>	<b>0.74332559</b>
5	0	0.00412227
6	0	0.00001066
7	0	0.00242628
8	0	0.00024741
9	0	0.00000030
10	0	0.00000953
11	0	0.00000082
12	0	0.00115376
13	0	0.00001010

Générateur 4		
No. de simulation	Stabilité	Prédiction d'instabilité sur 1
14	0	0.00008001
15	0	0.00001514
16	0	0.00588545
17	0	0.00006056
<b>18</b>	<b>1</b>	<b>0.99841440</b>
19	0	0.00004752
20	0	0.00000078
21	0	0.00001336
22	0	0.00015200
<b>23</b>	<b>1</b>	<b>0.99865794</b>
24	0	0.00000141
25	0	0.00000293
<b>26</b>	<b>1</b>	<b>0.99917883</b>
27	0	0.00048713
28	0	0.00000036
29	0	0.00008696
30	0	0.00003395
31	0	0.00014581
32	0	0.00004517
33	0	0.00687451
34	0	0.00026455
35	0	0.00024405
36	0	0.00029716
37	0	0.00000450
38	0	0.01283652
39	0	0.00007780
40	0	0.00000264

<b>Générateur 4</b>		
<b>No. de simulation</b>	<b>Stabilité</b>	<b>Prédiction d'instabilité sur 1</b>
41	0	0.00002596
42	0	0.00005401
43	0	0.00203545
44	0	0.00000002
45	0	0.00017179

## LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Anderson, Paul M. (1998). *Power System Protection*. Hoboken, N.-J.: Wiley-IEEE Press. 1330 p.
- Burhan, Syla. (2012). *Relais de perte de synchronisme par exploration de données*. (Mémoire de maîtrise, Université Laval, Québec, QC). 183 p.
- Chothani, N. G., Bhalja, B. R., & Parikh, U. B. (2014). New support vector machine-based digital relaying scheme for discrimination between power swing and fault. *IET Generation, Transmission & Distribution*, vol. 8, no. 1, p. 17-25.
- Danish Ali, Syed, (2016). Deep Neural network. Dans *Data science central: the online resource for big data practitioners*. Repéré à <https://www.datasciencecentral.com/profiles/blogs/on-going-developments-and-outlook-for-deep-learning>.
- Fawaz, Ismail, Forestier, G., Weber, J. et al. (2019). Deep Learning for time series classification: a review. *Data Min Knowl Disc* 33, 917–963. Repéré à <https://doi.org/10.1007/s10618-019-00619-1>.
- fdelache (2017). A diagram for a one-unit long short-term memory. Dans *Wikipedia*. Repéré à <https://commons.wikimedia.org/w/index.php?curid=60149410>
- fdelache (2017). Recurrent neural network unfold. Dans *Wikipedia*. Repéré à [https://commons.wikimedia.org/wiki/File:Recurrent\\_neural\\_network\\_unfold.svg](https://commons.wikimedia.org/wiki/File:Recurrent_neural_network_unfold.svg)
- Glosser.ca. (2013). Artificial neural network with layer coloring. Dans *Wikipedia*. Repéré à [https://upload.wikimedia.org/wikipedia/commons/4/46/Colored\\_neural\\_network.svg](https://upload.wikimedia.org/wikipedia/commons/4/46/Colored_neural_network.svg)
- Hashim, H., Abidin, I., Yap, K.S., Musirin, I., & Zulkepal, M. (2010). An Analysis of Transient Stability Using Center-of-Inertia: Angle and Speed. Power and Energy (PECon). *2010 IEEE International Conference on Power and Energy*, Kuala Lumpur, 2010, pp. 402-407, doi: 10.1109/PECON.2010.5697617
- Kamwa, Innocent, Samantaray, S. R., Joos, Geza. (2010). Catastrophe predictors from ensemble decision-tree learning of wide-area severity indices. *IEEE Transactions On Smart Grid*, vol. 1, no 2, p. 144-158.
- Karim, Fazole, Majumdar, Somshubra, Darabi, Houshang, & Harford, Samuel. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, vol. 116, p. 237-245.

- Kirby, B., Zou, L., Cao, J., Kamwa, I., Heniche, A., & Dobrescu, M. (2011). Development of predictive out of step relay using a model based design process. *Proc. 2nd IEEE PES Int. Conf. Exhib. Innovative Smart Grid Tech.*, Manchester, U.K., 5–7 dec., 2011, p. 1–6.
- Kundur, P. (1994). *Power Systems Stability and Control*. Toronto: McGraw-Hill Professional. 1176 p.
- Mazhari, S. M., Chung, C. Y., Kamwa, I. (2018). A Hybrid Fault Cluster and Thevenin Equivalent Based Framework for Rotor Angle Stability Prediction. *IEEE Transactions on Power Systems*, vol. 33, no 5, p. 5594–5603.
- Perron, Mathieu, Heniche, Annissa, Dobrescu, Manuela, Paré, Daniel, Robichaud, Jean-Luc & Kamwa, Innocent. (2018). *Innovative response-based special protection scheme for imminent loss of synchronism detection with surge arrester protection and energy monitoring functions*. CIGRE session, Conference paper, Paris, France, 10 sept. 2018. 12 p.
- Qef (2008). The logistic sigmoid function. Dans *Wikipedia*. Repéré à [https://en.wikipedia.org/wiki/Sigmoid\\_function#/media/File:Logistic-curve.svg](https://en.wikipedia.org/wiki/Sigmoid_function#/media/File:Logistic-curve.svg)
- Sauer, P.W. & Pai, M.A. (1998). *Power System Dynamics and Stability*. New Jersey: Prentice Hall, 1998.
- Sharma, Sagar. (2017). What the hell is perceptron? Dans *Towards data science*. Repéré à <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>
- The Mark I Perceptron, an early pattern recognition system at the Cornell Aeronautical Laboratory. (2018). Dans *Wikipedia*. Repéré à [https://upload.wikimedia.org/wikipedia/en/5/52/Mark\\_I\\_perceptron.jpeg](https://upload.wikimedia.org/wikipedia/en/5/52/Mark_I_perceptron.jpeg)
- Witten, I. & Eibe, F. (2005). *Data Mining Practical Machine Learning Tools and Techniques* (2e éd.). Burlington, Mass: Morgan Kaufmann. 664 p.
- Zhang, Aston, Lipton, Zachary C., Li, Mu, & Smola, Alexander J. (2019). *Dive into Deep Learning Release 0.7.1*. Chap. 9, p. 360. [https://d2l.ai/chapter\\_recurrent-modern/deep-rnn.html](https://d2l.ai/chapter_recurrent-modern/deep-rnn.html)
- Zhang, S., Wang, Y., Liu, M., & Bao, Z. (2018). Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM. *IEEE Access*, vol. 6, p. 7675-7686.