جامعة سيدي محمد بن عبد الله بفاس
UNIVERSITÉ SIDI MOHAMED BEN ABDELLAH DE FES

كلية العلوم والتقنيات فاس
FACULTÉ DES SCIENCES ET TECHNIQUES DE FÈS

N° d'ordre : 48/2023

Centre des Etudes Doctorales : Sciences et Techniques et Sciences Médicales
Formation Doctorale : SISPMI (Sciences de l'Ingénieur, Sciences Physiques, Mathématiques et Informatique),
Discipline : Sciences de l'Ingénieur
Spécialité* : Mécatronique
Laboratoire : Techniques Industrielles

Thèse pour l'obtention de DOCTORAT

Elaboration d'un superviseur basé sur les réseaux de neurones pour le diagnostic et le pronostic des fautes dans les Systèmes à Evènements Discrets.

Préparée par : Mr AMRI Omar

Soutenue le 08 Juillet 2023 devant le jury :

Nom et Prénom	Grade	Etablissement	Qualité
HAMEDI L'habib	PES	Faculté des Sciences et Techniques de Fès	Président
DEROUICH Aziz	PES	Ecole Supérieur de Technologie de Fès	Rapporteur & examinateur
MOUHSEN Ahmed	PES	Faculté des Sciences et Techniques de Settat	Rapporteur & examinateur
SOULHI Aziz	PES	Ecole Nationale Supérieur des Mines de Rabat	Rapporteur & examinateur
EN-NADI Abdelali	PES	Faculté des Sciences et Techniques de Fès	Examinateur
BELMAJDOUB Fouad	PES	Faculté des Sciences et Techniques de Fès	Directeur de thèse

Dédicace

A mes chers parents pour tout ce que vous avez fait pour moi ;

A mes chers cousins, pour vos encouragements et votre soutien ;

A toute ma famille ;

A tous mes professeurs ;

A tous mes ami(e)s ;

A tous ceux qui sont proches par le cœur et l'esprit ;

Je dédie ce travail qui symbolise pour nous, l'aboutissement

d'une expérience

de vie riche en apprentissage humain et intellectuel.

Remerciements

La thèse de doctorat était pour moi un grand défi et une aventure unique, pleine de surprise et des éblouissements qui mon permis d'enrichir et de diversifier mes compétences personnelles et professionnel. Cette thèse représente l'aboutissement d'un chapitre de ma vie. Elle n'aurait cependant pas été vécue ainsi sans la contribution et l'aide de nombreuses personnes que j'ai le plaisir et l'honneur de côtoyer, et qui m'ont aidé à acquérir une précieuse expérience et à atteindre mes objectifs.

Je tiens à exprimer tout d'abord mes profonds remerciements et reconnaissance à mon directeur de thèse Monsieur **BELMAJDOUB Fouad** pour la confiance qu'ils m'a accordé en m'acceptant à intégrer son équipe de recherche. Je souhaite le remercier pour son encadrement de très haut niveau, ses précieux conseils et orientations que j'ai toujours trouvées pertinents, pour sa disponibilité et son dévouement. Je tiens à exprimer ma profonde gratitude et remerciement pour le temps que vous avez consacré à superviser ma thèse. Ce fut un véritable plaisir de travailler sous votre direction. Je remercie également Monsieur **BENNANI Younes** de m'avoir encadré durant mon séjour à LaMSN - Université Sorbonne Paris Nord.

Je souhaite remercier mes professeurs Monsieur **EL HAMMOUMI Mohammed**, directeur Du laboratoire « Techniques industrielles », et Monsieur **EN-NADI Abdelali**, chef de département Génie industriel, pour l'ensemble des aides administratif, des conseils précieux et des ressources matérielles qu'ont mises à notre disposition.

Mes remerciements se dirigent également aux membres de jury qui ont accepté de sacrifier de leurs précieux temps et de leurs efforts, malgré leurs charges et leurs responsabilités, pour évaluer notre travaille. J'adresse mes remerciements aux :

- Professeur **HAMEDI L'habib**
- Professeur **DEROUICH Aziz**
- Professeur **MOUHSEN Ahmed**
- Professeur **SOULHI Aziz**
- Professeur **EN-NADI Abdelali**

Je n'oublierai pas de remercier tous mes ami(e)s et mes collègues enseignants, doctorants, personnels du laboratoire Techniques Industrielle à la faculté des sciences et techniques, je tiens à remercier tout particulièrement Messieurs **MSAAF Mohammed** et **FRI Mohammed**, merci pour votre amitié, vos

conseils et votre soutien permanent. Ainsi, je tiens à remercier tout le staff de La-MSN pour leur accueil chaleureux et pour l'atmosphère exceptionnelle et agréable que j'ai vécu durant mon séjour. Mes remerciements se dirigent aussi au Centre National pour la Recherche Scientifique et Technique pour le support financier.

Enfin, je remercie tout le corps enseignant et administratif de la FST de Fès pour leurs accueils et leurs aides, et aux membres de ma famille qui m'ont soutenu et encouragé tout au long de mon parcours.

Tables des Matières

Abréviations 12

Notations 13

Introduction Générale 15

Chapitre 1 : Formalismes et Méthodes du Diagnostic et de Pronostic des Systèmes à évènements discrets 18

1- Introduction 20

2- Le Concept d'un Système 20

3- Systèmes à Evènements Discret 21

4- Quelques Formalismes des SEDs 22

4-1- Modélisation des SEDs 22

4-2- Théorie du Langage 22

4-3- Méthode de Modélisation Qualitative 23

4-3-1- Automates Finies 23

4-3-2- Réseaux de Petri 24

4-4- Méthode de Modélisation Quantitative 25

4-4-1- Représentation d'Etat 25

4-4-2- Algèbre de Boole 25

4-4-3- La logique Temporelle 26

4-6- Structure de Prise de Décision 27

4-7- Diagnostic et Pronostic des SEDs 29

4-7-1- Terminologie du Diagnostic et Pronostic 29

4-7-2- Classification des Approches du Diagnostic et Pronostic des SEDs 30

4-7-3- Approches Basée sur des Modèles 30

A) Méthode pour le Diagnostic des SEDs 31

B) Méthodes pour le Pronostic des fautes dans les SEDs 34

4-6-4- Approches Basée sur la Connaissance 35

4-6-5- Approches Basée sur le Traitement de Données 35

5- Etude Comparative entre les Différentes Approches du Diagnostic et de Pronostic des SEDs 36

5-1- Critères de Comparaison 36

5-2- Forces et Faiblesses des Approches du Diagnostic et de Pronostic des SEDs 37

Chapitre 2 : Les Réseaux de Neurones 42

1- Introduction 44
2- Neurone Formel 44
3- Entraînement des Réseaux de Neurones 45
4- Algorithmes ou Optimiseurs d'Apprentissage 47
4-1- Notation 47
4-2- Règle de Hebb et Rosenblatt 47
4-2- La règle de Delta 48
4-3- Descente de gradient par rétropropagation 48
4-4- Variante de descente de gradient par rétropropagation 50
4-4-1- Descente de gradient par batch (BGD) 50
4-4-2- Descente de gradient stochastique (SGD) 50
4-4-3- Descente de gradient en mini-batch (MGD) 51
4-5- Algorithmes d'optimisation par Descente de Gradient 51
4-5-1- Descente de gradient avec Momentum 51
4-5-2- Gradient accéléré de Nesterov (NAG) 52
4-5-3- Rprop 52
4-5-4- ADAgrad 53
4-5-5- RMSprop 54
4-5-6- ADAM 54
4-5-7- LARS 55
5- Les Réseaux de Neurones Récurrents 56
5-1- Définition 56
5-2- Long Short-Term Memory Network (LSTM) 57
5-3- Gated Recurrent Unit (GRU) 57
6- Les Applications autour des Réseaux de Neurones 57
6-1- La classification 57
6-2- La régression 58
6-3- Diagnostic et Pronostic des SEDs par les RdNs 59
7- Conclusion 60

Chapitre 3 : Diagnostic et Pronostic des SEDs par Réseaux de Neurones à Propagation Avant 61

1- Introduction 63
2- Classification des fautes dans les SEDs 63
3- Diagnostic et pronostic des Fautes à l'aide des RdNs feed-foorward 64
4- Diagnostic et Pronostic des Fautes 67
4-1- Diagnosticabilité 67

4-2- Pronosticabilité ... 69
4-3- Temps Restant avant une Faute ... 71
5- Construction des Réseaux de Neurones et Etude de Cas ... 72
5-1- L'Architectures du Réseau de Neurone ... 72
5-2- Apprentissage des Réseaux de Neurones ... 73
5-3- Ensemble de Données du Diagnostic ... 73
5-4- Ensemble de Données du Pronostic ... 74
5-5- Ensemble de Données du Temps Restant avant une Faute ... 74
5-6- Algorithme du Diagnostic et du Pronostic des SEDs par paire temporelle ... 74
5-7- Etude de Cas ... 76
5-6-1- Présentation du SED ... 76
5-6-2- Analyses des Modèles RBFNNs ... 77
6-Conclusion ... 83
Chapitre 4 : Diagnostic et Pronostic des SEDs par Réseaux de Neurones Récurrent ... 84
1- Introduction ... 86
2- Diagnostic et Pronostic des Fautes par les Réseaux de Neurones Récurrents ... 86
2-1- Diagnosticabilité ... 88
2-2- Pronosticabilité ... 90
2-3- Temps restant avant une Faute ... 92
2-4- Les Dimensions Effectives du Graph Partiel ... 92
3- Construction des Réseaux de Neurones Récurrent et Etude de cas ... 93
3-1- Ensemble de Données du Diagnostic ... 94
3-2- Ensemble de Données de Pronostic ... 94
3-3- Ensemble de Données du Temps Restant avant une Faute ... 94
3-4- Algorithme du Diagnostic et du Pronostic des SED par paire temporelle ... 94
3-4- Etude de Cas ... 95
3-4-1- La Construction et l'Evaluation des Réseaux de Neurones Récurrents ... 96
4- Conclusion ... 100
Chapitre 5 : Diagnostic des SEDs en se Basant sur les Contraintes Temporelles ... 102
1- Introduction ... 104
2- Contraintes Temporelles ... 104
3-La Période comme Contrainte Temporelles ... 106
3-1- Signatures Temporelles Causales ... 106
3-2- Approche Proposé ... 107
3-3- Vérification des contraintes temporelles pour les SEDs ... 108

3-4- L'extraction des Contraintes Temporelles à Partir du Modèle Statistique 109
3-5- La Perte d'Observabilité et le Blocage du Système 109
3-6- Construction des Réseaux de Neurones 110
3-7- Base de Données des Contraintes Temporelles 110
3-4- Algorithme du Diagnostic des SED par Contraintes temporelles. 111
3-8- Etude de Cas 112
4- Conclusion 116
Conclusion Général et Perspectives 117
Références 120

Liste des Figures

Figure 1 : L'évolution d'une variable d'état pour les Système continus (a), à Evènements Discrets (b) et Hybrides (c) 21

Figure 2: Un SED modélisé par un automate finie déterministe. 24

Figure 3: Un SED modélisé par RdP. 25

Figure 4: Structure Centralisé 27

Figure 5: Structure Décentralisée 28

Figure 6: Structure Distribuée 29

Figure 7: Méthodes du Diagnostic et de Pronostic des SEDs 30

Figure 8: Les deux types des modèles du Diagnostic et de Pronostic 31

Figure 9: Un SED modélisé par automate fini et son diagnostiqueur à base d'événement 32

Figure 10 : Un SED modélisé par automate fini et son diagnostiqueur à base d'événement 33

Figure 11 : Neurone Formel 44

Figure 12 : Les fonctions d'activation les plus fréquemment utilisées, a) fonction linéaire b) fonction de seuil c) fonction linéaire par morceaux d) fonction sigmoïde e) fonction tangente hyperbolique f) fonction gaussienne 45

Figure 13: Les différentes étapes de l'algorithme d'apprentissage 46

Figure 14: Perceptron simple, modèle de Rosenblatt 47

Figure 15: Perceptron multi couche 49

Figure 16: Propagation de l'information dans un couche d'RNN 56

Figure 17 : Une cellule LSTM 57

Figure 18: Exemple d'une régression linéaire et régression non-linéaire 58

Figure 19: Exploitation des RdNs pour le Diagnostic et le Pronostic des SEDs 'Etape 1' 59

Figure 20: Exploitation des RdNs pour le Diagnostic et le Pronostic des SEDs 'Etape 2' 60

Figure 21: Diagnostic et pronostic des défauts à l'aide des RdNs feed-foorward 65

Figure 22: L'architecture du RBFNN 73

Figure 23: Algorithme du Diagnostic et du Pronostic des SED par paire temporelle 75

Figure 24: Le Chariot suspendu 76

Figure 25 : Performances d'apprentissage des RBFNNs du Diagnostic, du Pronostic et du Temps Restant avant une Faute 79

Figure 26: Performance de Test des RBFs du Diagnostic, Pronostic et de du Temps Restant avant une Faute 80

Figure 27: Résultats du Pronostic 81

Figure 28: Résultat de Temps restant avant une faute 82

Figure 29: Résultats du Diagnostic 83

Figure 30: Diagnostic et Pronostic des Fautes par RNNs 86

Figure 31: Algorithme d'Augmentation de l'Efficacité du Diagnosticabilité et Pronosticabilité 93

Figure 32: Algorithme du Diagnostic et du Pronostic des SED par graph partiel 95

Figure 33: La variation du EQM d'apprentissage et de validation de modèle du diagnostic de F1(a), de F2(b), de F3(c), de pronostic de F1(d), de F2(e), de F3(f), de calcule du temps minimal restant avant F1(g), F2(h), F3(i) et de calcule du temps maximal restant avant F1(j), F2(k), F3(l) 97

Figure 34: Pronostic (a) et Diagnostic (b) de F1 99

Figure 35: Pronostic (a) et Diagnostic (b) de F2 99

Figure 36: Pronostic (a) et Diagnostic (b) de F3 100

Figure 37: Temps restant avant les fautes 100

Figure 38: TC basée sur la Date 105

Figure 39: TC Basée sur la Période 105

Figure 40: Approche Proposé 107

Figure 41: GRAFCET de Fonctionnement d'un SED 108

Figure 42: Vérification des TCs par RBFNN 110

Figure 43: Algorithme du Diagnostic et du Pronostic des SED par graph partiel 112

Figure 44: Performances d'apprentissage du RBF de la Vérifications des TCs 114

Figure 45: Performance de Test du RBF de la Vérification des TCs 114

Figure 46: Résultats du Diagnostic du SED en se Basant sur la Contrainte Temporelle de la Période 116

Liste des Tableaux

Tableau 1 : Forces et faiblesses des approches du diagnostic et de pronostic des SEDs 38
Tableau 2 : Avantages et Inconvénients de la Règle de Hebb et Rosenblatt 48
Tableau 3 : Avantages et Inconvénients de la Règle de Delta 48
Tableau 4: Avantages et inconvénients de BGD 50
Tableau 5 : Avantages et inconvénients de SGD 51
Tableau 6 : Avantages et inconvénients de MGD 51
Tableau 7: Avantages et Inconvénients du Descente de Gradient avec Momentum 52
Tableau 8 : Avantages et Inconvénients du NGD 52
Tableau 9 : Avantages et Inconvénients du Rprop 53
Tableau 10 : Avantages et Inconvénients d'ADAgrad 53
Tableau 11 : Avantages et Inconvénients d'RMSProp 54
Tableau 12 : Avantages et Inconvénients d'Adam 55
Tableau 13 : Avantages et Inconvénients du LARS 55
Tableau 14 : Les Paramètres de RBFNNs 78
Tableau 15 : Désignation des Etat du chariot suspendu 96
Tableau 16 : Résultats de l'Application de l'Algorithme 1 sur le Chariot Suspendue 96
Tableau 17 : Les Classes du nRMSE 98
Tableau 18 : la RMSE et la nRMSE des RNNs sur les Données de Test. 98
Tableau 19 : Désignation des Événements de Référence et des fautes relatives à des violation des contraintes temporelles 113
Tableau 20 : Paramètres du RBFNN pour la vérification des TCs 113

Abréviations

AF : Automate Fini

BGD : Descente de Gradient par Batch

EQM : Erreur Quadratique Moyenne

FA : Automate Floue

GRU : Gated Recurrent Unit - Unité Récurrente Fermée

IoT : Internet des Objets

SED : Systèmes à Evénements Discrets

LSTM : Long Short Term Memory - Longue Mémoire à Court Terme

MGD : Descente de Gradient en Mini-batch

NAG : Gradient Accéléré de Nesterov

nRMSE: Erreur quadratique moyenne normalisée.

RBFNN : Réseaux de Neurones à Fonction de Base Radiale

RdP : Réseaux de Petri

RNN : Réseaux de Neurone Récurrent

RMSE : Racine de l'Erreur Quadratique Moyenne

SGD : Descente de Gradient Stochastique

Notations

$\sum$ **:** L'ensemble des Evénements Générés par le SED ;

$\sum_o$ **:** L'ensemble des Evénements Observables Générés par le SED ;

$\sum_{uo}$ **:** L'ensemble des Evénements Non-observables Générés par le SED ;

$\sum_f$ **:** L'ensemble des Evénements de Faute Générés par le SED ;

$\sum_r$ **:** L'ensemble des Evénements Régulier générés par le SED ;

e_i **:** Événement Observable ;

f_α **:** Événement de Faute ;

F_α **:** Un Etat de Faute ;

r_m **:** Événement Régulier ;

x_i **:** Un Etat du SED ;

X **:** L'espace d'Etat ;

X_N **:** L'ensemble des Etats Normaux ;

X_F **:** L'ensemble des Etats de Faute ;

TW^*: L'ensemble des mots temporisés qui peuvent être générés par un SED ;

σ **:** Un mot temporisé fini ;

t_i **:** Date d'occurrence d'un Evénement ;

ε : Mot vide ou Evènement Vide ;

s **:** Chaîne de Caractères ;

L : Un langage ;

$\sum^*$ **:** Kleene-closure ;

$\mu_A(x)$ **:** Coefficient d'appartenance d'une variable x a un ensemble A ;

$\sigma_{j'}^{j}$ **:** $j'^{ème}$ cycle d'un SED fonctionnant selon le $j^{ème}$ cycle de fonctionnement ;

TW^{j*} L'ensemble de toutes les données historiques, qui concernent le cycle de fonctionnement j ;

Tp : Une paire temporelle ;

$P_o(Tp)$: La projection, qui élimine le temps et les événements non observables d'une paire temporelle ;

$P_u(Tp)$**:** la projection qui ne conserve que les événements non observables d'une paire temporelle ;

$T(Tp)$**:** La projection, qui donne la date d'occurrence de Tp ;

$S_{j'}^{j}$**:** L'ensemble de toutes les paires temporelles dérivées de $\sigma_{j'}^{j}$;

$\boldsymbol{S^j_{j'f_\alpha}(Tp)}$: L'ensemble de toutes les paires temporelles dérivées de $\sigma^j_{j'}$ et ont la même projection que Tp ainsi que contenir un événement de faute $f_\alpha \in \sum_f$;

$\boldsymbol{S^j_{j'}(Tp)}$: L'ensemble de tous les couples temporels dérivés d'un $\sigma^j_{j'}$, qui ont la même projection que Tp ;

$\boldsymbol{S^j_{j'.r}}$ **:** L'ensemble de toutes les paires temporelles dérivées d'un cycle $\sigma^j_{j'} \in TW^{j*}$ et appartenant au même sous-mot temporisé (r) ;

$\boldsymbol{r}$: Indice de la phase de fonctionnement du SED ;

$\boldsymbol{P^j_r(f_\alpha/Tp)}$ **:** Diagnosticabilité de Tp par rapport à f_α ;

$\boldsymbol{Pg^j_r(f_\alpha/Tp)}$ **:** Pronosticabilité de Tp par rapport à f_α ;

$\boldsymbol{Suf^j_r(Tp/f_\alpha)}$ L'ensemble des suffixes de la première paire temporelle de Tp appartenant à un sous- mot temporel "r" ;

$\boldsymbol{t^j_r(f_\alpha/Tp)}$ **:** Le temps restant avant une faute pour Tp par rapport à f_α ;

$\boldsymbol{E^j}$: L'ensemble de tous les chemins partiels dérivés d'un σ^j ;

$\boldsymbol{E^j_{F_\alpha}}$: L'ensemble de tous les chemins partiels dérivés d'un σ^j, qui conduisent à un état de faute F_α ;

$\boldsymbol{E(SP)}$**:** L'ensemble des chemins d'échantillonnage partiels qui ont la même projection P_o comme SP ;

$\boldsymbol{E^{-1}(SP)}$: L'ensemble des continuations possibles de SP ;

$\boldsymbol{P(F_\alpha|Sp)}$: Diagnosticabilité de $\boldsymbol{Sp}$ par rapport à $\boldsymbol{F_\alpha}$ **;**

$\boldsymbol{Pg(F_\alpha|Sp)}$: Pronosticabilité de $\boldsymbol{Sp}$ par rapport à $\boldsymbol{F_\alpha}$ **;**

$\boldsymbol{P_o(SP)}$: The projection that keeps only observable events;

$\boldsymbol{SP}$**:** Graph Partiel ;

$\overline{\boldsymbol{SP}}$: Le Prefix de $\boldsymbol{SP}$;

$\boldsymbol{SP^{-1}}$**:** La continuation d'un Graphe Partiel $\boldsymbol{SP}$;

$\boldsymbol{TG}^*$: L'ensemble de tous les Graphes qui peuvent être formés dans un SED ;

$\boldsymbol{TC}$ **:** Contrainte Temporelle ;

STC : Signature Temporelle Causale ;

$\boldsymbol{ref(e_i)}$ **:** Evénement de référence de e_i ;

Introduction Générale

Le monde industriel ne cesse de se développer, ce qui crée une importante compétitivité sur le marché et pousse les industries à une exploitation optimale de leurs moyens, non seulement des équipements, mais aussi des ressources humaines [1]. Dans ce contexte, l'efficacité de la fonction de maintenance des systèmes industriels est devenue l'un des grands défis à relever pour qu'elle ne soit plus considérée comme un poste de dépense coûteux. Au contraire, elle est maintenant identifiée comme une fonction majeure et génératrice de profits [2], de sorte que le fait de maintenir les équipements dans leur état optimal pendant la production est devenu un point fondamental du succès du produit et de l'entreprise.

Durant les phases de conception, de développement et de mise en œuvre des systèmes industriels, divers types de vérifications et de tests sont effectués afin de garantir autant que possible leur fonctionnement sûr [3]. Cependant, malgré toutes les mesures qui peuvent être prises, il reste toujours sous le risque d'un comportement inattendu qui peut compromettre l'intégrité de ces systèmes [4]. Cela est lié à leur complexité qui ne cesse de croître, ainsi qu'à leur environnement opérationnel [4], [5]. L'apparition de tels agissements peut avoir des répercussions qui peuvent être majeurs et graves, allant de l'indisponibilité du système à des dommages humains, matériels, et bien sûr économiques [6], [7]. Par exemple, plus la phase de maintenance est longue, plus elle est coûteuse en raison de l'indisponibilité du système et parfois de l'endommagement du matériel [8]. Par conséquent, pour faire face à ces problèmes, les actions visant à améliorer la maintenance, dans laquelle la surveillance et ses composants principales "le diagnostic" et le "le pronostic" , qui sont des fonctions nécessaires pour assurer la disponibilité, la fiabilité et la sécurité de l'équipement, sont fondamentales et essentielles, d'autant plus que l'environnement technologique est de plus en plus complexe, les entreprises doivent soit externaliser les tâches de diagnostic et de pronostic pour mieux se concentrer sur leur activité principale et leurs cœurs métier, soit de passer du diagnostic et du pronostic traditionnels aux techniques à distance et en temps réel [9], [10], en plus que l'Internet des objets, qui est une composante principale de l'industrie intelligente, est en augmentation continue de sorte que le nombre de dispositifs connectés devrait passer de 700 millions de dispositifs à 75.44 milliards d'ici à 2025. Si plusieurs facteurs contribuent à cette croissance, l'un des plus importants est le développement des réseaux 5G, qui permettent une augmentation significative des vitesses de transfert de données et une plus grande couverture qui permettront aux appareils d'envoyer et de recevoir des données et des notifications plus rapidement que jamais [11]. Dans ce contexte, il existe un intérêt croissant pour l'utilisation de technologies intelligentes et de techniques plus innovantes et sophistiquées pour garantir le fonctionnement du diagnostic et de pronostic des systèmes industriels afin de profiter de ces améliorations.

Dans cette thèse, nous nous concentrerons sur les systèmes à événements discrets (SED), qui a un large éventail d'applications dans une variété d'industries (Les systèmes manufacturiers, Les systèmes de gestion de

l'information, Les réseaux de transport, etc) ; en outre, la majorité des systèmes continus (à un certain niveau d'abstraction) peuvent être considérés comme des SEDs, où la dynamique peut être considérée et représentée par des entrées et des sorties discrets. En générale un SED est un système avec un espace d'état discret. La transition entre un état à un autre se produit suite à l'occurrence d'événements asynchrones [12], [13], [14], [15], les comportements défectueux dans un SED se produisent à la suite de l'occurrence d'un événement particulier désigné comme une faute. Selon Ammor et al. [4], une faute est un événement dont l'occurrence est aléatoire et qui provoque une déviation du comportement nominal et normal vers un comportement défectueux. Dans ce contexte, le diagnostic des SEDs se concentre sur la détection des causes primaires des défaillances, c'est-à-dire les événements de faute basés sur une série de mesures [16]. Le pronostic des SEDs s'intéresse à la prédiction des défaillances avant qu'elles ne se produisent [17] (Une présentation et des définitions beaucoup plus détaillées sur le diagnostic et le pronostic des SEDs seront présentés dans la suite de cette thèse). A cet effet, les méthodes basées sur les modèles sont souvent utilisées, notamment les automates finis (AF) [18], [19], [20], les réseaux de Petri (RdP) [21], [22], [23], [24], ou leurs extensions [4], [25], [26], [27], [28], [29], [30], [31], [32], [33] mais l'utilisation de telles approches est généralement confrontée à divers écueils et difficultés, à savoir le développement de modèles et la difficulté de mise en œuvre. De plus, les systèmes sont généralement soumis à une reconfiguration et une adaptation permanente à leur environnement [5]. Dans ce sens, l'utilisation de techniques d'intelligence artificielle fait l'objet de nombreuses recherches. La possibilité d'effectuer une surveillance indépendamment des outils de modélisation, c'est-à-dire sans modéliser le système et en s'appuyant sur une base de données qui peut être constituée soit à partir de l'historique de fonctionnement du système, soit à partir de l'expertise, devient un intérêt évident pour les industriels. D'autant plus que le monde se dirige explicitement vers la maintenance intelligente ou la maintenance dite 4.0 au lieu d'une maintenance classique [2] et que la non-utilisation des données collectées est considérée comme une source de perte et de gestion inefficace des données.

L'objectif de nos recherches, c'est l'exploitation des réseaux de neurones artificiels (RdN), qui ont une grande capacité d'apprentissage et un haut niveau d'aptitude à s'adapter à des environnements dynamiques [9], afin d'assurer le diagnostic et le pronostic des SEDs. Dans cet effet, ce rapport sera organisé en cinq chapitres que nous pouvons les résumés comme suit :

- Le premier chapitre sera dédié pour une présentation d'un aperçu général sur les différentes classes des systèmes dynamiques, en se focalisant principalement sur la catégorie des SEDs, ainsi que quelques formalismes en relation avec ces systèmes, suivie par les différentes définitions et notions de base concernant les SEDs, ainsi que quelques outils utilisés pour leur modélisation. Ensuite, nous présenterons les différentes approches élaborées dans la littérature pour aborder les problèmes de diagnostic et de pronostic des SEDs. Eventuellement, la dernière section sera consacrée à une comparaison des différentes méthodes afin de justifier le choix de l'approche pour accomplir les fonctions du diagnostic et de pronostic.

- Dans le deuxième chapitre, un aperçu général concernant les RdNs, leurs modes de fonctionnement, algorithmes d'apprentissage, leurs types, à savoir les RdNs 'feed-forward' et les RdNs récurrents, ainsi que les structures des couches récurrentes, plus précisément la couche GRU et LSTM, seront présentés.

- Le troisième chapitre présente une nouvelle approche de diagnostic et de pronostic des fautes dans les SEDs basée sur les RdNs de types 'feed-forward' et son cadre théorique ainsi que les conditions suffisantes et nécessaires pour la diagnosticabilité et la pronosticabilité sont présentés. Une base de données d'apprentissage pour former les RdNs afin d'obtenir les opérations souhaitées est construite.

- Le quatrième chapitre s'inscrit dans le même objectif que le précèdent, de telle sorte qu'un type particulier des RdNs, à savoir les RdNs récurrents qui sont dédiés au traitement de données séquentielles, seront appliqué pour assurer le diagnostic et le pronostic des SEDs.

- Le cinquième chapitre est considéré comme une extension des deux derniers chapitres, dans ce dernier, l'aspect temporelle des SEDs est intégrée afin de proposer une nouvelle approche du diagnostic des SED qui contrôle et vérifie en temps réel tout comportement anormal ou déviation associé à une violation d'une contrainte temporelle pour chaque événement généré par le SED.

Et afin de déterminer et de prouver la pertinence des approches proposée, une étude de cas sur un SED réel est présentée à la fin de chacun des trois derniers chapitres.

Chapitre 1 : Formalismes et Méthodes du Diagnostic et de Pronostic des Systèmes à évènements discrets

Table des Matières

1- Introduction 20
2- Le Concept d'un Système 20
3- Systèmes à Evènements Discret 21
4- Quelques Formalismes des SEDs 22
4-1- Modélisation des SEDs 22
4-2- Théorie du Langage 22
4-3- Méthode de Modélisation Qualitative 23
4-3-1- Automates Finies 23
4-3-2- Réseaux de Petri 24
4-4- Méthode de Modélisation Quantitative 25
4-4-1- Représentation d'Etat 25
4-4-2- Algèbre de Boole 25
4-4-3- La logique Temporelle 26
4-6- Structure de Prise de Décision 27
4-7- Diagnostic et Pronostic des SEDs 29
4-7-1- Terminologie du Diagnostic et Pronostic 29
4-7-2- Classification des Approches du Diagnostic et Pronostic des SEDs 30
4-7-3- Approches Basée sur des Modèles 30
A) Méthode pour le Diagnostic des SEDs 31
B) Méthodes pour le Pronostic des fautes dans les SEDs 34
4-6-4- Approches Basée sur la Connaissance 35
4-6-5- Approches Basée sur le Traitement de Données 35
5- Etude Comparative entre les Différentes Approches du Diagnostic et de Pronostic des SEDs 36
5-1- Critères de Comparaison 36
5-2- Forces et Faiblesses des Approches du Diagnostic et de Pronostic des SEDs 37

1- Introduction

L'efficacité de la fonction de maintenance des systèmes industriels est l'un des principaux défis économiques pour leur exploitation commerciale. Les principales difficultés et sources d'inefficacité sont liées au choix des actions de maintenance appropriées. Par ailleurs, pour répondre aux exigences de compétitivité, les entreprises doivent exploiter leurs outils de production en prenant en compte la sécurité des hommes et des équipements, ainsi que la qualité du produit. Dans ce contexte, le rôle de la fonction maintenance prend une dimension encore plus importante. Tout système est censé être confronté à différents types de fautes, qui peuvent entraîner un changement radical du comportement normal du processus ainsi que la dégradation de ses performances, de telle sorte qu'il ne puisse plus remplir ses fonctions. Pour faire face à ce type de problèmes, l'opération du diagnostic et du pronostic sont considérées comme une phase très importante de la maintenance, de sorte que plus le diagnostic et le pronostic sont efficaces, plus les opérations et les interventions de maintenance sont percutantes. Par conséquent, un module de diagnostic (diagnostiqueur) et de pronostic (pronostiqueur) sont nécessaires pour éviter la propagation des pannes et limiter leurs conséquences, qui peuvent être catastrophiques à tous les niveaux : Disponibilité, fiabilité et sécurité des équipements.

2- Le Concept d'un Système

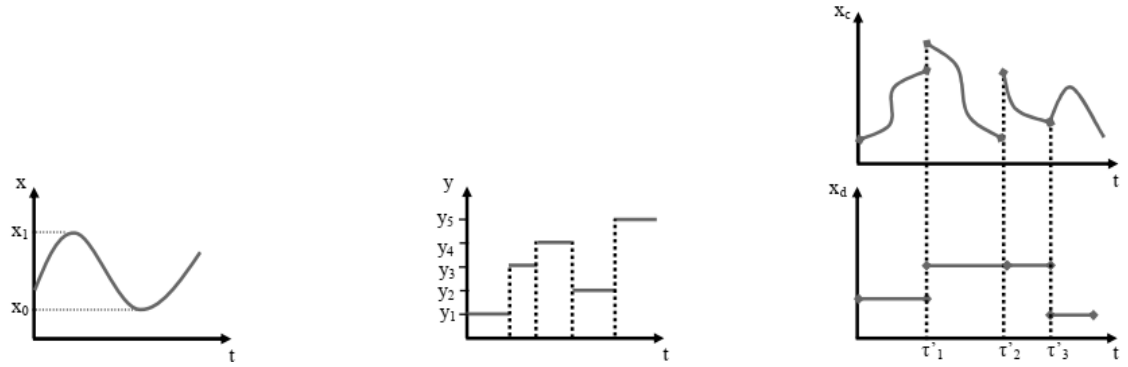
Selon le dictionnaire standard des termes électriques et électroniques de l'IEEE, un système est « *Une combinaison de composants qui agissent ensemble pour réaliser une fonction impossible à réaliser avec aucune des parties individuelles* » [13], en d'autres termes, un système est constitué d'un ensemble des composants en interaction entre eux et qui sont associés à une fonction qu'ils sont censés remplir afin d'assurer des tâches bien déterminer dans le but d'atteindre un objectif bien précis. La nature de leurs variables d'état distingue particulièrement ces systèmes. Par conséquent, nous pouvons identifier [14]:

- **Les Systèmes Continues :** Ce sont des systèmes où toutes les variables d'état sont des variables continues (le débit d'une pompe, la vibration d'une machine rotative, le niveau de liquide dans un bassin, etc.), l'évolution dynamique de ces systèmes par rapport au temps est généralement modélisée par des modèles mathématiques dont les équations de transfert et les espaces d'états.

- **Les Systèmes à Evénements Discret :** Ces systèmes, et contrairement aux systèmes continus, sont des systèmes dynamiques dont les signaux changent de valeur d'une façon non-continue, c'est-à-dire, qu'un SED est un système dynamique, dans lequel l'espace d'état est un ensemble discret et l'état ne change qu'à un moment donné dans le temps.

- **Les Systèmes Hybride :** Ce sont est des systèmes dynamiques qui présentent un comportement à la fois continu et discret, c'est-à-dire que ces systèmes peuvent contenir à la fois des variables événementielles (Variables d'états discrets) et continues (Variables d'états continus) simultanément.

La figure suivante (Figure 1) illustre la différence des trajectoires d'évolution entre les systèmes dynamiques à variables continues, les SEDs et les systèmes hybrides [14].



a) Système continus b) Systèmes à Evènements Discrets c) Systèmes Hybrides

Figure 1 : L'évolution d'une variable d'état pour les Système continus (a), à Evènements Discrets (b) et Hybrides (c)

3- Systèmes à Evènements Discret

Selon Cassandrass et al. [13], un SED est un système dynamique, où l'espace d'état est un ensemble discret et l'état ne change qu'à un moment donné, de telle sorte que l'évolution et la progression des états sont lancées par l'occurrence d'un événement ponctuel spécifique (arrivée d'un signal, début d'une tâche, achèvement d'une tâche, etc.), c'est-à-dire que les transitions entre un état à un autre se produisent suite aux occurrences des événements asynchrones. Le mot "discret" ne fait pas référence à état discret ou à temps discret, mais que la dynamique est composée d'événements, qui peuvent être le début et la fin d'évolutions continues. Par conséquent, un SED peut être considéré comme un générateur d'événements $\sum$ qui peuvent être observables ou inobservables [34], [35]. Donc, $\sum$ peuvent être divisés en deux ensembles principaux $\sum_o$ et $\sum_{uo}$ de telle sorte que $\sum = \sum_o \cup \sum_{uo}$, tel que $\sum_o$=$\{e_1, e_2 \dots, e_i, \dots\ e_N\}$ représente l'ensemble des événements observables, dont l'occurrence peut être visualisée, enregistrée et analysée, et $\sum_{uo} = \sum_f \cup \sum_r$ est l'ensemble des événements non observables, où $\sum_f$= $\{f_1, f_2, \dots, f_\alpha, , \dots, f_p\}$ représente l'ensemble des fautes ou événements de faute que le SED pourrait rencontrer et $\sum_r$= $\{r_1, r_2, \dots, r_m\}$ est l'ensemble des événements réguliers. Chaque événement généré par le SED conduit à un état spécifique x_i qui appartient à l'espace d'état X, avec $X = \{x_1, x_2, \dots, x_n\}$ et qui peut référer à un état normal ou à un état de faute, de sorte que chaque état de faute F_α est atteint par l'occurrence de l'événement de faute f_α. Par conséquent, l'ensemble X peut également être divisé en deux ensembles principaux, de sorte que $X = X_N \cup X_F$, avec X_N représentent l'ensemble de tous les états correspondant au fonctionnement normal, et X_F désigne l'ensemble de tous les modes de défaillance, tels que $X_F = \{F_1, F_2, \dots, F_\alpha, \dots, F_P\}$.

En général, un SED fonctionne par cycles appelés cycles de fonctionnement et chacun d'eux traite différentes tâches pour atteindre un état prédéterminé. Chaque cycle de fonctionnement est noté par "σ" et il est composé par une séquence d'événements, qui peuvent être observables ou inobservables, formant un mot temporisé fini (on note l'ensemble de tous les mots temporisés qui peuvent être générés par un SED par TW^*), de manière

que chaque événement soit indexé par sa date d'occurrence, par exemple $\sigma = e_1^{t_1} e_2^{t_2} \dots e_i^{t_i} \dots e_k^{t_k}$ est un mot temporisé fini tel que e_i est un événement appartenant à $\sum$ et t_i est sa date d'apparition [36], de sorte que $t_i \in \mathbb{R}^+$.

Il existe plusieurs approches pour représenter le temps dans un mot temporisé fini, la plus courante est celle présentée dans [26], [37]. Dans cette représentation, au lieu d'indexer les événements avec leurs dates d'occurrence, les auteurs proposent de coupler les événements avec le temps entre leur occurrence propre et l'occurrence de l'événement précédent. Dans ce cas, étant donné un $\sigma = e_1^{t'_1} e_2^{t'_2} \dots e_i^{t'_i} \dots e_k^{t'_k}$, $t'_i = t_i - t_{i-1}$.

4- Quelques Formalismes des SEDs

4-1- Modélisation des SEDs

La modélisation d'un système implique la description de sa structure ainsi qu'en une représentation fonctionnelle ou comportementale de chacun de ses composants. Comme la représentation comportementale repose sur des relations entre les différentes variables du système, une étape de modélisation est souvent nécessaire pour décrire le comportement normal ou aberrant du système [38]. Pour l'étude ainsi que l'analyse des SEDs, une variété d'approches de modélisation ont été présentées dans la littérature, qui peuvent être classés en deux principales catégories [39] :

- **Méthode Qualitative :** Permet de représenter le comportement ou le fonctionnement du système avec un certain degré d'abstraction en utilisant des modèles symboliques [39], les plus répondues sont les AFs et les RdPs.

- **Méthode Quantitative :** Permet de représenter le comportement du système à travers des modèles mathématiques et/ou structurels, parmi les plus utilisées, on trouve la représentation d'état, l'algèbre de Boole et la logique temporelle.

Par la suite, des définitions formelles de ces outils seront présentées, précédées par un aperçu général et de définitions de base de la théorie des langages, qui serviront de base à certains des formalismes que nous décrirons plus tard.

4-2- Théorie du Langage

Dans la théorie du langage, l'ensemble des événements $\sum$ d'un SED est considéré comme un alphabet. Une séquence d'événements tirés de cet alphabet forme un "mot" ou une "chaîne". Un mot ne comportant aucun événement est appelée mot vide ou évènement vide et il est désignée par ε. La longueur d'un mot est le nombre des événements qu'elle contient, en comptant les occurrences multiples du même événement. Si s est une chaîne de caractères, nous désignerons sa longueur par $|s|$.

Définition 1 : (Langage) Un langage défini sur un ensemble d'événements $\sum$ est un ensemble des mots de longueur finie formés à partir d'événements dans $\sum$. Un langage est généralement dénoté par L. À titre

d'exemple, nous supposons que $\sum = \{\alpha, \beta, \gamma, \mu\}$, le langage $L = \{\alpha, \beta, \alpha\beta, \alpha\beta\gamma\mu\beta, \alpha\beta\alpha\beta\}$ est un langage sur l'alphabet $\sum$ composé de cinq mots.

Définition 2 : (Kleene-closure) Noté par $\sum^*$, un « Kleene-closure » sur un ensemble d'événements $\sum$ est l'ensemble de tous les mots finis d'une longueur arbitraire, y compris l'évènement vide ε d'éléments de $\sum$. Soit $\sum$ le même ensemble que la définition précédente (Définition 1), le « Kleene-closure » sur $\sum$ est :

$$\sum^* = \{\varepsilon, \alpha, \beta, \gamma, \mu, \alpha\alpha, \alpha\beta, \alpha\,\gamma, \alpha\mu, \beta\alpha, \alpha\beta\gamma, \alpha\beta\mu, \beta\alpha\alpha, \beta\beta\beta, \dots\}$$

Définition 3 : (Concaténation) Soit s_a et s_b deux mots de l'ensemble des mots finis $\sum^*$, le mot $s = s_a s_b$ est la concaténation du mot s_a et le mot s_b, de telle sorte que s constituée des événements en s_a suivie directement par les événements de s_b. Considérons s_a, $s_b \in \sum^*$ le même ensemble que la définition 1 tel que $s_a = \alpha\mu$ et $s_b = \alpha\beta\gamma$, la concaténation de s_a et s_b est le mot $s = \alpha\mu\alpha\beta\gamma$.

Remarque 1 : l'événement ε est considéré comme un élément identité de la concaténation ($s = \varepsilon s = s\varepsilon$).

Définition 4 : (Préfixe, sous-mot et Suffixe) Soit s un mot de $\sum^*$ composé par la concaténation de trois mots tel que $s = s_a\, s_b\, s_c$, s_a, s_b, s_c sont respectivement le préfixe, le sous-mot et le suffixe de s. Par convention, ε et s sont tous deux à la fois des préfixes, des sous-chaînes et des suffixes de s.

4-3- Méthode de Modélisation Qualitative

4-3-1- Automates Finies

Dans la littérature, un AF est défini comme un outil adapté à la modélisation de l'information sous forme d'événements. Un AF est représenté par une succession d'états et de transitions dotée par des événements [13], [40]. En général, un AF déterministe G est un 6-tuple :

$$\boldsymbol{G = (X, \sum, \delta, \tau, x_0, X_m)}$$

Où :

- $\boldsymbol{X}$ est l'ensemble des états du SED ;
- $\sum$ est l'ensemble fini des événements associés à G ;
- $\boldsymbol{\delta}: X\text{x}\sum \rightarrow X$ est la fonction de transition, $\delta(x_1, e) = x_2$ signifie qu'il y a une transition de l'état x_1 à l'état x_2 après l'occurrence de l'événement e ;
- $\boldsymbol{\tau}: \sum \rightarrow 2^{\Sigma}$ est l'ensemble des événements active ; pour un état $x \in X$, $\tau(x)$ est l'ensemble des événement pour lesquelles $\delta(x_1, e)$ est définie. Plus formellement :

$$\tau(x) = \{e \in \sum : \exists x' \in X, \delta(x, e) = x'\};$$

- $\boldsymbol{x_0}$ est l'état initial ;
- $\boldsymbol{X_m}$ est l'ensemble des états marqué ou finie, marqués par deux cercles dans le modèle.

L'automate G fonctionne comme suit : à partir d'un état initial $x_0 \in X$ et lors de l'occurrence d'un événement e ∈ ∑, si $e \in \tau(x_0)$, c'est-à-dire que la fonction $\delta(x_0, e)$ est définie, l'automate fait une transition vers l'état $x' = \delta(x_0, e)$. Ce processus est ensuite répété pour chaque événement généré, menant l'automate de l'état initial à un état final. Un automate est dit déterministe si et seulement s'il dispose d'un état initial unique et qu'il ne peut y avoir deux transitions avec le même événement $e \in \sum$ sortant d'un état $x \in X$. En revanche, pour un automate non déterministe, il peut y avoir plusieurs états initiaux et plusieurs transitions avec le même évènement sortant du même état. La figure suivante (Figure 2) représente le comportement d'un SED modéliser par un AF déterministe tel que : $X = \{x, y, z\}$, $\sum = \{a, b, g\}$, $x_0 = x$ et $X_m = \{x, z\}$.

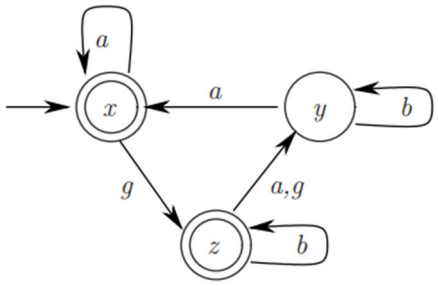


Figure 2: Un SED modélisé par un automate finie déterministe.

4-3-2- Réseaux de Petri

Un Réseaux de Petri est un graphe orienté biparti, c'est-à-dire ayant deux types de nœuds (places et transitions) reliés par des arcs orientés (tout arc doit obligatoirement relier deux nœuds de type différent) [41]., Un RdP est un 5-tuplet $(\boldsymbol{P}, \boldsymbol{T}, \boldsymbol{Pre}, \boldsymbol{Post}, \boldsymbol{M_0})$ Où :

- $\boldsymbol{P}$ est l'ensemble des places ;
- $\boldsymbol{T}$ est l'ensemble des transitions ;
- $\boldsymbol{Pre}$: $\boldsymbol{P}$ x $\boldsymbol{T} \rightarrow$ *{0.1}* est l'application de l'incidence avant ;
- $\boldsymbol{Post}$: $\boldsymbol{P}$ x $\boldsymbol{T} \rightarrow$ *{0.1}* est l'application d'incidence arrière ;
- $\boldsymbol{M_0}$: $\boldsymbol{P} \rightarrow \mathbb{N}$ est le marquage initial, où, pour chaque place $p \in$ P , un nombre positif $n \in \mathbb{N}$ de jetons est associés.

La figure suivante (figure 3) représente le comportement d'un SED modéliser par un réseau de Petri tel que $\boldsymbol{P} = \{P_1, P_2, P_3, P_4\}$, $T = \{T_1, T_2\}$ et $M_0 = (1,3,0,0)^T$

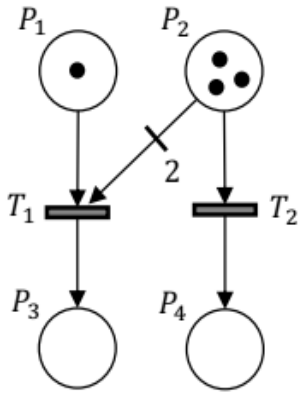


Figure 3: Un SED modélisé par RdP.

4-4- Méthode de Modélisation Quantitative

4-4-1- Représentation d'Etat

L'approche de l'espace d'état est une méthode unifiée de modélisation, d'analyse et de conception d'un large éventail des systèmes. L'analyse et la conception des représentations d'états reposent principalement sur les matrices et les opérations matricielles, ils permettent d'utiliser des variables d'état pour représenter un système dynamique sous forme de matrice. Cette représentation, qui peut être linéaire ou non, variable dans le temps ou non, continue ou discret, permet de prédire l'état du système à tout moment dans le futur, en tenant compte de son état initial et du comportement des variables exogènes qui l'affectent [13]. Un système linéaire invariant et à temps continus est modélisé en utilisant la représentation d'état par les équations suivantes :

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases}$$

Tel que pour $t \geq t_0$, x, $\dot{x}$, y, u, A, B, C et D représentent respectivement le vecteur d'état, la dérivée du vecteur d'état par rapport au temps, le vecteur de sortie, le vecteur d'entrée ou de contrôle, la matrice du système, la matrice d'entre, la matrice de sortie et la matrice de feed-forward.

4-4-2- Algèbre de Boole

L'algèbre booléenne est une technique mathématique qui est utilisée pour décrire, analyser et construire des circuits logiques et numériques. Ces systèmes décrivent l'évolution de la commande d'un actionneur logique (tout ou rien) ou numérique en réponse à des commandes et instructions variées provenant d'un utilisateur ou de capteurs industriels. Une algèbre de Boole est constituée des éléments suivants :

- Des variables logiques qui correspondent à des entrées, des sorties ou des variables intermédiaires, chacune de ces variables est à tout moment soit égale à 0 (Faux) ou égale à 1 (Vrai).
- Des opérateurs logiques dont on trouve particulièrement l'addition logique ou le « OU » logique désigné par « + », la multiplication logique ou le « ET » logique désigné par «.» et l'inversion logique ou le « NON » logique symbolisé par « $\bar{A}$ ».

4-4-3- La logique Temporelle

La logique temporelle est une extension de la logique conventionnelle, elle permet d'intégrer des nouveaux opérateurs afin d'exprimer la notion du temps. On peut distinguer deux types principaux : la logique temporelle linéaire qui permet de spécifier le comportement attendu d'un système, en spécifiant l'unique futur possible, et la logique du temps arborescent qui permet de prendre en considération différents futurs possibles à partir d'un état du système au lieu d'avoir une vue linéaire du système considéré.

4-5- Les Opérations autour des SEDs

En pratique, différents types d'opérations sont envisageables autour des SEDs afin de garantir la sûreté du système ainsi que la sécurité des personnels, parmi les opérations on trouve :

- **La Maintenance :** La maintenance est l'ensemble des actions à la fois technique et administrative permettant de maintenir (maintenance préventive) ou de rétablir (maintenance corrective) un bien dans un état spécifié ou en mesure d'assurer un service déterminé.

- **La Surveillance :** C'est l'ensemble des opérations implémenter afin d'observer, suivre et surveiller l'état de fonctionnement du système en temps réel dans le but de déterminer l'état actuel du système, pour gérer correctement les risques probables qu'un système peut rencontrer pendant ses phases d'exploitation.

- **La Supervision :** Ce processus traite des opérations normales et anormales d'un système, son objectif principal est de calculer et de définir les paramètres de la séquence de commande à exécuter en fonction de l'état du système. Dans le cas du fonctionnement normal, le superviseur surveille et contrôle l'exécution d'une opération et le fonctionnement du système. Par contre, dans le cas du fonctionnement anormal, et en se basant sur les causes de la faute (Soit au niveau matériel ou logiciel), le superviseur s'engage à prendre les décisions nécessaires et les actions correctives afin de retourner le système à son fonctionnement normal.

- **Le Diagnostic :** Le diagnostic ou la localisation d'un défaut est le processus qui consiste à détecter le plus tôt possible une déviation du système de son fonctionnement normal et nominal vers un fonctionnement défectueux et à identifier ses causes et ses conséquences. L'opération de diagnostic nécessite trois phases importantes, La première phase est le processus de détection de l'occurrence de l'anomalie. La deuxième consiste à localiser cette anomalie et à identifier ses causes, éventuellement, et la troisième phase consiste à analyser les conséquences et l'impact de cette anomalie sur l'ensemble du système. Dans le contexte des SEDs, le diagnostic consiste à détecter l'occurrence des événements de faute $f_\alpha \in \sum_f$, et à préciser ses localisations par rapport aux événements observables générés par le SED.

- **La Diagnosticabilité :** La diagnosticabilité représente la capacité d'un système à être diagnostiqué. En d'autres termes, la diagnosticabilité est la capacité pour un système de surveillance à présenter des symptômes différents pour chaque situation de faute.

- **Le Pronostic :** le pronostic, la prédiction ou l'anticipation des défauts, est une opération qui vise à prédire le comportement futur probable et l'état de fonctionnement du système sur un intervalle de temps allant de l'instant où la prédiction est faite jusqu'à l'apparition de l'anomalie, qui peut faire passer le système d'un comportement normal à un comportement défectueux. Concernant le pronostic des SEDs, il consiste à prédire l'apparition des événements de faute $f_\alpha \in \sum_f$ et à déterminer le temps restant avant que ces événements n'apparaissent afin de prendre les actions nécessaires pour assurer le bon fonctionnement du système.

- **La Pronosticabilité :** La pronosticabilité représente la capacité d'un système à être pronostiqué. En d'autres termes, elle représente la capacité à prédire l'occurrence d'une faute.

Ces définitions montrent que les opérations de diagnostic et de pronostic sont des fonctions décisionnelles et opérationnelles pour assurer le fonctionnement régulier du système. Dans cette thèse, nous nous intéressons au diagnostic et au pronostic des SEDs.

4-6- Structure de Prise de Décision

Les opérations présentées dans la section précédente peuvent être implémentées aux SEDs selon différentes structures, nous citons dans le cas du Diagnostic et de Pronostic :

- **La Structure Centralisé :** Dans la structure centralisée, les indices du diagnostic et du pronostic sont calculés à l'aide d'un diagnostiqueur et pronostiqueur global (monolithique), qui est construit à partir du modèle global du système. Cette structure est décrite dans la figure 4. Les approches centralisées présentent un avantage majeur en termes de précision de diagnostic et de pronostic et de simplicité conceptuelle. Leur principal inconvénient réside dans leur complexité informatique prohibitive, qui nécessite un modèle d'usine centralisé pour générer le diagnostiqueur et le pronostiqueur centralisé.

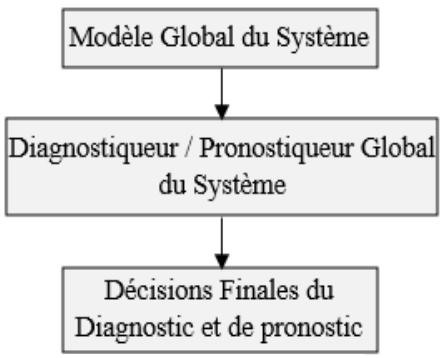


Figure 4: Structure Centralisé

- **La Structure décentralisée :** Dans la structure décentralisée (Figure 5), le système est divisé en un certain nombre de sites. Chaque site connaît le modèle complet du système, dispose d'observations locales et utilise un diagnostiqueur et pronostiqueur local qui calcule des décisions locales basées sur ses observations partielles du système complet. Chacun de ces diagnostiqueurs et de pronostiqueurs reçoivent des données particulières et sont invitées à faire un choix local sur la base de ses observations. Un coordinateur fournit la décision finale de diagnostic et de pronostic en se basant sur les décisions qui lui sont communiquées. Les diagnostiqueurs et les pronostiqueurs locaux et le coordinateur sont construits à partir d'un modèle global du système. Les diagnostiqueurs et les

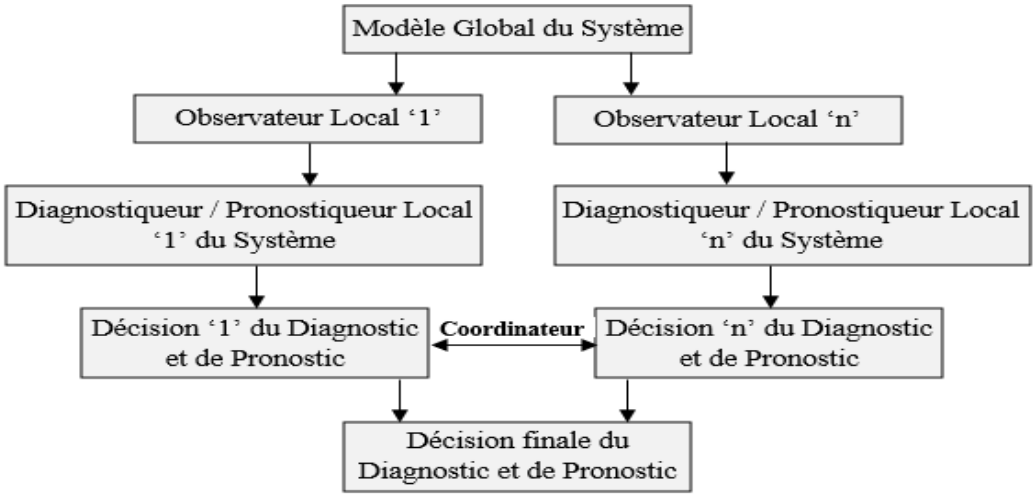


Figure 5: Structure Décentralisée

pronostiqueurs locaux ne peuvent pas communiquer directement entre eux, et généralement seule une communication limitée entre eux par le biais du coordinateur est autorisée. Le principal problème des architectures décentralisées est de savoir comment les sites découvre ensemble l'apparition d'une faute, sachant que les informations peuvent être erronées, incomplètes, différées. Le coordinateur doit donc disposer d'une certaine mémoire et de capacités de traitement pour coordonner l'échange d'informations nécessaire entre les diagnostiqueurs et les pronostiqueurs locaux afin de résoudre les ambiguïtés des décisions locales.

- **La Structure Distribuée :** Les approches distribuées réalisent le diagnostic et le pronostic en utilisant un ensemble de modèles locaux sans tenant compte d'un modèle global du système. Son principal objectif est l'amélioration de la robustesse et de l'évolutivité. Chaque module local ne connaît que sa propre partie du modèle global et possède son diagnostiqueur et son pronostiqueur local afin d'effectuer un diagnostic et un pronostic local. Les calculs sont basés sur le modèle local et les informations sont communiquées par les autres locaux à travers un protocole de communication. Les informations échangées entre les diagnostiqueurs et les pronostiqueurs locaux sont utilisés pour mettre à jour leurs propres informations et compensent leur observation partielle. Le défi du diagnostic et du pronostic distribué est de réaliser un diagnostic et un pronostic local équivalent, si possible, au

diagnostic et pronostic global, en utilisant un protocole de communication évolutif (par rapport au nombre de modules composants), et sans avoir besoin d'utiliser un modèle global.

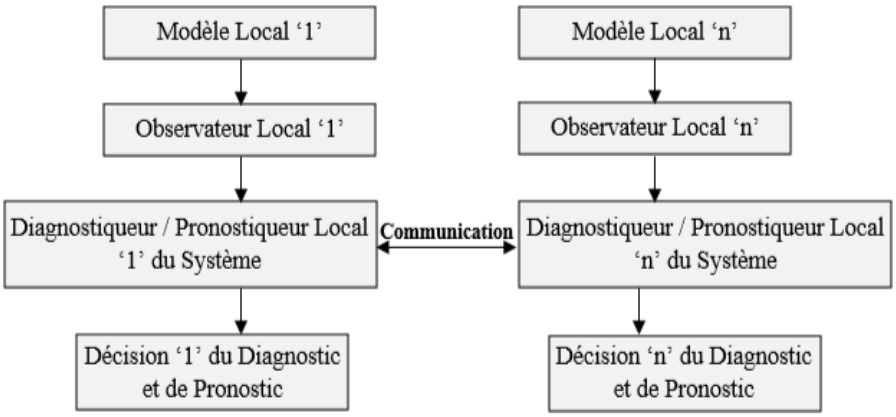


Figure 6: Structure Distribuée

4-7- Diagnostic et Pronostic des SEDs

4-7-1- Terminologie du Diagnostic et Pronostic

Avant d'aller plus loin dans ce mémoire de thèse, il est essentiel d'introduire une certaine terminologie nécessaire dans le domaine du Diagnostic et de Pronostic.

- **Mode de Fonctionnement :** Il décrit comment le système agit pour atteindre un certain objectif. Un système peut fonctionner selon plusieurs modes, notamment les modes Normal (Optimal), Aberrant (Fautif ou Anormal) et Détérioré (Dégradation).

- **Défaut ou Faute :** il est considéré comme une divergence par rapport au comportement habituel, il est interprété comme un événement qui perturbe le comportement ou le fonctionnement normal de système. Il se caractérise par un écart par rapport à la valeur caractéristique ou de propriété d'un paramètre du processus. Par conséquent, un défaut est une anomalie de comportement qui se produit avant une défaillance potentielle.

- **Défaillance :** La défaillance est définie comme une anomalie fonctionnelle qui empêche totalement ou partiellement un processus de remplir ses fonctions requises avec les performances définies dans les spécifications techniques. Une unité fonctionnelle est en état de panne après une défaillance.

- **Panne :** C'est un arrêt partiel ou total d'un processus après l'apparition d'une défaillance de sorte que le système est incapable d'assurer ses fonctions et ses tâches requise.

4-7-2- Classification des Approches du Diagnostic et Pronostic des SEDs

Il existe plusieurs techniques pour diagnostiquer et pronostiquer les fautes dans les SEDs, ils se distinguent sur différents critères : la nature de l'information, la structure de décision, l'implémentation des modèles de diagnostic ou de pronostic (implémentation en ligne ou hors-ligne), etc.

En général, ces méthodes sont divisées en deux catégories : Des méthodes basées sur l'existence d'un modèle qui utilisent généralement des techniques d'automatiques et d'autres méthodes sans modèle basées sur des outils statistiques ou d'intelligence artificielle, une classification des différentes méthodes est présentée en figure 7 [16], [42], [43]. A notre connaissance, le pronostic des SEDs a été peu exploré et présenté dans la littérature en le comparant au diagnostic et plusieurs noms peuvent s'y référer, notamment prédictibilité et prédiction de la trajectoire [17] :

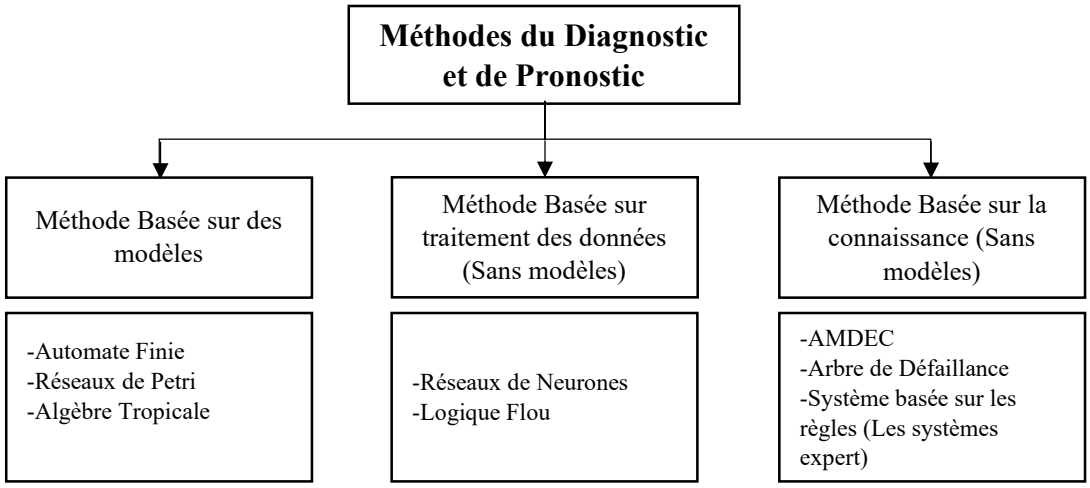
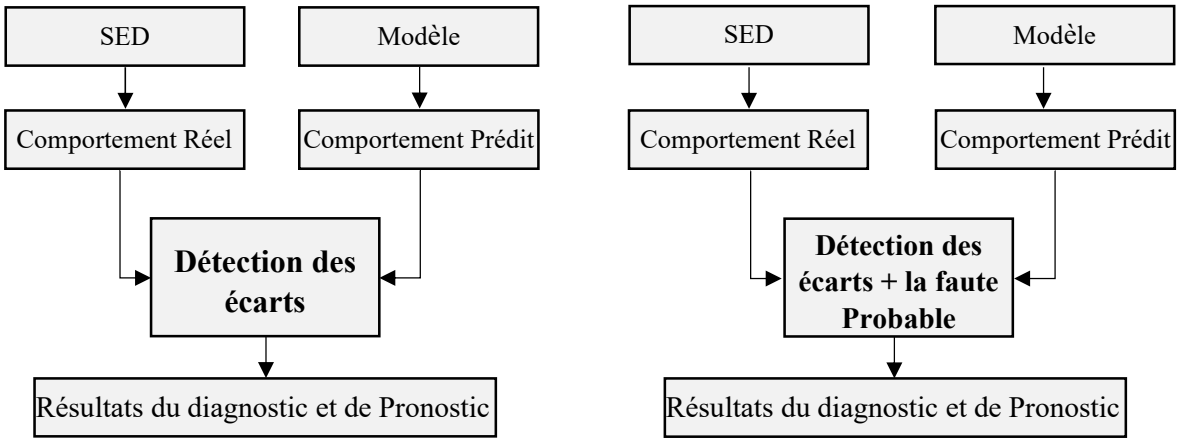


Figure 7: Méthodes du Diagnostic et de Pronostic des SEDs

4-7-3- Approches Basée sur des Modèles

Les approches de diagnostic et de Pronostic basées sur des modèles nécessitent une connaissance physique profonde du SED sous investigation, les résultats du diagnostic et de pronostic se basent principalement sur la comparaison du comportement réel observé sur le SED avec le comportement prédit à l'aide des modèles. Il existe deux types principaux de modèles [4], [8]: des modèles basés sur le comportement normal du system (Figure 8-a), ce type de modèles permet la détection d'incohérences dans le fonctionnement du system, le deuxième type, est des modèles basés à la fois sur le comportement nominal et son comportement en présence de faute (Figure 8-b), ce type de modèles permet la détection d'incohérences ainsi que fournir une estimation de la faute probable responsable de cet écart.



a) Modèle basée sur le fonctionnement b) Modèle basée sur le fonctionnement normal et

Figure 8: Les deux types des modèles du Diagnostic et de Pronostic

Plusieurs travaux de recherche et des approches ont été proposés en se basant sur des modèles dans le contexte du diagnostic et du pronostic des fautes dans les SEDs. Deux principales peuvent se distinguer : des modèles du diagnostic et de pronostic basé sur les AFs, Ces méthodes peuvent être classées en deux catégories : des méthodes basées sur les événements et des méthodes basées sur l'état, et des modèles du diagnostic et de pronostic basé sur les RdPs.

A) Méthode pour le Diagnostic des SEDs

- ***Approches à Base d'Evénements ou Diagnostiqueur***

L'approche diagnostiqueur a été introduite dans les travaux de Sampath et al., [18], [19], elle représente les bases formelles du diagnostic des SEDs en utilisant les AFs. L'approche est basée sur un modèle d'AF non temporisés. Le diagnostiqueur basé sur un au tomate G notée par Gd est un graphe déterministe qui joue le rôle d'un estimateur d'état qui permet de connaître l'état actuel du système grâce à des mesures tel que $Gd = (X, \sum, \delta, \tau, X_0, X_m)$ de sorte que $X_0 = \{0N\}$, ce qui signifie que le SED est à son état initiale dans son fonctionnement 'Normal'. Chacun de ses états dispose une désignation (une étiquette) qui reflète l'état actuel probable du système : **N** (Pour Designer l'état 'Normal'), **F** (Pour designer l'état 'Fautif'), le diagnostiqueur est basé à la fois sur le comportement normal et fautif du SED. Un état contient donc une ou plusieurs étiquettes, c'est-à-dire que chaque état est considéré comme un sous ensemble de X x $\{N, F_1, F_2, \dots, F_\alpha, \dots, F_P\}$. Un état est dit normal (respectivement fautif) si toutes ses étiquettes sont des N (Respectivement F) dans ce cas l'état est dit $N - Certain$ (Respectivement $F - Certain$). Dans le cas contraire, l'état est dit incertain ou ambigu. Un langage généré par un automate est considéré comme diagnosticable si et seulement si son diagnostiqueur ne comporte aucun état incertain.

Exemple 1 : On considère le SED modélisé par AF et son diagnostiqueur (figure 9) adopté de [4] tel que $\sum_O = \{a, b, c, d\}$, $\sum_f = \{f\}$, $\sum_r = \{u\}$, $X_N = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ et $X_F = \{F_7\}$.

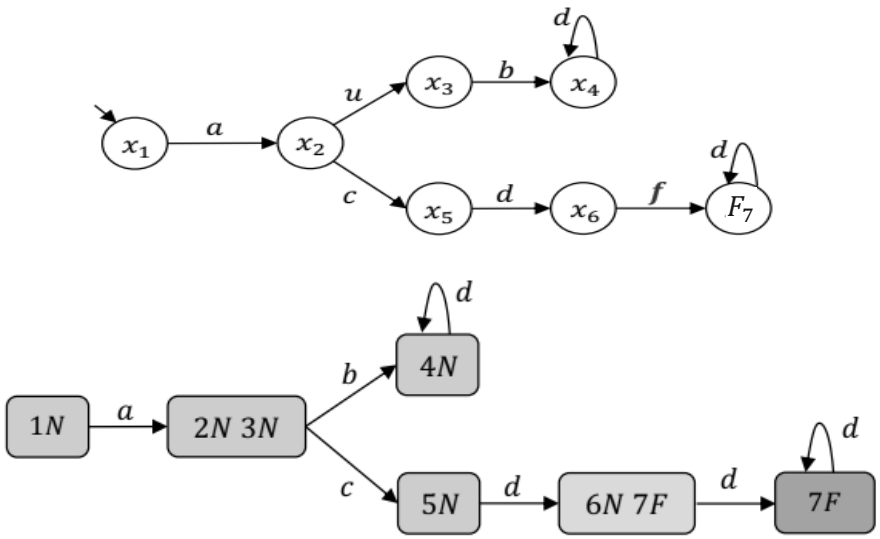


Figure 9: Un SED modélisé par automate fini et son diagnostiqueur à base d'événement

Considérons que l'évènement a c'est produit, Le diagnostiqueur montre qu'on est dans l'état $\{2N\ 3N\}$, c'est-à-dire que le système peut être à l'état x_2, ou à l'état x_3 qui sont des états de fonctionnement normal (L'état probable du SED est mis à jour à chaque occurrence d'un nouveau événement), si on considère que la séquence des événements générés après a est $s = cdd$, dans ce cas, il n'existe qu'une seule évolution possible du système en passant l'état $\{2N\ 3N\}$ (état normal), $\{6N\ 7F\}$ (état ambigu), et qui se termine à l'état $\{7F\}$, c'est-à-dire qu'après l'occurrence de la séquence d'événement $s' = acdd$ le SED se trouve certainement dans un mode de fonctionnement fautif, dans ce cas, on peut déduire que l'évènement de faute f ou l'état de faute est diagnostiqué sans ambigüité.

- ***Approches à Base d'Etat***

Une méthode basée sur l'état, proposé par Zad et al., [44], a été proposée parmi les travaux de recherche qui ont été élaborés pour le diagnostic des SEDs. Au lieu de s'appuyer sur les événements, ce type de méthode se concentre sur l'identification de l'état de défaillance du système, c'est-à-dire cette approche vise à déterminer le fonctionnement de faute du système sans avoir besoin des évènements. Cette méthode part du principe qu'il est possible de diviser l'ensemble des états du système en sous-ensembles basés sur le comportement du système (un mode de fonctionnement normal distinct et un certain nombre de modes de fonctionnement de fautes), de plus, le modèle est étendu pour incorporer toutes les sorties du SED, afin que le fonctionnement du système soit déduit en se basant sur l'ensemble des sorties du SED. Chaque état de cet outil de diagnostic comprend une sortie du système ainsi que tout état potentiel associé à cette sortie et tout mode de fonctionnement potentiel du système correspondant à ces états. L'interprétation du diagnostiqueur se fait de la même manière de celle présenté dans l'approche à base d'événements.

Exemple 2 : On considère le SED illustre dans la figure 10, un exemple inspiré de [44], qui présente un modèle SED et son diagnostiqueur à base d'état, où les arcs en pointillés représentent un événement de faute, et les arcs complets représente les événements observables. Dans ce système, $X = \{1, 2, \ldots, 9\}$, $X_N = \{1, 7, 8, 9\}$, $X_F = \{2, 3, 4, 5, 6\}$ et $Y = \{\eta, \alpha, \beta, \gamma, \nu\}$ tel que Y représente l'ensemble des sorties du SED, Le diagnostiqueur

indique que les états {3, 7}, {4, 8}, et {5, 9} forment un cycle 'incertain' (des états ambigus), c'est-à-dire, il est possible d'avoir une faute et de retrouver le système dans un mode de faute F, après l'apparition d'une défaillance, la séquence de sortie $\alpha\beta\gamma$ ne sera généré qu'une seule fois, après quoi ν sera généré, ce qui amènera le diagnostiqueur à un état de faute certain (état {6}).

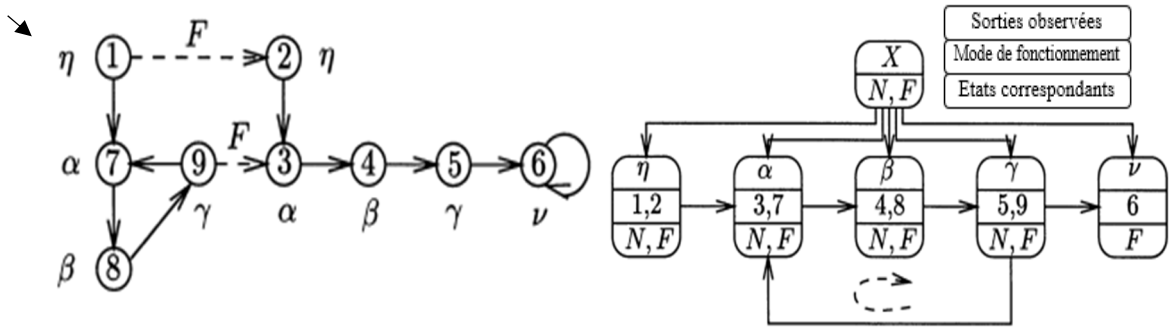


Figure 10 : Un SED modélisé par automate fini et son diagnostiqueur à base d'événement

- ***Approche basée sur les P-invariant***

En général, les approches de diagnostic des SEDs en se basant sur les RdPs visent à déterminer la distribution ou le nombre des jetons dans chaque place (place bornée et non-bornée) du RdP. L'une des premières approches est celle présentée dans [23], dans cette approche l'auteur a développé une technique de diagnostic des SEDs en se basant sur le nombre de jetons dans les P-invariant, c'est-à-dire que lorsque le nombre de jetons dans un P-invariant change dans ce cas une faute est survenu dans le système, sinon le système fonctionne correctement.

- ***Approches basées sur les techniques de décodage algébrique***

L'une des approches de diagnostic basées sur les techniques de décodage algébrique est l'approche développée par Wu et al., [45] dans laquelle deux types de faute sont considérées : Des fautes sur la place du RdP qui peuvent modifier le nombre des jetons dans les places ou le marquage du RdP, et des fautes sur les transitions qui peuvent provoquer une évolution incorrecte du marquage de RdP suite à l'occurrence d'un événement. Cette approche a été rétrécie par Lefebvre et al, [46] de telle sorte que les auteurs ont cherché de déterminer des places exactes qu'il faut suivre pour une estimation exacte et immédiate de l'apparition des fautes.

- ***Approche basée sur les Graphes des Classes d'Etats***

L'approche présentée dans cette partie est basée sur les RdPs temporisés, cette approche permet de prendre en considération le facteur du temps pour assurer l'opération du diagnostic des SEDs, c'est-à-dire, d'exploiter l'ensemble d'événements mesurés avec leurs dates d'occurrence, dans le but de détecter l'occurrence des fautes. Le principe de l'approche est similaire à celui du diagnostiqueur en ce qu'il est basé sur le développement d'un graphe d'estimateur d'état. Dans cette approche, chaque transition du RdPs est dotée d'un intervalle de temps de franchissement, chaque intervalle de temps est composé d'une durée minimale et maximale ou une transition sensibilisée (Une transition est dite sensibilisée si toutes ses places d'entrée

contiennent au moins un jeton.) doit être tirée, dans le cas contraire, une violation d'une contrainte temporelle est détectée, ce qui résulte à un état de faute [47].

- ***Approche basée sur les Graphes d'accessibilité de base***

Dans [48], [49], [50] les auteurs ont proposé une autre technique basée sur les graphes d'accessibilité de base, cette approche s'appuie sur les RdPs étiquetés afin d'élaborer une partie réduite du graphe de marquage. Ce graphe est construit en utilisant la notion des explications minimales inspirée de [51] et qui correspondent aux séquences d'évènements non mesurés de longueur minimale qui doit être produit avant le franchissement d'un évènement mesuré.

- ***Approche basée sur l'automate flou***

Dans cette approche, les auteurs essayent de combiner le principe de l'AF et celle de la logique floue, ce qui résulte à une automate floue (FA). Dans l'FA, les états flous sont représentés par des vecteurs, dont les valeurs indiquent la possibilité (ou le degré d'appartenance) du système à un état flou. Les événements flous sont représentés par des matrices de transition d'état qui indiquent la probabilité que le système passe d'un état flou à un autre. En outre, les événements flous ont un degré d'appartenance aux ensembles d'événements observables ou inobservables. La modélisation par FA permet de modéliser des situations subjectives, ce qui la rend adéquate pour modéliser les occurrences de fautes.

B) Méthodes pour le Pronostic des fautes dans les SEDs

L'approche du Pronostic basée sur les AFs développée par Genc et al., [52] est inspirée d'une grande partie de l'approche diagnostiqueur proposé par Sampath el al., son principe de base est d'analyser les mesures actuelles afin d'anticiper l'apparition des fautes. Pour cela, un modèle pronostiqueur équivalent à celui-ci présenté précédemment, est construit. Sur la base des événements mesurés, pour chaque état atteint du diagnostiqueur, toutes les continuations possibles sont analysées. Lorsque celles-ci mènent vers des états de fautes certaines, alors la faute peut être prédite.

Exemple 3 : On considère le SED présenté dans l'exemple 1, le diagnostiqueur illustré dans la figure 9 est utilisé à la fois pour estimer l'état actuel du SED, mais aussi pour prédire les comportements futurs. Si on considère que l'événement a est généré, l'état $\{2N\ 3N\}$ est obtenu, c'est un état normal, car il ne contient que les étiquettes N. A partir de cet état, il existe deux évolutions possibles, soit une évolution à l'état $\{4N\}$ ou une autre évolution a l'état $\{5N\}$ dans ce cas il existe une seule évolution possible du SED qui passe par l'état incertain $\{6N\ 7F\}$ et qui se termine par l'état $\{7F\}$, qui est un état de faute certaine. Dans ce cas, en se basant sur la séquence d'événement $s = ac$ le système évoluera forcément vers un état de faute, dans ce cas l'événement f ou l'état F_7 sont prédictible.

Le pronostic des SEDs modélisés par des automates temporisés a été étudié aussi par Cassez et Grastien [37]. Ils évoquent l'importance du temps dans le pronostic des fautes des SEDs, dans lequel il est plus important de connaître la période de temps plutôt que le nombre d'occurrences d'événements avant l'apparition des fautes. La connaissance du temps restant avant l'apparition de la faute est utile pour prendre des mesures telles que l'arrêt ou la reconfiguration du système.

Une approche temporisée est combinée avec des réseaux de Petri stochastiques dans les méthodes de pronostic est développée par Ammour et al. dans [53], [54], [55], [56].

Zhao et al. [57] ont défini la notion de pronosticabilité relative. Ils créent un taux qui indique combien d'occurrences de l'événement sont pronostiquables. Ainsi, un système qui ne satisfait pas les conditions de pronosticabilité, peut satisfaire les conditions de prédictibilité relative.

4-6-4- Approches Basée sur la Connaissance

Le concept de base de ces approches est de construire des liens de causalité entre les symptômes, les défaillances et les erreurs en se basant sur les connaissances observables du système. Ces informations peuvent être obtenues par un examen structurel ou fonctionnel du système. Il existe plusieurs techniques qui s'appuient sur la connaissance. Dans cette partie, nous décrivons brièvement quelques approches.

AMDEC : Cette méthode est basée sur une analyse détaillée du SED, composant par composant, de tous leurs modes de défaillance, leurs causes, leurs effets et leurs degrés de criticité sur le fonctionnement du système et de son environnement. La première étape de la méthode AMDEC consiste à effectuer une analyse détaillée des différentes fonctions et composants du SED, puis il faut déterminer leurs modes de défaillance possibles ainsi que leurs causes possibles et enfin déterminer ces effets sur le fonctionnement du SED pour chaque couple (cause, mode de défaillance) [58].

Arbre de Défaillance : C'est une méthode très puissante et efficace pour identifier les causes profondes menant à un mode de défaillance. En général, ses arbres de défaillance sont utilisés pour l'analyse de la sécurité et de la fiabilité des systèmes et il permet de mettre en évidence les relations de cause à effet. Il permet de rechercher les causes directes de l'apparition d'un événement de faute en corrélation avec une défaillance du SED. Les relations entre les différents événements sont construites à l'aide des portes logiques d'algèbre booléenne. Afin d'établir un diagnostic correct à l'aide de l'arbre de défaillance, il est nécessaire de représenter toutes les relations de cause à effet du SED, ainsi que toutes les différentes défaillances possibles [58].

Systèmes Experts : Ce sont des systèmes informatiques dont le but est de résoudre un problème spécifique en analysant et en représentant les connaissances et les raisonnements des spécialistes de ce problème. D'une autre manière, les systèmes experts sont basés sur des règles qui permettent d'associer les effets à leurs causes, ceci permet de relier chaque comportement anormal de système avec une défaillance potentielle [58].

SI [A = B] et [A => C] donc [B = C]

Ces relations sont généralement fondées sur l'expérience des experts plutôt que sur la connaissance de la structure et du comportement des systèmes.

4-6-5- Approches Basée sur le Traitement de Données

Les approches fondées sur les données nécessitent la disponibilité de vastes volumes de données provenant des enregistrements du système. Ces approches s'appuient en général sur des méthodes mathématiques et des outils d'intelligence artificielle qui permettent l'apprentissage et la catégorisation de l'ensemble de la base de données afin de construire des modèles de diagnostic et de pronostic de référence.

Réseaux de Neurones : Ce sont des outils d'intelligence artificielle largement utilisés pour résoudre une variété de problèmes, de sorte que ses progrès s'étendent à plusieurs industries et applications. En effet, grâce à leur capacité d'apprentissage, ils sont utilisés pour résoudre les problèmes les plus complexes dans plusieurs domaines. Cette capacité est le résultat d'un processus intelligent appelé apprentissage ou formation. Le rôle principal de ce processus est le suivant : Sur la base d'un ensemble de données présentées au RdN, celui-ci établit une relation entre l'ensemble des entrées et l'ensemble des sorties afin de tirer parti et de généraliser les connaissances obtenues au cours de la formation à des nouveaux ensembles de données. Un RdN est généralement formé de nœuds organisés en couches, qui peuvent être divisées en trois types : couche d'entrée, couche cachée et couche de sortie. Ainsi, chaque nœud peut être connecté au nœud de la couche suivante, dans le cas des RdNs à action directe (feed-forward), ou peut être relié à n'importe quel autre nœud, voire à lui-même, dans le cas des RdNs récurrents [59], [60] (Une description détaillée sur les réseaux de neurones, leurs processus d'apprentissage et leurs types sera introduit dans le chapitre 2).

Logique Floue : La notion de logique floue permet d'étendre la notion de logique classique, de sorte que contrairement à la logique booléenne, qui permet d'associer à une variable deux valeurs possibles 0 ou 1, la logique floue permet d'associer à la variable "x" un coefficient d'appartenance $\mu_A(x)$ prenant des valeurs comprises entre 0 et 1 quantifiant l'incertitude de la variable d'appartenir à un ensemble flou A. Dans le cas où le degré d'appartenance à un ensemble flou est égal à 0/1, il représente respectivement une incertitude/certitude totale de l'événement de la variable [61], [62].

5- Etude Comparative entre les Différentes Approches du Diagnostic et de Pronostic des SEDs

5-1- Critères de Comparaison

En se basant sur une analyse approfondie de [63], [64], [65], [66] et [67], nous avons été en mesure d'extraire les exigences suivantes pour la mise en œuvre d'un outil de diagnostic et de pronostic des SEDs dans les industries modernes :

- **Facilité de mise en œuvre :** Facile à mettre en place et à appliquer le système de diagnostic et de pronostic, c'est-à-dire que l'approche doit être facile à réaliser, à exécuter et à mettre en pratique.
- **L'Internet industriel des objets :** Les machines, les diagnostiqueurs et les pronostiqueurs doivent être connectés dans un sens multidirectionnel et en temps réel à travers un réseau qui les relie et les combine avec l'Internet, afin d'assurer leur communication et l'échange d'informations pour garantir le fonctionnement du diagnostic et de pronostic.
- **Flexible :** Adaptation rapide et facile de la méthode à des petits changements.
- **Facile à reconfigurer :** Il est facile de redéfinir les paramètres du model pour les adapter à un nouvel environnement.

- **Analyse de grandes quantités de données :** La méthode de diagnostic doit assurer une combinaison ainsi qu'une analyse intelligente de grandes quantités de données provenant de divers capteurs pour une aide à la prise de décisions ciblées, basées sur les données.
- **Minimiser l'intervention humaine :** La méthode de diagnostic et de pronostic doit optimiser au maximum toutes les formes d'intervention humaine (évaluation, mesure des grandeurs du diagnostiques, décisions simples, etc.).
- **Diagnostic/Pronostic en ligne :** Le diagnostiqueur et le pronostiqueur doivent effectuer l'opération de diagnostic et de pronostic pendant le fonctionnement du système.
- **Conserver la traçabilité en temps réel :** La méthode doit conserver les informations nécessaires et suffisantes (résultats du diagnostic et de pronostic) tout au long du fonctionnement du système.
- **Basé sur des algorithmes automatiques et intelligents :** Les algorithmes permettent la déduction automatique et intelligente des décisions de diagnostic et de pronostic, par exemple, le développement basé sur l'apprentissage automatique d'algorithmes intelligents pour diverses utilisations, c'est-à-dire que la méthode de diagnostic et de pronostic doit être basée sur des approches basées sur les données comme la reconnaissance des formes.

5-2- Forces et Faiblesses des Approches du Diagnostic et de Pronostic des SEDs

Dans la pratique, chacune de ces approches présentées ci-dessus présente des avantages et des inconvénients. Sur la base des différentes recherches et études de cas présentées comme références dans le tableau 1, il a été possible d'extraire les différentes forces et faiblesses que présente chaque méthode en pratique.

Tableau 1 : Forces et faiblesses des approches du diagnostic et de pronostic des SEDs

	Forces	Réf.	Faiblesses	Réf.
AMDEC	-Aucun modèle de système n'est nécessaire. -Donne une bonne vision du système. -Fournit un support pour la réflexion, la détection et l'amélioration. -Aide à la mise en place de plans de maintenance préventive. -Aide à déterminer les points faibles du système. -Peut s'adapter facilement à un nouvel environnement.	[58], [68], [69]' [70].	-Nécessite beaucoup d'expérience pour l'établir et très lourd à mettre en place. -N'est pas adapté au diagnostic et de pronostic des systèmes en temps réel (en ligne). -Ne permet pas de détecter la propagation des défaillances. -La détermination de la probabilité exacte de défaillance est difficile et parfois impossible. -L'influence des paramètres est considérée souvent avec un même poids et l'abutent entre les scores n'est pas clair. -Il ne considère que l'impact d'une seule défaillance sur le système ; l'évaluation de modes de défaillance multiples avec toutes les combinaisons possibles n'est pas pratique.	[68], [71], [72], [73].
Arbre de défaillance	-Facilité de mise en œuvre et de compréhension du système. -Aide à l'identification des points faibles du système. -Effectuer des calculs rapides. -Capacité à s'adapter facilement à un nouvel environnement.	[58], [74].	-L'analyse ne porte que sur un seul événement déclencheur à la fois. -L'analyste doit identifier l'événement déclencheur et les voies d'accès, et le niveau de perte pour chaque voie d'accès peut ne pas pouvoir être distingué sans une analyse plus approfondie. -Les probabilités de réussite ou d'échec sont difficiles à trouver et ne peuvent pas être distinguées.	[75], [76]
Systèmes Expert	-Aucun modèle de système n'est requis. -Temps de calcul optimal. -Les résultats sont très simples à comprendre par un opérateur. -Facilité de mise en œuvre de ces systèmes. -Haute capacité de traitement des données. -Détection des fautes en ligne.	[77], [78], [79]	-Ces systèmes dépendent de l'expertise. -Les règles obtenues pour un système ne peuvent jamais être appliquées à un autre. -Ces systèmes ne peuvent pas être opérationnels au début de leurs opérations. -Ces systèmes ne peuvent pas donner des interprétations et des explications des résultats obtenus.	[14], [80], [5],

Logique Flou	-Aucun modèle de système n'est requis. -Donne une réponse simple qui est difficile à modéliser. -Tient compte plusieurs variables. -Détection des fautes en ligne.	[62], [81], [82]	-Difficile de s'adapter à un environnement changeant. -Les résultats dépendent de la fonction d'appartenance choisie et de la fiabilité des données d'apprentissage. -Doit être utilisé à côté d'autres techniques de contrôle avancées. -Le développement d'un système flou requiert une grande expertise.	[62], [83]
Réseaux de Neurones	-Offre des solutions compactes et rapides pour un large éventail de problèmes. -Algorithmes d'apprentissage multiples. -L'utilisation ne dépend pas d'un modèle formel du système. -Grande capacité de traitement des données. -Les modèles sont réutilisables et peuvent être mis à jour. -Détection des fautes en ligne. -Des connaissances de base en Python ou Matlab sont suffisantes pour un diagnostic et un pronostic approfondi.	[35], [84], [85], [86].	-Apprentissage à partir de zéro. -Les résultats dépendent de la fiabilité des données d'apprentissage et de l'architecture neuronale choisie. -L'expert doit connaître le fonctionnement correct et les modes de défaillance.	[87], [88]
Automate Finie	-Description précise du comportement du système. -Visualisation des conflits de fautes possibles dans le système. -Facilité de manipulation grâce aux outils de composition ainsi qu'à la théorie des langages et des projections. -Détection des fautes en ligne.	[19], [18], [26], [89].	-Risque d'explosion combinatoire. -Nécessite une connaissance physique approfondie du système. -Difficulté de modélisation et de reconfiguration du modèle pour les systèmes complexes. -Requiert la traduction du modèle par automate fini vers un observateur et un diagnostiqueur. -L'existence d'états incertains. -Difficile d'être compris par un opérateur.	[4], [84], [90].
Réseaux de Petri	-Bonne adaptation pour la description de la dynamique des systèmes. -Détection de fautes en ligne. -Les modèles utilisés pour le diagnostic et le pronostic du système ont la même structure que le système. -Existence d'un grand nombre de méthodes pour réaliser le diagnostic et le pronostic.	[4], [23], [21], [22].	-Risque d'explosion combinatoire. -Nécessite une connaissance physique approfondie du système. -Difficile de modéliser et de reconfigurer les modèles pour les systèmes complexes. -Il nécessite un effort de calcul considérable. -Difficile d'être compris par un opérateur.	[84], [90], [91].

L’étude comparative présentée montre que :

- L'AMDEC et les arbres de défaillance ont une bonne flexibilité face à l'intégration d'un nouveau composant dans le système. Cependant, dans les mêmes conditions, d'autres outils nécessitent une reconstruction totale, mais la mise en œuvre de l'AMDEC, de l'arbre de défaillance ne peut pas être programmée dans un matériel permettant la connexion au SED, ils ne peuvent donc pas traiter le diagnostic et le pronostic en ligne.

- L'AMDEC, l'arbre de défaillance et les réseaux de neurones peuvent être facilement reconfigurables, afin d'adapter leur fonctionnement à un nouveau mode de fonctionnement du système. D'autres outils nécessitent une modification totale, mais les systèmes complexes à événements discrets peuvent inclure un grand nombre de composants et une énorme variété de modes de fonctionnement, ce qui rend impossible l'application des méthodes basées sur l'expertise, comme l'AMDEC, l'arbre de défaillance ou les systèmes experts.

- Les RdNs, les systèmes experts et la logique floue sont considérés comme des branches de l'intelligence artificielle, car ils peuvent traiter l'analyse de grandes données en utilisant leurs capacités d'intelligence artificielle et sont basés sur de nombreux automatismes et algorithmes intelligents, mais leurs performances sont directement liées à la fiabilité des données fournies à ces outils.

- Les automates finis et les réseaux de Petri sont difficiles à mettre en œuvre, car leur réalisation nécessite une grande maîtrise des outils de modélisation et une forte connaissance du fonctionnement du système. Par conséquent, dans les systèmes complexes à événements discrets, les outils basés sur les modèles peuvent générer des modèles très complexes qui peuvent être difficiles à construire et à élaborer. Alors que d'autres outils tels que les réseaux de neurones ne nécessitent que des données collectées et un réseau de neurones préprogrammé pour analyser ces données ou un système expert qui construit un ensemble de règles pour faire dépendre chaque symptôme de sa cause probable. De plus, l'utilisation des réseaux de Petri et des automates finis peut générer une explosion combinatoire.

- Les systèmes experts ne peuvent pas traiter l'opération de diagnostic et de pronostic au début de ses opérations ; d'autres outils peuvent être utilisés au début de leurs opérations.

Le résultat de cette comparaison montre que les RdNs peuvent fournir certains avantages, que d'autres outils ne peuvent pas traiter. En effet, grâce à leurs algorithmes automatiques et intelligents, les RdNs sont capables d'apprendre, d'évoluer et de généraliser, ainsi que de traiter différents types de données. En outre, à partir d'une grande base de données, ils peuvent extraire des informations utiles, qui peuvent être utilisées pour l'opération de diagnostic et de pronostic, et grâce à cela, ils peuvent assurer et garantir la rapidité et l'efficacité de ces opérations.

6- Conclusion

Dans ce premier chapitre de thèse, nous avons mis en évidence quelques définitions et formalismes à propos des SEDs, par la suite, nous avons approfondi les méthodes de diagnostic et de pronostic des SEDs retrouvées dans la littérature et à travers une étude comparative, qui a été planifiée à l'aide de plusieurs critères, nous avons constaté que des modules de Diagnostic et de Pronostic basées sur des modèles deviennent difficile et coûteuse, en offrant des performances moins satisfaisantes, ce qui fait appel à des outils de plus en plus sophistiqués et intelligentes. A la fin de cette comparaison, nous avons conclu qu'un module de diagnostic et de pronostic basée sur les réseaux de neurones pouvaient parfaitement traiter les problèmes de diagnostic et de Pronostic des SEDs, grâce aux différents avantages présentés par cet outil.

Dans le chapitre suivant, une présentation détaillée sur les premiers travaux, qui ont servi à mettre les formalismes de base des RdNs et comment ils ont évolué pour donner naissance aux systèmes que nous connaissons aujourd'hui.

Chapitre 2 : Les Réseaux de Neurones

Table des Matieres

1- Introduction 44

2- Neurone Formel 44

3- Entraînement des Réseaux de Neurones 45

4- Algorithmes ou Optimiseurs d'Apprentissage 47

4-1- Notation 47

4-2- Règle de Hebb et Rosenblatt 47

4-2- La règle de Delta 48

4-3- Descente de gradient par rétropropagation 48

4-4- Variante de descente de gradient par rétropropagation 50

4-4-1- Descente de gradient par batch (BGD) 50

4-4-2- Descente de gradient stochastique (SGD) 50

4-4-3- Descente de gradient en mini-batch (MGD) 51

4-5- Algorithmes d'optimisation par Descente de Gradient 51

4-5-1- Descente de gradient avec Momentum 51

4-5-2- Gradient accéléré de Nesterov (NAG) 52

4-5-3- Rprop 52

4-5-4- ADAgrad 53

4-5-5- RMSprop 54

4-5-6- ADAM 54

4-5-7- LARS 55

5- Les Réseaux de Neurones Récurrent 56

5-1- Définition 56

5-2- Long Short-Term Memory Network (LSTM) 57

5-3- Gated Recurrent Unit (GRU) 57

6- Les Applications autour des Réseaux de Neurones 57

6-1- La classification 57

6-2- La régression 58

6-3- Diagnostic et Pronostic des SEDs par les RdNs 59

7- Conclusion 60

1- Introduction

Dans un passé récent, les réseaux de neurones ont représenté un progrès extraordinaire dans le domaine de l'intelligence artificielle [92], de sorte que les conséquences de ce progrès s'étendent à plusieurs industries et applications. En effet, grâce à leur capacité d'apprentissage, ces outils sont utilisés pour résoudre les problèmes les plus complexes : classification et génération d'images et de textes, analyse de sentiments, prévision de séries temporelles, etc. Les réseaux de neurones offrent une réponse intrigante aux problèmes du diagnostic et de pronostic des équipements industriels. En effet, leur utilisation ne dépend pas de la disponibilité d'un modèle formel de cet équipement. De plus, leurs capacités de mémorisation, d'apprentissage et d'adaptation, sont des caractéristiques extrêmement utiles pour tout système de surveillance [9]. Dans ce chapitre, nous allons présenter une vue générale sur ces outils, une brève présentation de l'évolution historique, leur mode de fonctionnement, leur algorithme d'apprentissage ainsi que leurs types.

2- Neurone Formel

Les neurones simples ou neurones formels [93] (figure 11) sont considérés comme un modèle mathématique inspiré du principe de fonctionnement d'un neurone biologique, Mc. Culloh et al., l'ont développé en 1943. Le comportement d'un neurone formel est composé de deux phases principales :

Phase 1 : La phase d'activation, qui permet de calculer la somme pondérée des entrées telles que :

$$a_i = \sum_n W_{in} * x_{ni} + b_i \qquad 1$$

Tel que : W_{in} représente les poids des neurones et b_i son bias.

Phase 2 : La phase de calcul de la sortie du neurone, à partir de la valeur de a_i, $\hat{Y}$ est calculé, tel que :

$$\hat{Y} = f(a_i) \qquad 2$$

Telle que la fonction f est une fonction de transfert (ou fonction d'activation) appliquée à a_i.

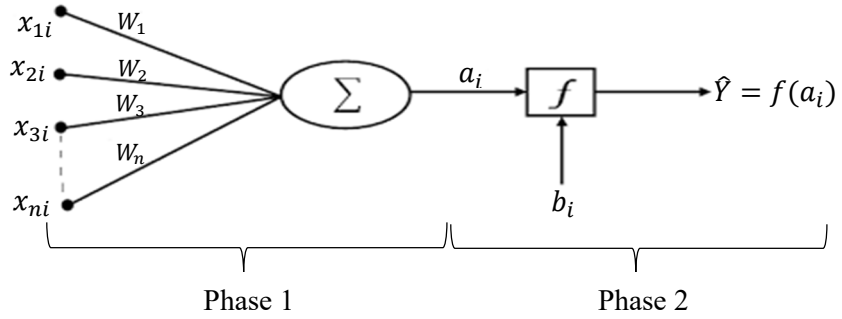


Figure 11: Neurone Formel

Remarque 2 : La fonction d'activation ou de transfert relie la sommation pondérée au signal de sortie. Il existe plusieurs types de fonctions d'activation. La figure 12 montre les fonctions d'activation les plus couramment utilisées.

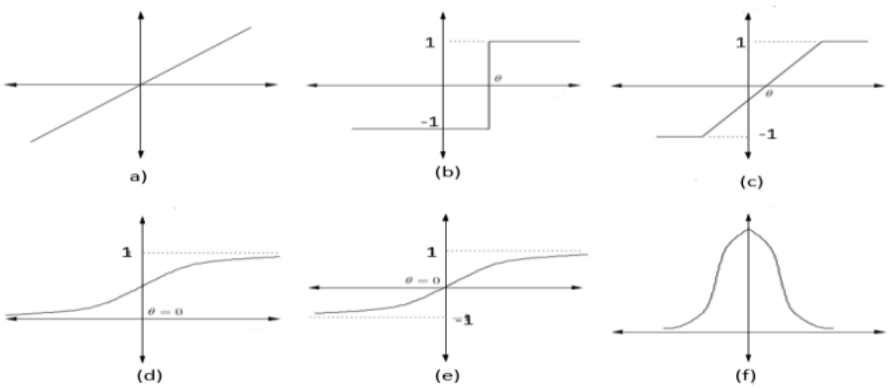


Figure 12 : Les fonctions d'activation les plus fréquemment utilisées, a) fonction linéaire b) fonction de seuil c) fonction linéaire par morceaux d) fonction sigmoïde e) fonction tangente hyperbolique f) fonction gaussienne

Le neurone formel seul n'a pas une grande force de calcul, mais cette force apparaît lorsqu'il est connecté à d'autres neurones formels. Un RdN est composé d'un ensemble de neurones formels formant des couches connectées les unes aux autres, et chaque connexion entre deux neurones formels est associée à un poids W.

3- Entraînement des Réseaux de Neurones

La capacité de calcul des réseaux de neurone est le résultat d'un processus intelligent appelé apprentissage, optimisation ou processus de formation. La formation, l'apprentissage ou l'optimisation des RdNs est la phase la plus importante pour leur bon fonctionnement. Dans cette phase, un ensemble de données (jeu de données d'apprentissage) est injecté au RdN, et en fonction de ce jeu de données, certains changements ont lieu entre les connexions des neurones, de sorte que le RdN ajuste ces paramètres (poids et biais), de sorte que certains paramètres sont renforcés et d'autres affaiblis ou parfois inhibés. Afin d'obtenir les meilleures performances possibles et les résultats les plus précis. L'apprentissage peut être de différents types [94] :

- **Apprentissage supervisé :** Consiste à présenter au réseau le vecteur d'entrée avec ses sorties associées.
- **Apprentissage non supervisé :** consiste à présenter au réseau les entrées uniquement et à le laisser reconfigurer ses paramètres sans prendre en compte les sorties.
- **Semi-supervisé :** consiste à reconfigurer les paramètres du réseau en deux phases, une supervisée et un non supervisé.
- **Apprentissage par renforcement :** Dans ce type, le réseau de neurone reçoit des informations sur son environnement et apprend à choisir des actions qui maximiseront certaines récompenses.

Les applications pratiques actuelles sont principalement axées sur l'apprentissage supervisé [95]. Presque toutes les applications appartiennent à cette catégorie [96].

L'entraînement d'un RdN d'une manière supervisée s'articule autour de 5 objets [59] :

- Les couches : Ce sont les blocs de construction du réseau neuronal, et il y a 3 types de couches :

  - Les couches d'entrée : Elles initialisent les données pour le réseau de neurone.

  -Les couches de sortie : Elles produisent le résultat pour chaque entrée donnée.

  -Couches cachées : Ce sont les couches situées entre les couches d'entrée et de sortie, et c'est là que tous les calculs sont effectués.

- Les données d'entrée et leur cible correspondante.
- La fonction de perte : La fonction de perte mesure les performances du RdN et le succès de la tâche de formation.
- Le score de perte : La quantité que l'on cherche à minimiser pendant l'apprentissage.
- Optimiseur : Déterminer comment les paramètres du réseau seront mis à jour en fonction de la fonction de perte.

La figure suivante (figure 13) montre les différentes étapes de l'algorithme d'apprentissage :

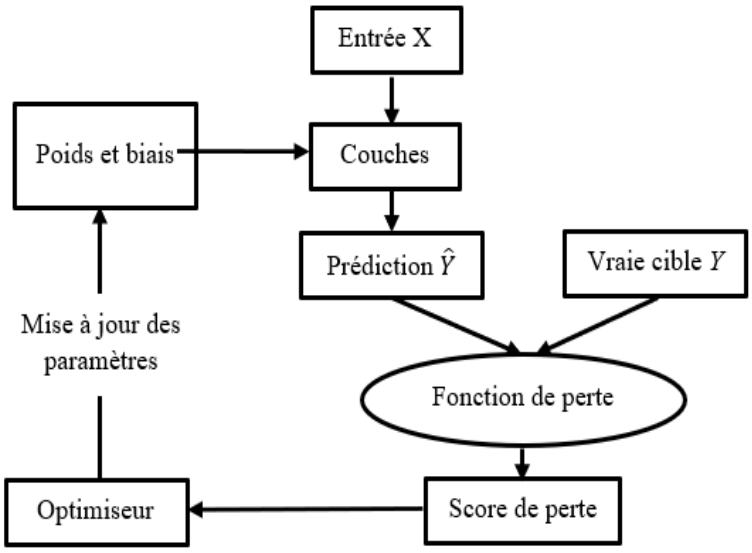


Figure 13: Les différentes étapes de l'algorithme d'apprentissage

Initialement, les paramètres du RdN sont initialisés avec des valeurs aléatoires, après les données d'entrée sont injectées dans le RdN et pour chaque entrée, le réseau essaie de distinguer correctement les exemples d'entrées à leurs cibles associées. Bien sûr, au début, la sortie du RdN est très éloignée de ce qu'elle devrait être. Ainsi, la fonction de perte prend les prédictions du réseau ($\hat{Y}$) et la vraie cible (Y) et calcule un score de distance (score de perte ou erreur, qui est très élevé au début de l'apprentissage), et capturant la façon dont le réseau a réussi sur cet exemple spécifique. Ensuite, l'optimiseur utilise ce score comme un signal de retour afin d'ajuster les valeurs des poids dans une direction qui réduira le score de perte, c'est-à-dire qu'à chaque exemple, les poids sont modifiés dans la bonne direction, et le score de perte diminue. Ainsi, cette opération est répétée un nombre suffisant de fois (chaque répétition est nommée "époque"), afin d'atteindre les valeurs des paramètres qui minimisent le score de perte. Un réseau avec un score de perte minimal est un réseau pour lequel les sorties

ou les prédictions sont aussi proches qu'elles peuvent l'être des cibles réelles. Nous pouvons clairement observer que la partie la plus importante de l'entraînement du RdN est l'ajustement des paramètres. Dans la littérature, il y a plusieurs algorithmes ou optimiseurs qui peuvent s'occuper de ce travail (l'ajustement des paramètres).

Note : Concernant les ajustements, nous ne nous intéresserons qu'à l'ajustement des poids car ils représentent la majeure partie des paramètres du réseau de neurones. De plus, la modification des biais se fera de la même manière que celle des poids.

4- Algorithmes ou Optimiseurs d'Apprentissage

4-1- Notation

x_k : Entrées du réseau de neurones.

a_{li} : L'entrée d'un neurone i dans une couche l.

V_{li} : La sortie du neurone i dans une couche l.

W_{ij}: Le poids entre un neurone i d'une couche l et un neurone j d'une couche $l-1$

$\hat{Y}$: La sortie i du réseau de neurones.

Y : La vraie cible i.

4-2- Règle de Hebb et Rosenblatt

Hebb a introduit la règle de Hebb [97] en 1949 ; elle représente la $1^{ère}$ méthode d'apprentissage qui permet de modifier les poids d'un perceptron simple (figure 14). Le principe de cet algorithme est le suivant : S'il y a deux neurones connectés et activés en même temps, c'est-à-dire que les neurones qui ont tendance à être soit tous les deux positifs, soit tous les deux négatifs en même temps, doivent avoir une forte pondération positive, dans ce cas, le poids de la connexion doit augmenter afin de lui permettre de jouer un rôle déterminant dans le processus d'apprentissage, mais dans le cas contraire, le poids ne doit pas être modifié.

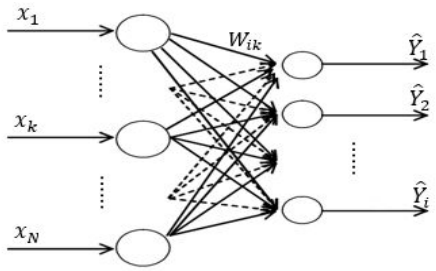


Figure 14: Perceptron simple, modèle de Rosenblatt

Cet algorithme a été l'inspiration ainsi que les bases des travaux de Rosenblatt [98]. Rosenblatt a développé la première version du perceptron, qui est un réseau composé de deux couches : une d'entrée et une de sortie. Cet algorithme permet de modifier les poids des connexions entre les neurones, dans le cas où ils ont obtenu la sortie $\hat{Y}_i$ différente de la sortie souhaitée, en ajoutant une quantité ΔW_{ik}. Par conséquent, la mise à jour des paramètres est :

$$W_{ik}^{new} = W_{ik}^{old} + \Delta W_{ik}^{new} \quad 3$$

De sorte que, ΔW_{ik}^{new} est nouveau pour chaque exemple X de l'ensemble des exemples d'apprentissage, on peut écrire :

$$\Delta W_{ik}^{new} = \alpha \left(Y_i^X - \hat{Y}_i^X\right) x_k^X \quad 4$$

Ainsi, α est le taux d'apprentissage, son rôle est de déterminer la dynamique selon laquelle les modifications des paramètres du réseau seront effectuées. Si alpha est très grand, il permet à l'algorithme de converger rapidement vers le minimum de la fonction de coût, mais en même temps il risque de perdre la valeur minimale en oscillant autour de celle-ci jusqu'à l'infini. En revanche, si la valeur de alpha est très petite, l'algorithme peut prendre un temps infini afin de converger vers la valeur minimale. Par conséquent, pour choisir la meilleure valeur d'alpha, il faut essayer plusieurs valeurs jusqu'on trouve la meilleure.

Tableau 2 : Avantages et Inconvénients de la Règle de Hebb et Rosenblatt

Avantages	Inconvénients
- Une architecture simple	- N'accepte pas les données bruyantes et la classification non linéaire

4-2- La règle de Delta

Dans le modèle de Hebb et Rosenblatt, les fonctions d'activation utilisées sont des fonctions d'activation à seuil. Dans [99], les auteurs ont proposé un modèle avec une fonction d'activation linéaire, et grâce à cette fonction, une fonction de coût qui permet de calculer l'erreur faite par le réseau, peut être calculée. Cette erreur est définie par :

$$C(W,b)\ \frac{1}{2}\sum_i (Y_i - \hat{Y}_i) \quad 5$$

Cet algorithme permet de modifier les valeurs du paramètre de manière à minimiser la valeur de cette erreur, la modification s'effectue en ajoutant une quantité 'delta' proportionnelle au gradient de l'erreur, dans ce cas ΔW_{ik}^{new} pour l'ensemble des exemples d'apprentissage, on écrit :

$$\Delta W_{ik}^{new} = -\alpha \frac{\partial \mathrm{c}}{\partial \mathrm{W_{ik}}} = -\alpha \sum_X \left(Y_i^X - \hat{Y}_i^X\right) x_k^X \quad 6$$

Tableau 3 : Avantages et Inconvénients de la Règle de Delta

Avantages	Inconvénients
- Architecture simple -Permet d'ajuster les valeurs des poids du réseau afin de minimiser l'erreur.	- Cet algorithme ne peut être appliqué qu'aux réseaux à une seule couche avec des poids adaptatifs.

4-3- Descente de gradient par rétropropagation

En étendant la dernière méthode (règle de Delta), dans [100], [101] les auteurs ont proposé un nouveau algorithme afin d'avoir un algorithme qui permet de modifier tous les paramètres d'un perceptron multicouche, puis cet algorithme a été généralisé dans [102]. Cet algorithme est considéré comme l'algorithme le plus utilisé

pour l'entraînement des RdNs en apprentissage supervisé et il est la base de plusieurs autres algorithmes d'apprentissage. La méthode de rétropropagation du gradient permet de calculer comment un paramètre spécifique influence la valeur totale de l'erreur C. La minimisation de la fonction d'erreur globale sera effectuée par une descente de gradient ; par conséquent, la valeur de chaque connexion sera modifiée de la même manière que l'ajustement dans (6), de sorte que :

$$\Delta \mathrm{W} = -\alpha \frac{\partial c}{\partial W} \qquad 7$$

Pour cela, le calcul du delta repose sur le calcul de $\frac{\partial c}{\partial w}$:

Par conséquent, nous considérons un réseau de neurone multicouche (figure 16) qui contient L couches :

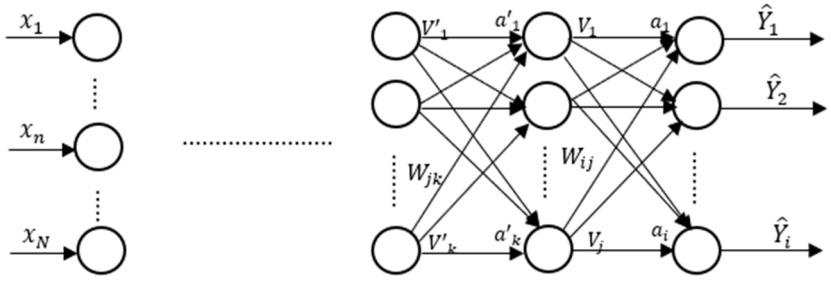


Figure 15: Perceptron multi couche

Pour simplifier les calculs, ils seront effectués pour les connexions entre la couche de sortie (couche L) et la couche $L-1$ ainsi que pour les connexions entre la couche $L-1$ et la couche $L-2$, et les mêmes calculs seront répétés pour les autres couches. De plus, concernant la fonction de perte, nous utiliserons l'erreur quadratique moyenne présentée dans la formule 5, mais en pratique, il existe plusieurs fonctions de perte qui peuvent être utilisées.

- **Cas des connexions entre la couche cachée ($L-1$) et la couche de sortie (L) :**

Dans le cas des neurones de sortie, $\frac{\partial c}{\partial w}$ devient en fonction du paramètre W_{ij} qui n'affecte que la sortie du neurone d'indice i. Par conséquent, nous pouvons la décomposer et nous obtenons :

$$\frac{\partial c}{\partial W_{ij}} = \frac{\partial a_i}{\partial W_{ij}} \frac{\partial \hat{Y}_i}{\partial a_i} \frac{\partial c}{\partial \hat{Y}_i} = V_j(\hat{Y}_i - Y_i) = V_j(\hat{Y}_i) {f_i}'^{(a_i)} \qquad 8$$

Par conséquent, pour tous les exemples d'apprentissage, l'expression devient :

$$\Delta W_{ij} = -\alpha \sum_X {f_i}'(a_i^X)(Y_i^X - \hat{Y}_i^X) V_j^X$$

- **Cas des connexions entre la couche cachée ($L-2$) et la couche cachée ($L-1$) :**

Dans le cas des neurones cachés, $\frac{\partial c}{\partial w}$ devient en fonction du paramètre W_{jk}, qui influence non seulement la sortie du neurone de deuxième couche j, mais aussi tous les neurones de couches de sortie i qui lui sont connectés. Par conséquent, nous pouvons le décomposer et nous obtenons :

$$\frac{\partial c}{\partial W_{jk}} = \frac{\partial a_i}{\partial W_{jk}} \frac{\partial V_j}{\partial a_j} \frac{\partial c}{\partial V_j} = \frac{\partial c}{\partial V_j} f_j'(a_j) V'_k \quad (9)$$

Le 1^{er} terme de cette expression devient :

$$\frac{\partial c}{\partial V_j} = \sum_i W_{ij} \left(\hat{Y}_i - Y_i\right) f_i'(a_i) \quad (10)$$

Par conséquent, pour tous les exemples d'apprentissage, l'expression que nous obtenons est :

$$\Delta W_{jk} = -\alpha \sum_X (f_j'(a_j^X) V'^X_k \sum_i W_{ij} f_i'(a_i^X)(Y_i^X - \hat{Y}_i^X) \quad (11)$$

Enfin, pour chaque mise à jour de poids, nous notons :

$$W^{new} = W^{old} - \Delta W^{new} \quad (12)$$

4-4- Variante de descente de gradient par rétropropagation

4-4-1- Descente de gradient par batch (BGD)

La descente de gradient par batch (lot) calcule le gradient des fonctions de coût des paramètres pour l'ensemble du jeu de données, c'est-à-dire que les paramètres du modèle seront ajustés après avoir calculé le gradient de l'ensemble du jeu de données, c'est-à-dire que les paramètres seront mis à jour une fois par cycle d'apprentissage ou époque. Ainsi, la mise à jour des paramètres pour chaque époque sera la suivante :

$$W(t) = W(t-1) + \Delta W(t) \quad (13)$$

Tel que t est un cycle d'apprentissage.

Tableau 4: Avantages et inconvénients de BGD

Avantages	Inconvénients
- Modifier les paramètres de toutes les couches du réseau. - Facile à mettre en œuvre.	- Si l'ensemble de données est trop grand, l'algorithme peut prendre beaucoup de temps pour converger vers un minimum car il change les valeurs des paramètres après le calcul du gradient de l'ensemble des données. - Nécessite une grande mémoire pour effectuer tous les calculs. - Peut rester bloqué dans des minima locaux. -α est constant pendant l'apprentissage. -Ne peut pas s'adapter à de grands lots.

4-4-2- Descente de gradient stochastique (SGD)

La descente de gradient stochastique [103] est développée pour accélérer la descente de gradient par rétropropagation. Dans cet algorithme, au lieu d'ajuster les paramètres du modèle après avoir calculé le gradient de l'ensemble des données, les paramètres du modèle seront mis à jour après le calcul de la perte sur chaque exemple de l'ensemble d'entraînement. Ainsi, si l'ensemble d'apprentissage contient N exemples, les paramètres seront mis à jour N fois en un cycle d'apprentissage.

Maintenant la mise à jour des paramètres est :

$$W_N(t) = W_N(t-1) + \Delta W_N(t) \quad (14)$$

N est un exemple de l'ensemble d'entrainement et "t" est un cycle d'apprentissage.

Tableau 5 : Avantages et inconvénients de SGD

Avantages	Inconvénients
-Demande moins de mémoire que BGD, c'est-à-dire plus rapide que BGD. -L'algorithme converge en moins de temps.	-Les paramètres peuvent avoir une variance élevée. - Les paramètres peuvent varier même si le minimum de la fonction de coût est obtenu. - Le SGD ne peut jamais converger vers les minima de la fonction de coût ; il peut juste osciller dans sa région. -Pour obtenir la convergence du SGD, le taux d'apprentissage doit être réduit lentement. -Ne pas tirer profit de la vectorisation. -α est constant pendant l'apprentissage. -Ne peut pas s'adapter à de grands lots.

4-4-3- Descente de gradient en mini-batch (MGD)

L'algorithme de descente de gradient par mini-batch [104] permet de tirer parti des avantages des deux algorithmes précédents. Dans cet algorithme, l'ensemble de données est divisé en plusieurs lots et au lieu de mettre à jour les paramètres après avoir calculé le gradient pour tout l'ensemble de données, ils sont mis à jour après avoir calculé la perte de chaque lot passé par le réseau de neurones, c'est-à-dire que si nous divisons l'ensemble de formation en K lots, les paramètres seront mis à jour K fois en un cycle de formation.

La mise à jour des paramètres est donc la suivante :

$$W_B(t) = W_B(t-1) + \Delta W_B(t) \qquad 15$$

B est un lot de l'ensemble d'apprentissage et t est un cycle de formation.

Table 6 : Avantages et inconvénients de MGD

Avantages	Inconvénients
-Profiter des avantages de la vectorisation. -Réduire la variance élevée des paramètres dans SGD. -Nécessite une quantité moyenne de mémoire.	-Peut rester bloqué dans des minima locaux. -Si le taux d'apprentissage est petit, l'algorithme peut prendre beaucoup de temps pour converger. -α est constant pendant l'apprentissage. -Ne peut pas s'adapter à de grands lots.

4-5- Algorithmes d'optimisation par Descente de Gradient

4-5-1- Descente de gradient avec Momentum

L'idée de base de la descente de gradient avec Momentum [105] est de calculer une moyenne exponentielle (par exemple, le type de moyennes mobiles qui élimine les fluctuations transitoires dans le temps, en donnant plus de poids aux événements les plus récents et moins de poids aux plus anciens) du gradient, puis l'algorithme l'utilise pour mettre à jour les paramètres. La première chose que fait cet algorithme est de calculer un $V_{\Delta w}(t)$ de sorte que :

$$V_{\Delta w}(t) = \beta V_{\Delta w}(t-1) + (1-\beta)\Delta W(t) \quad 16$$

En outre, la mise à jour des paramètres sera la suivante :

$$W(t) = W(t-1) - \alpha V_{\Delta w}(t) \quad 17$$

Tel que t est un cycle de formation.

Tableau 7: Avantages et Inconvénients du Descente de Gradient avec Momentum

Avantages	Inconvénients
-Le momentum peut être utilisé par le BGD, le SGD et le MGD. -Accélérer BGD, SGD et MGD. -Réduire la variation et les oscillations.	- Un hyperparamètre est ajouté (β) qui doit être fixé manuellement. -Si β est trop élevé, l'algorithme peut manquer les minima locaux. -α est constant pendant l'apprentissage. -Ne peut pas s'adapter à de grands lots.

4-5-2- Gradient accéléré de Nesterov (NAG)

Le gradient accéléré de Nesterov [106], [107] est étroitement lié à l'algorithme du Momentum, la différence se situe dans : Au lieu de calculer le gradient pour le paramètre courant $W(t)$, le gradient est calculé pour un paramètre intermédiaire $\widetilde{W}(t)$ de sorte que :

$$\widetilde{W}(t) = W(t) - \beta V(t-1) \quad 18$$

Et

$$V(t) = \beta V(t-1) + \alpha \Delta \widetilde{W}(t) \quad 19$$

En outre, la mise à jour des paramètres sera la suivante :

$$W(t) = W(t-1) - V(t) \quad 20$$

Tel que t est un cycle de formation.

Tableau 8 : Avantages et Inconvénients du NGD

Avantages	Inconvénients
-Cet algorithme peut être utilisé par le BGD, le SGD et le MGD. -Accélérer BGD, SGD et MGD. -Réduire la variation et les oscillations. -Résoudre le problème des minima locaux présenté par le momentum.	-Un hyperparamètre est ajouté (β)qui doit être fixé manuellement. -α est constant pendant l'apprentissage. -Ne peut pas s'adapter à de grands lots.

4-5-3- Rprop

Dans [108], [109], les auteurs ont proposé un algorithme de taux d'apprentissage adaptatif c'est-à-dire que le taux d'apprentissage change au cours de l'apprentissage. Le principe de cet algorithme est de prendre en considération uniquement le signe du gradient sur l'ensemble des paramètres, et d'agir indépendamment sur chaque paramètre. L'idée de cet algorithme est la suivante :

- Si le gradient actuel a le même signe que la valeur précédente, plus formellement : $\frac{\partial c}{\partial w(t)} * \frac{\partial c}{\partial w(t-1)} > 0$ cela signifie que l'algorithme est dans la bonne direction, et la valeur du taux d'apprentissage doit augmenter pour aller rapidement vers la valeur optimale de la fonction de coût et la mise à jour sera :

$$W(t) = W(t-1) - \alpha \eta^{+} \Delta W(t) \quad 21$$

Tel que : $\eta^{+} = 1{,}2$

- Si le gradient actuel et la valeur précédente avaient des signes différents, plus formellement : $\frac{\partial c}{\partial w(t)} * \frac{\partial c}{\partial w(t-1)} < 0$ c'est-à-dire que l'algorithme vient de sauter par-dessus un optimum et que l'algorithme est dans la mauvaise direction. Donc, la mise à jour sera :

$$W(t) = W(t-1) - \alpha \eta^{-} \Delta W(t) \quad 22$$

Tel que : $\eta^{-} = 0{,}5$

Tel que t est un cycle de formation.

Tableau 9 : Avantages et Inconvénients du Rprop

Avantages	Inconvénients
-Le taux d'apprentissage a changé pendant l'opération de formation.	-Tenir compte uniquement du signe du gradient mais pas de sa magnitude et cela influence l'accélération de la formation. -Cet algorithme ne peut pas fonctionner pour les grandes données. -Il ne peut pas s'adapter à de grands lots.

4-5-4- ADAgrad

Le gradient adaptatif ou ADAgrad [110], [111] est une amélioration de l'algorithme Rprop, dans cet algorithme l'historique du gradient est accumulé pendant la formation dans une fonction V, de sorte que:

$$V(t) = V(t-1) + \left(\Delta W(t)\right)^2 \quad 23$$

La clé de cet algorithme est la fonction V, qui conserve les valeurs du gradient pendant la formation et les utilise pour mettre à jour les paramètres du réseau de neurone. Par conséquent, la mise à jour sera la suivante :

$$W(t) = W(t-1) - \frac{\alpha}{\sqrt{V(t) + \varepsilon}} \Delta W(t) \quad 24$$

Tel que ε est une petite valeur pour éviter la division par zéro et t est un cycle de formation.

Tableau 10 : Avantages et Inconvénients d'ADAgrad

Avantages	Inconvénients
-Le taux d'apprentissage a changé pendant l'opération de formation. -Fonctionne très bien avec les données à caractéristiques éparses. -Nécessite une quantité moyenne de mémoire.	-L'algorithme diminue progressivement le taux d'apprentissage, c'est-à-dire que si le gradient est petit, la valeur de "V" sera petite aussi, ce qui entraîne une grande valeur du taux d'apprentissage et un grand pas dans l'optimisation. -Impossible de s'adapter à de grands lots.

4-5-5- RMSprop

RMSprop [112], [113] est un algorithme à taux d'apprentissage adaptatif, et il est basé sur l'idée présentée par ADAgrad. Il a été développé pour résoudre la diminution radicale du taux d'apprentissage dans ADAgrad. Son principe est qu'au lieu de cumuler les valeurs de gradient, il utilise une moyenne mobile de sorte que :

$$V(t) = \beta V(t-1) + (1-\beta)\big(\Delta W(t)\big)^2 \quad 25$$

En outre, la mise à jour des paramètres sera :

$$W(t) = W(t-1) - \frac{\alpha}{\sqrt{V(t)+\varepsilon}}\Delta W(t) \quad 26$$

Tel que ε est une petite valeur pour éviter la division par zéro et t est un cycle de formation.

Tableau 11 : Avantages et Inconvénients d'RMSProp

Avantages	Inconvénient
-Le taux d'apprentissage a changé pendant l'opération de formation. -Solutionner la diminution radicale du taux d'apprentissage dans ADAgrad. -Fonctionne très bien pour la plupart des applications, en particulier avec des données à caractéristiques éparses.	-Coûteux en termes de calcul. -Ne pouvait pas s'adapter à de grands lots.

4-5-6- ADAM

L'optimiseur ADAM [114] est considéré comme l'un des algorithmes d'optimisation récents, c'est l'un des algorithmes les plus efficaces pour l'optimisation en utilisant la descente de gradient, il est considéré comme une combinaison entre RMSprop et ADAgrad. La première chose que fait cet algorithme est de calculer deux estimations adaptatives de sorte que :

La première estimation adaptative est :

$$V(t) = \beta_1 V(t-1) + \big(1-\beta_1\big)\Delta W(t) \quad 27$$

Et la deuxième estimation adaptative est :

$$S(t) = \beta_2 S(t-1) + \big(1-\beta_2\big)\big(\Delta W(t)\big)^2 \quad 28$$

Les auteurs suggèrent des valeurs par défaut pour les hyperparamètres d'ADAM de sorte que $\beta_1 = 0{,}9$ et $\beta_2 = 0.999$.

Le principal problème de cette estimation adaptative est que la $V(t)$ et $S(t)$ sont initialisés par 0, et cela rend leurs valeurs proches de 0 dans la première étape si β_1 et β_2 sont proches de 1. Par conséquent, pour résoudre ce problème, les auteurs avaient proposé d'utiliser l'abaissement en calculant $V'(t)$ et $S'(t)$ tel que :

$$V'(t) = \frac{V(t)}{1-(\beta_1)^t} \quad 29$$

Et

$$S^{'}(t) = \frac{S(t)}{1-(\beta_2)^t} \quad 30$$

Par conséquent, la mise à jour des paramètres sera la suivante :

$$W(t) = W(t-1) - \alpha \frac{V'(t)}{\sqrt{S'^{(t)} + \varepsilon}} \quad 31$$

Tel que ε est une petite valeur pour éviter la division par zéro et t est un cycle de formation.

Tableau 12 : Avantages et Inconvénients d'Adam

Advantages	Inconvénients
-Le taux d'apprentissage a changé pendant l'opération de formation. -Fonctionne très rapidement et converge plus vite que les autres algorithmes. -Fonctionne très bien pour toutes les applications, en particulier pour les données à caractéristiques éparses.	-Coûteux en termes de calcul. -Ne pouvait pas s'adapter à de grands lots.

4-5-7- LARS

En raison de l'augmentation de la quantité de données disponibles, la taille des lots commence à augmenter également, mais cela peut entraîner des instabilités pendant l'apprentissage. Par conséquent, pour résoudre ce problème, les auteurs proposent dans [115], [116] un nouvel algorithme appelé Layer-Wise Adaptive Rate scaling ou LARS. Cet algorithme détermine un taux d'apprentissage spécifique pour chaque couche du réseau, où la valeur du taux d'apprentissage est mise à jour à l'aide d'un coefficient ŋ (ŋ<1) appelé "coefficient de confiance". Dans cet algorithme, les auteurs tentent d'analyser la stabilité de l'apprentissage en mesurant le rapport entre la norme de la mise à jour du gradient et la norme des poids de la couche. De plus, si le rapport est trop élevé, les auteurs observent que la formation peut devenir instable, et si le rapport est trop petit, alors les poids ne changent pas assez rapidement.

$$\Lambda^l(t) = \alpha \frac{\left\|W^l(t)\right\|_2^2}{\|\Delta W^l(t)\|_2^2} \quad 32$$

Après, la moyenne pondérée $V(t)$ sera calculé de telle sorte que :

$$V(t) = \mu V(t-1) + \Lambda^l(t-1)\Delta W(t) \quad 33$$

Maintenant la mise à jour des paramètres sera :

$$W(t) = W(t-1) - V(t) \quad 34$$

Tel que, t est un cycle de formation.

Tableau 13 : Avantages et Inconvénients du LARS

Avantages	Inconvénients
- Le taux d'apprentissage a changé pendant l'opération de formation. -Fonctionne très rapidement et converge plus vite que les autres algorithmes.	- Coûteux en termes de calcul.

-Fonctionne très bien pour toutes les applications, en particulier pour les données à caractéristiques éparses. - Fonctionne très bien pour les grands lots (jusqu'à 32K).	

5- Les Réseaux de Neurones Récurrents

5-1- Définition

L'une des principales caractéristiques de tous les réseaux de neurones à propagation avant (Perceptron multi couche (Figure 15)), est qu'ils n'ont pas de mémoire, chaque entrée qui leur est présentée est traitée indépendamment, sans qu'aucun état ne soit conservé entre les entrées. Pour mener à bien ce problème, on fait appel à des couches récurrentes afin de former des réseaux de neurones récurrents (RNNs).

Les RNNs sont définis comme une classe de modèles d'apprentissage automatique supervisés ou l'ensemble de données à une nature séquentielle, constitues de neurones artificiels avec une ou plusieurs boucles de rétroaction, c'est-à-dire les RNNs sont un type des RdNs qui possède une boucle interne, l'état du RNN est réinitialisé entre le traitement de deux séquences différentes et indépendantes, de sorte qu'une séquence est considéré toujours comme un point de données unique : une entrée unique pour le réseau. Ce qui change, c'est que ce point de données n'est plus traité en une seule étape, mais que le réseau boucle en interne sur les éléments de la séquence. En général, ces réseaux sont constitués d'unités non linéaires, mais simples, capables de stocker, de mémoriser et de traiter des signaux complexes passés pendant de longues périodes. De cette façon, les RNNs peuvent apprendre le contexte temporel des séquences d'entrée, mapper une séquence d'entrée sur la séquence de sortie au pas de temps actuel et prédire la séquence au pas de temps suivant. Les unités spéciales dans la couche cachée avec des éléments de contrôles d'état "mémorisation / oubli" sont des caractéristiques des structures RNNs. Cette propriété est utile pour prédire les séries temporelles, en particulier pour apprendre leurs dépendances temporelles [59].

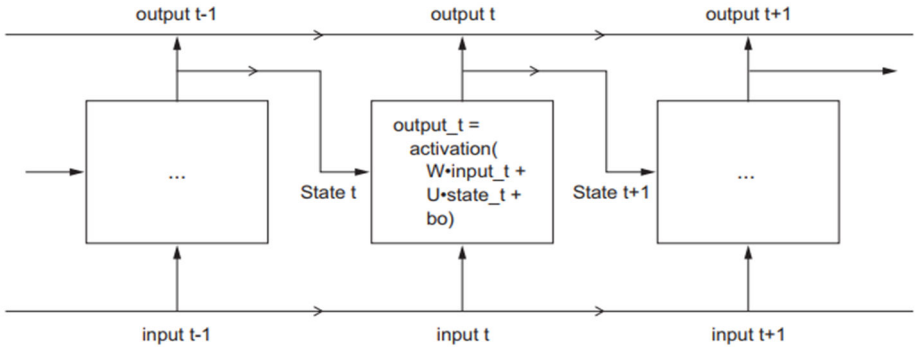


Figure 16: Propagation de l'information dans un couche d'RNN

Actuellement, diverses configurations des RNNs sont proposées sur la base de différentes unités dans des couches cachées, par exemple LSTM (Long Short Term Memory - une longue mémoire à court terme), GRU (Gated Recurrent Unit - unité récurrente fermée).

5-2- Long Short-Term Memory Network (LSTM)

Une unité récurrente LSTM essaie de ”se souvenir” de toutes les connaissances passées que le réseau a vues jusqu'à présent et d'oublier les données non pertinentes. Le principal avantage par rapport aux RNN traditionnels est le fait que les réseaux LSTM peuvent garder en mémoire les dépendances temporelles entre les différentes entrées pendant une période plus longue. Cela se fait en introduisant différentes couches de fonction d'activation appelées ”portes” à des fins différentes. Chaque unité récurrente LSTM maintient également un vecteur appelé l'état interne de la cellule qui décrit conceptuellement les informations qui ont été choisies pour être conservées par l'unité récurrente LSTM précédente. Un LSTM se compose de trois portes différentes à des fins différentes [59], comme d´écrit ci-dessous :

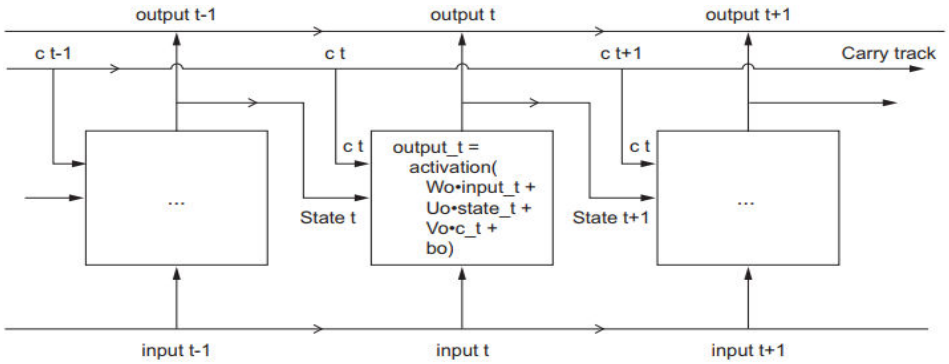


Figure 17 : Une cellule LSTM

Pour les couches LSTM, une nouvelle composante est ajoutée, ses valeurs à différents pas de temps ont appelé Ct, où Ct représente le report. Il sera combiné avec la connexion d'entrée et la connexion récurrente (via une transformation dense : un produit scalaire avec une matrice de poids suivi d'une addition de biais et de l'application d'une fonction d'activation), et il affectera l'état envoyé au prochain pas de temps (via une fonction d'activation et une opération de multiplication). Conceptuellement, le flux de données de report est un moyen de moduler la prochaine sortie et le prochain état. C'est un moyen d'oublier délibérément les informations non pertinentes dans le flux de données de report.

5-3- Gated Recurrent Unit (GRU)

GRU est essentiellement un LSTM simplifié, il a exactement le même rôle dans le réseau. La principale différence réside dans le nombre de portes et de poids, GRU est un peu plus simple. Comme il n'a pas de porte de sortie, il n'y a aucun contrôle sur le contenu de la mémoire. Les GRUs ont moins d'opérations tensorielles; par conséquent, ils sont un peu plus rapides à former que les LSTM. En Pratique, le LSTM ou GRU sont généralement utilisés, car un RNN simple est généralement trop simpliste pour être réellement utile [59].

6- Les Applications autour des Réseaux de Neurones

6-1- La classification

La classification est l'opération qui consiste à séparer diverses entités en plusieurs classes. Ces classes peuvent être définies par des règles de gestion, des limites de classe ou une fonction mathématique. L'opération de

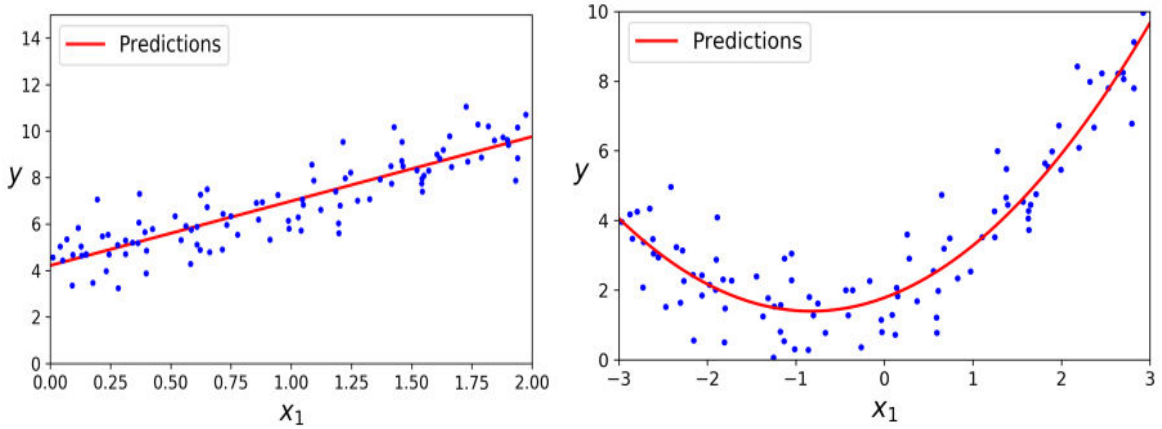
classification peut être basée sur une relation entre une affectation de classe connue et les caractéristiques de l'entité à classer, ce type de classification est dit supervisé. Si aucun exemple connu d'une classe n'est disponible, la classification est non supervisée. L'approche de classification non supervisée la plus courante est le 'Clustering'. Dans la pratique, il existe différents types de classification, parmi eux, on trouve :

- **Classification Binaire :** Fait référence aux tâches de classification qui ont deux étiquettes de classe.
- **Classification Multi-Class :** Se réfère aux tâches de classification qui ont plus de deux étiquettes de classe.
- **Classification Multi-Etiquette :** Se réfère aux tâches de classification qui ont deux ou plusieurs étiquettes de classe, où une ou plusieurs étiquettes de classe peuvent être prédites pour chaque exemple.
- **Classification Déséquilibrée :** Se réfère aux tâches de classification où le nombre d'exemples dans chaque classe est inégalement distribué.

6-2- La régression

La régression est l'application la plus classique des RdNs. La principale différence entre la régression et la classification est que les algorithmes de régression sont utilisés pour prédire les valeurs continues. Il permet la mise en relation d'un ensemble de variables d'entrée avec une variable de sortie et ainsi, il permet, après l'apprentissage, d'approximer à travers une sortie continue la valeur d'une variable en fonction des autres variables regroupées sous forme d'un vecteur d'entrée. Il y' a deux types de régression :

- **Régression Linéaire :** Dans ce cas, le réseau de neurone est entraîné afin de rechercher un vecteur de paramètres pour créer une relation linéaire entre la prédiction et les entrées (Figure 18-a).
- **Régression Non-Linéaire :** Dans ce cas, le réseau de neurones est entrainé afin d'établir une courbe pour représenter la relation entre deux variables au lieu d'une ligne droite utilisée par la régression linéaire (Figure 18-b).



a) Régression Linéaire b) Régression non-linéaire

Figure 18: Exemple d'une régression linéaire et régression non-linéaire

6-3- Diagnostic et Pronostic des SEDs par les RdNs

Les RdNs sont largement applicables aux problèmes réels et concrets des industries. Etant donné que les RdNs sont les plus aptes à identifier des modèles ou des tendances dans les données, ils conviennent bien aux problèmes de prédiction et de détection. En effet, ils ont déjà été appliqués avec succès dans les industries, notamment pour la prédiction des ventes, le Marketing ciblé, le contrôle des processus, le diagnostic et le pronostic des fautes, etc.

L'application des RdNs pour le diagnostic et le pronostic des systèmes industriels vise à exploiter les signaux issus et générés par des capteurs lors du fonctionnement actuel du système afin de donner une estimation sur l'état actuel et futur probable du système. L'utilisation des RdNs pour traiter ce type des problèmes peut être considérée à la fois comme un problème de classification, de sorte qu'on peut utiliser les modèles élaborer pour classifier le fonctionnement actuel du système à un fonctionnement normal ou fautif, ou de régression, qui permet de calculer des paramètres en relation avec le diagnostic et pronostic pour chaque faute possible. Dans le cadre de nos recherches, les évènements générés par les SEDs sont des séquences que nous pouvons les considérer comme des données temporelles. Ils seront présentés et analysés par des RdNs en temps réel, afin d'assurer un diagnostic et un pronostic en ligne. Cette opération est implémentée en deux étapes principales :

- $1^{ère}$ Étape : L'apprentissage, représenté dans la Figure 19, qui consiste à configurer, à partir d'une base de données, les paramètres des RdNs (Feed-Forward ou Récurrent) en fonction des relations existantes entre un ensemble de vecteurs d'entrée (Les événements générés par le SED) et de sortie (les paramètres de diagnostic ou de pronostic). Ceci nécessite la collecte des données du SED (mots temporisés enregistrés lors des cycles de fonctionnement de SED, c'est-à-dire l'historique de fonctionnement du SED ou le Modèle statistique du SED), les formater selon l'approche du diagnostic et de pronostic développer et les ranger dans des bases de données qui seront utilisées pour l'apprentissage.

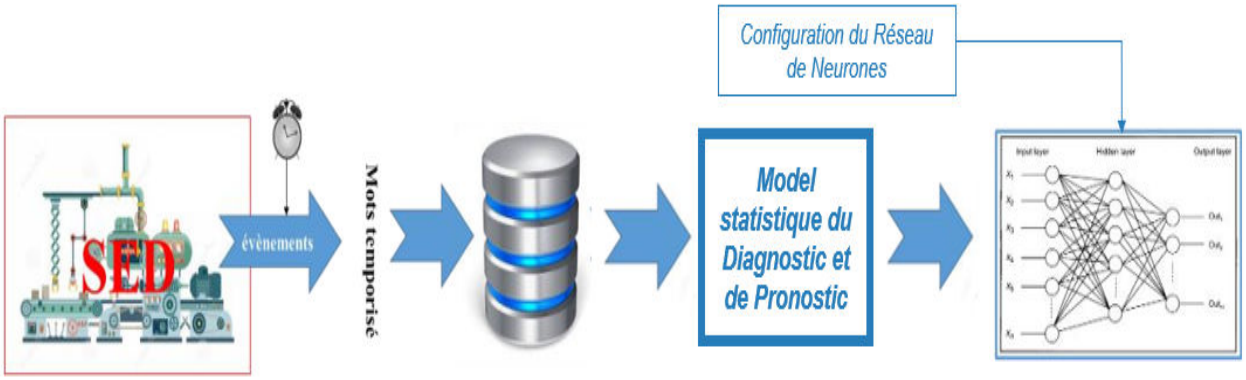


Figure 19: Exploitation des RdNs pour le Diagnostic et le Pronostic des SEDs 'Etape 1'

- $2^{ème}$ Étape : L'exploitation des RdNs configurés (Entrainés) lors de la phase d'apprentissage (Etape 1), pour traiter les informations reçues du SED afin de mettre en place un diagnostic/pronostic en ligne, dans cette étape, les événements générés par le SED en temps réel doivent être formaté selon l'approche développée dans l'étape 1 afin de former un vecteur d'entrée compatible pour les réseaux de neurones utilisés (une présentation détaillée de la manière de construire le vecteur d'entrée sera présentée dans les prochains chapitres de cette thèse.).

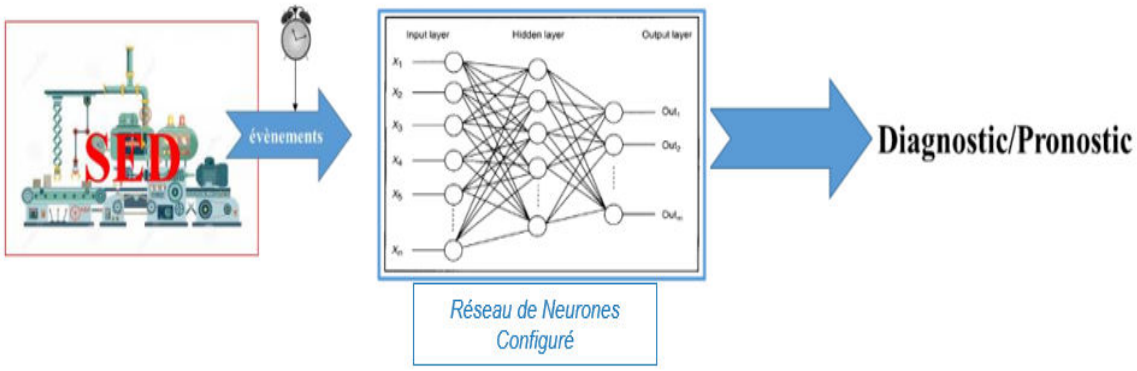


Figure 20: Exploitation des RdNs pour le Diagnostic et le Pronostic des SEDs 'Etape 2'

7- Conclusion

Dans ce chapitre, nous avons élaboré une vision générale sur les RdNs, après avoir introduit leurs concepts de base nous avons mis en évidence les différents algorithmes qui sont couramment utilisés pour accélérer l'apprentissage en utilisant la rétropropagation de la descente de gradient, qui peuvent être divisés en deux types : les algorithmes utilisant un taux d'apprentissage constant (BGD, SGD, MGD, ...etc.) et les algorithmes avec un taux d'apprentissage adaptatif (Rprop, RMSprop,...etc.), ainsi que les différentes architecture des RNNs. Eventuellement, nous avons clôturés ce chapitre par une vision globale sur les modes d'applications des RdNs, ainsi que la manière d'exploitation des RdNs pour assurer le diagnostic et le pronostic des SEDs.

Dans le chapitre suivant, nous proposons une nouvelle approche basée sur les RdNs à propagation avant (feed-forward), qui assurera les fonctions de diagnostic et du pronostic des fautes dans les SEDs, de sorte que les événements générés par le SED seront présentés et analysés par les RdNs en temps réel, afin d'effectuer un diagnostic et un pronostic en ligne.

Chapitre 3 : Diagnostic et Pronostic des SEDs par Réseaux de Neurones à Propagation Avant

Table des Matieres

1- Introduction 63
2- Classification des fautes dans les SEDs 63
3- Diagnostic et pronostic des Fautes à l'aide des RdNs feed-foorward 64
4- Diagnostic et Pronostic des Fautes 67
4-1- Diagnosticabilité 67
4-2- Pronosticabilité 69
4-3- Temps Restant avant une Faute 71
5- Construction des Réseaux de Neurones et Etude de Cas 72
5-1- L'Architectures du Réseau de Neurone 72
5-2- Apprentissage des Réseaux de Neurones 73
5-3- Ensemble de Données du Diagnostic 73
5-4- Ensemble de Données du Pronostic 74
5-5- Ensemble de Données du Temps Restant avant une Faute 74
5-6- Algorithme du Diagnostic et du Pronostic des SEDs par paire temporelle 74
5-7- Etude de Cas 76
5-6-1- Présentation du SED 76
5-6-2- Analyses des Modèles RBFNNs 77
6-Conclusion 83

1- Introduction

Dans les chapitres précédents un état de l'art concernant les méthodes du diagnostic et de pronostic des SEDs, ainsi qu'une vision générale sur les RdNs ont été présenté. Dans ce chapitre une nouvelle approche du diagnostic et de pronostic des SEDs sera illustrée. Précédée par une classification non exhaustive des fautes auxquels les SEDs sont sujets. Nous nous s'intéressons principalement aux SEDs larges et complexe, où l'utilisation d'outils de modélisation devient une tâche complexe et épuisante. L'objectif de ce chapitre est l'exploitation des réseaux de neurones feed-forward afin de déterminer l'état actuel probable du SED en détectant si un événement de faute $f_\alpha \in \sum_f$ est généré ou non, ainsi que d'identifier l'état futur probable du SED en détectant si un événement de faute peut-être généré dans le fonctionnement futur du SED ou non. Pour ce faire, une nouvelle approche et son cadre théorique, ainsi que les conditions suffisantes et nécessaires pour la diagnosticabilité et la pronosticabilité seront présentés, et sur la base des résultats obtenus, une base de données d'apprentissage pour former les réseaux de neurones afin d'obtenir les opérations souhaitées est construite. Enfin, et afin de déterminer et de prouver la pertinence de l'approche proposée, une étude de cas sur un système réel à événements discrets est présentée.

2- Classification des fautes dans les SEDs

Dans le contexte du Diagnostic et de Pronostic des SEDs, une faute est considérée comme un évènement dont l'occurrence est aléatoire et qui provoque une déviation du comportement nominal et normal vers un comportement défectueux, c'est-à-dire que les fautes sont tout type de déviation d'au moins une caractéristique, une propriété ou un paramètre du système par rapport à leur état de fonctionnement normal. Dans la littérature, les fautes sont classifiées en se basant sur deux critères principaux : 1- Le composant responsable de la faute, 2- la dynamique d'occurrence de la faute.

Concernant le composant responsable de la faute, il existe 3 catégories principales [14], [117]:

- **Fautes Actionneurs :** comprennent toutes les pannes liées à la défaillance d'un composant matériel : actionneur ou pré actionneur. Ces pannes comprennent également la destruction d'équipements qui assurent des mouvements spécifiques ou qui assurent la transmission du mouvement (roulement, engrenages, bandes transporteuses, etc.).

- **Fautes capteurs :** ce sont des fautes qui impliquent un capteur, une connexion de capteur au contrôleur ou la carte d'entrée du contrôleur. Ces pannes comprennent également les pannes résultant de la destruction des connexions qui relient les capteurs au contrôleur, ce type de faute se manifeste soit par l'incapacité du capteur à fournir des informations sur la grandeur physique à mesurer, soit par des informations erronées.

- **Fautes Processus :** Ce sont celles qui impliquent le processus et son environnement (la chute d'un colis ou un paramètre erroné par exemple), c'est-à-dire la partie opérationnelle en dehors des capteurs et de l'actionneur.

En ce qui concerne la dynamique d'occurrence de la faute, ils sont classés aussi en 3 catégories :

- **Fautes Permanentes :** Ils se distinguent par leur existence continue dans le temps après leur apparition. Par conséquent, le retour à un fonctionnement normal après ce type de fautes est indispensable sans intervention sur le composant.
- **Fautes Intermittents :** Ils arrivent et disparaissent de manière aléatoire et extrêmement rapide. Ils passent du fonctionnement normal au fonctionnement de faute plusieurs fois dans les deux sens. Ils peuvent être causés par des connexions défectueuses, l'usure des capteurs, etc.
- **Fautes Abrutes :** Ils s'agissent des fautes typiques d'usure ou d'encrassement d'un composant. Il est difficile à identifier en raison de son évolution temporelle, qui peut être confondue avec une modification paramétrique progressive exprimant la non-stationnarité du processus.

3- Diagnostic et pronostic des Fautes à l'aide des RdNs feed-foorward

L'approche principale pour analyser et exploiter les événements générés par un SED en utilisant les RdNs de type 'feed-forward', qui peut traiter en même temps le diagnostic et le pronostic, c'est la représentation des événements générés par le SED sous forme des fenêtres temporelles, c'est-à-dire une représentation spatio-temporelle [35] (cette technique peut être utilisée pour n'importe quelle architecture ou type de RdN 'feed-forward'). L'idée derrière cette approche c'est au lieu de présenter au RdN chaque événement généré par le SED dès qu'il se produit, il doit attendre qu'un certain nombre d'événements "q", qui dépendent nécessairement du problème traité, se soient produits, c'est-à-dire de retarder les événements d'un certain temps avant de les présenter au RdN et chaque retard temporel représente une dimension de la fenêtre temporelle, c'est-à-dire que le vecteur d'entrée sera composé par "q" événement avec un ensemble des retards temporels entre les évènements successifs, et pour chaque fenêtre temporelle présentée aux RdNs, les RdNs fournissent la probabilité qu'un événement de faute soit apparu dans la fenêtre temporelle pour le cas du diagnostic et la probabilité qu'un événement de faute puisse apparaître après l'apparition de la fenêtre temporelle en plus du temps restant avant une faute dans le cas du pronostic. Dans la littérature, Il y existe plusieurs recherches qui exploite cette technique, pour le traitement des données séquentielles, nous pouvons trouver celles présentées dans [118], [119] où les auteurs ont utilisé une représentation spatio-temporelle pour le traitement de la parole, détection des fraudes, prédiction des stocks et également d'autres recherches présentées dans [35], [120]. La figure suivante (figure 21) montre comment le diagnostic et le pronostic peuvent être réalisés en utilisant les fenêtres temporelles :

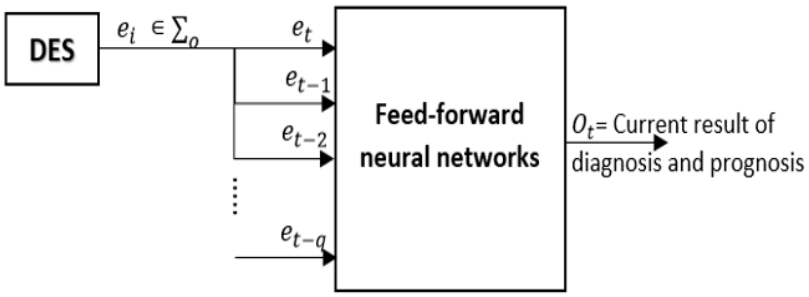


Figure 21: Diagnostic et Pronostic des Fautes à l'Aide des RdNs Feed-foorward

En pratique, l'utilisation de cette méthode présente plusieurs limites : Tout d'abord, pour assurer les fonctions de diagnostic et de pronostic, il est nécessaire de définir la taille de la fenêtre temporelle, ce qui va déterminer le nombre d'entrées des réseaux de neurones qui modifie directement l'architecture et la structure du RdN ainsi que ses performances et son temps de réponse, qui peut être très important [121]. De plus, si la taille de la fenêtre temporelle est grande, le diagnostic et le pronostic ne peuvent pas être opérationnels dès le début du fonctionnement du SED, car il faut attendre l'occurrence des "q" événements pour former une fenêtre temporelle et commencer le diagnostic et le pronostic. De plus, en utilisant des fenêtres temporelles, il n'est pas possible de déterminer la localisation exacte de l'occurrence d'un événement inobservable, c'est-à-dire de déterminer exactement entre quels événements observables un événement inobservable s'est produit, ce qui ne permet pas d'accomplir la deuxième phase du diagnostic, qui consiste à localiser exactement l'anomalie. De plus, si deux fenêtres temporelles sont plus ou moins similaires, les RdNs peuvent ne pas les distinguer et les traiter de la même manière [30], [88]. Par conséquent, dans ce chapitre, afin de résoudre les problèmes ci-dessus et d'atténuer ces limites, nous proposons une nouvelle approche, dont le but est qu'au lieu de diagnostiquer et de pronostiquer des fenêtres temporelles, nous utiliserons des paires temporelles, c'est-à-dire deux événements observables successifs. De manière à l'utiliser pour calculer la probabilité qu'un événement inobservable se soit produit entre deux événements observables successifs (diagnostic) et aussi pour calculer la probabilité qu'un événement inobservable puisse être généré pendant les prochains états de fonctionnement du SED (pronostic). En utilisant cette approche, nous aurons la possibilité de distinguer exactement l'endroit où un événement inobservable s'est produit. En outre, nous ne nous soucierons pas de la complexité de l'architecture du RdN, qui peut donner les meilleurs résultats, car en utilisant seulement deux événements comme entrés des RdNs, la taille des vecteurs d'entrée est beaucoup plus réduite, par rapport aux fenêtres temporelles, qui contiennent généralement un grand nombre d'événements, ce qui permet un entraînement plus efficace. En outre, nous aurons la possibilité de lancer le diagnostic et le pronostic à partir des deux premiers événements générés par le système. En ce qui concerne le problème de la similarité, qui peut également exister entre les paires temporelles, la solution proposée sera présentée et analysée en détail dans les sections suivantes.

La phase la plus importante de la construction d'un RdN est la phase d'apprentissage, au cours de laquelle un ensemble de données contenant les entrées et leurs sorties correspondantes est présenté au RdN. Par

conséquent, pour assurer le diagnostic et le pronostic des SEDs, la première étape consiste à construire un modèle statistique du SED, qui est considéré comme une base de données contenant la description de tous les comportements possibles (normaux et anormaux) du système souhaité sous la forme de données historiques pour chaque cycle de fonctionnement possible $\sigma^j \in TW^*$ dans les états normaux et de fautes. Tels que, σ^j est le $j^{ème}$ cycle de fonctionnement selon lequel le SED peut fonctionner.

Définition 5 (Une paire temporelle à partir d'un mot temporisé) : Nous considérons un mot temporisé $\sigma_{j'}^{j} \in TW^{j*}$ (tel que TW^{j*} l'ensemble de toutes les données historiques, qui concernent le cycle de fonctionnement j) tel que $\sigma_{j'}^{j} = e_1^{t_1} e_2^{t_2} \dots e_i^{t_i} \dots e_k^{t_k}$ enregistré à partir du $j'^{ème}$ cycle d'un SED fonctionnant selon le $j^{ème}$ cycle de fonctionnement et 'k' est le nombre d'événements observables. Une paire temporelle est un couple d'événements observables, c'est-à-dire deux événements successifs générés par le SED, plus formellement : $TP_{j'.i}^{j} = e_{i-1}^{t_{i-1}} e_i^{t_i}$ de telle sorte que $Tp_{j'.i}^{j}$ est le $i^{ème}$ paire temporelle dérivé d'un $\sigma_{j'}^{j}$, et contiennent deux événements observables successifs.

Étant donné Tp une paire temporelle, nous considérons les notations suivantes :

- $P_o(Tp)$: $Tp \to \sum_o$: La projection, qui élimine le temps et les événements non observables d'une paire temporelle dans le cas où ils existent.
- $P_u(Tp)$: $Tp \to \sum_f$: La projection qui ne conserve que les événements non observables d'une paire temporelle et supprime les événements observables et le temps.
- $T(Tp)$: $Tp \to \mathbb{R}^+$: La projection, qui donne la date d'occurrence de Tp.
- $S_{j'}^{j}$: L'ensemble de toutes les paires temporelles dérivées de $\sigma_{j'}^{j}$.
- $S_{j'f_\alpha}^{j}(Tp)$: L'ensemble de toutes les paires temporelles dérivées de $\sigma_{j'}^{j}$ et ont la même projection que Tp ainsi que contenir un événement de faute $f_\alpha \in \sum_f$, de manière plus formelle :

$$S_{j'f_\alpha}^{j}(Tp) = \{TP_{j'.i}^{j} \in S_{j'}^{j} / P_o(TP_{j'.i}^{j}) = Tp \; et \; P_u(Tp_{j'.i}^{j}) = f_\alpha\}$$

- $S_{j'}^{j}(Tp)$: L'ensemble de tous les paires temporelles dérivés d'un $\sigma_{j'}^{j}$, qui ont la même projection que Tp, de manière plus formelle :

$$S_{j'}^{j}(Tp) = \{TP_{j'.i}^{j} \in S_{j'}^{j} / P_o(TP_{j'.i}^{j}) = Tp\}$$

Exemple 3 : Nous considérons un SED défini par leurs ensembles $\sum = \{a, b, c, d, f_1, f_2, f_3, r_1, r_2\}$ répartis aux ensembles d'événements suivants $\sum_o = \{a, b, c, d\}$, $\sum_f = \{f_1, f_2, f_3\}$ et $\sum_{reg} = \{r_1, r_2\}$ tels que les événements f_1, f_2 et f_3 correspondent à des fautes, qu'ils conviennent de diagnostiquer et de pronostiquer. Nous considérons σ_1^1 un mot temporisé généré par le SED pendant le premier cycle suivant le premier cycle de fonctionnement, de sorte que :

$$\sigma_1^1 = b^1 a^{1,5} f_1^{2,3} c^4 b^5 f_2^{5,6} a^7 r_1^{7,3} d^{9,1} f_1^{9,3} c^{11} b^{12} a^{12,4} f_3^{12,9} c^{13,1} b^{14}$$

En appliquant les projections et la répartition des ensembles présentés nous obtenons :

$$S_1^1 = \left\{ \begin{array}{c} Tp_{1.1}^1 = b^1 a^{1,5}, Tp_{1.2}^1 = a^{1,5} f_1^{2,3} c^4, Tp_{1.3}^1 = c^4 b^5, Tp_{1.4}^1 = b^5 \\ f_2^{5,6} a^7, Tp_{1.5}^1 = a^7 r_1^{7,3} d^{9,1}, Tp_{1.6}^1 = d^{9,1} f_1^{9,3} c^{11}, Tp_{1.7}^1 = c^{11} b^{12} \\ Tp_{1.8}^1 = b^{12} a^{12,4}, Tp_{1.9}^1 = a^{12,4} f_3^{12,9} c^{13,1}, Tp_{1.10}^1 = c^{13,1} b^{14} \end{array} \right\}$$

Concernant $S_{j'}^j(Tp)$ et $S_{j'f_\alpha}^j(Tp)$, on prend $Tp = ba$ comme exemple, donc nous obtenons :

$$S_1^1(ba) = \{Tp_{1.1}^1, Tp_{1.4}^1, Tp_{1.8}^1\} \text{ et } S_{1f_2}^1(ba) = \{Tp_{1.4}^1\}.$$

Mais nous pouvons clairement observer qu'une paire temporelle Tp peut se produire plusieurs fois dans différentes phases lorsqu'un mot temporisé est généré par le SED (les inconvénients déjà discutés dans la section précédente), par conséquent, chacun d'entre eux doit être diagnostiqué et pronostiqué séparément, de cette façon, nous pouvons définir avec précision les différents comportements du SED. Pour cette raison, nous proposons de diviser chaque mot temporisé en sous-mots temporisés ou rangés (r) , de sorte que chaque sous-mot temporisé représente une phase de fonctionnement par rapport à un $\sigma_{j'}^j$ et cette partition sera faite de telle sorte qu'un sous-mot temporisé ne doit pas contenir des paires temporelles avec la même projection P_o.

Remarque 3 : L'indice (r) sera également une autre entrée du RdN, qui sera incrémentée automatiquement, en plus de l'indice de cycle de fonctionnement (j) et la paire temporelle actuelle, afin de donner au RdN la capacité de distinguer dans quelle phase et dans quel cycle de fonctionnement le SED fonctionne, et ce qui permet au RdN de présenter un comportement dynamique.

Pour cela, nous allons définir un nouvel ensemble $S_{j'.r}^j$, qui sera l'ensemble de toutes les paires temporelles dérivées d'un cycle $\sigma_{j'}^j \in TW^{j*}$ et appartenant au même sous-mot temporisé (r).

Exemple 4 : Considérons à nouveau σ_1^1 de l'exemple précédent, les sous-mots temporisés qui en sont dérivés seront les suivants :

$$S_{1.1}^1 = \{Tp_{1.1}^1 = b^1 a^{1,5}, Tp_{1.2}^1 = a^{1,5} f_1^{2,3} c^4, Tp_{1.3}^1 = c^4 b^5\},$$

$$S_{1.2}^1 = \{Tp_{1.4}^1 = b^5 f_2^{5,6} a^7, Tp_{1.5}^1 = a^7 r_1^{7,3} d^{9,1}, Tp_{1.6}^1 = d^{9,1} f_1^{9,3} c^{11}, Tp_{1.7}^1 = c^{11} b^{12}\},$$

$$S_{1.3}^1 = \{Tp_{1.8}^1 = b^{12} a^{12,4}, Tp_{1.9}^1 = a^{12,4} f_3^{12,9} c^{13,1}, Tp_{1.10}^1 = c^{13,1} b^{14}\}$$

4- Diagnostic et Pronostic des Fautes

4-1- Diagnosticabilité

Dans la littérature, il existe plusieurs définitions de la diagnosticabilité, la plus courante étant celle proposée par Sampath et al. [4], qui définissent la diagnosticabilité d'un SED comme la capacité de détecter avec un délai finit les occurrences de certains événements inobservables distingués, à savoir les événements de faute. Dans le cas des paires temporelles, nous définissons la diagnosticabilité comme la capacité à détecter

l'occurrence d'un événement d'une faute $f_\alpha \in \sum_f$ entre deux événements observables successifs et nous disons qu'une paire temporelle est diagnosticable si nous pouvons détecter avec certitude son occurrence. Ainsi, pour assurer cette opération, nous définissons un indice, similaire à celui défini dans [122], appelé indice de diagnosticabilité noté par $P_r^j(f_\alpha/Tp)$, qui est un indice statistique permettant de calculer la probabilité qu'un événement de faute $f_\alpha \in \sum_f$ se produit entre deux événements observables successifs générés par le SED, sur la base des données historiques et du modèle statistique recueillis pour le cycle de fonctionnement σ^j dans la phase de fonctionnement (r) du SED souhaité, de sorte que :

$$P_r^j(f_\alpha|Tp) = \frac{Nombre\ des\ TP_{j'.i}^j\ similaire\ a\ Tp\ contenant\ la\ faute\ f_\alpha\ dans\ le\ rang\ "r"}{Nombre\ des\ TP_{j'.i}^j\ similaire\ a\ Tp\ dans\ le\ rang\ "r"} \quad 35$$

Plus Formellement :

$$P_r^j(f_\alpha|Tp) = \frac{\sum_{j'} card\left(S_{j'f_\alpha}^j(Tp) \cap S_{j'.r}^j\right)}{\sum_{j'} card\left(S_{j'}^j(Tp) \cap S_{j'.r}^j\right)} \quad 36$$

Où : $S_{j'f_\alpha}^j(Tp) \cap S_{j'.r}^j$: est l'ensemble de toutes les paires temporelles similaires à Tp et appartenant au sous-mot temporisé r et contenant l'événement de faute f_α , et $S_{j'}^j(Tp) \cap S_{j'.r}^j$ est l'ensemble de toutes les paires temporelles similaires à Tp qui appartiennent au sous-mot temporisé r tel que :

$$0 \leq P_r^j(f_\alpha/Tp) \leq 1$$

Et on dit que Tp est diagnosticable par rapport à l'événement de faute f_α si : $P_r^j(f_\alpha|Tp) = 1$ ou $P_r^j(f_\alpha|Tp) = 0$.

Preuve 1 :

- Dans le cas ou $P_r^j(f_\alpha/Tp) = 1$:

$$P_r^j(f_\alpha|Tp) = 1 \Leftrightarrow \sum_{j'} card\left(S_{j'f_\alpha}^j(Tp) \cap S_{j'.r}^j\right) = \sum_{j'} card\left(S_{j'}^j(Tp) \cap S_{j'.r}^j\right)$$
$$\Leftrightarrow \forall \sigma_{j'}^j \in TW^{j*}, \forall Tp_{j'.i}^j \in S_{j'.r}^j : P_o\left(Tp_{j'.i}^j\right) = Tp \Rightarrow P_u\left(Tp_{j'.i}^j\right) = f_\alpha$$

Dans ce cas, on peut dire que Tp est diagnosticable par rapport à f_α et que l'occurrence de la faute f_α est certaine.

- Dans le cas ou $P_r^j(f_\alpha|Tp) = 0$:

$$P_r^j(f_\alpha|Tp) = 0 \Leftrightarrow \sum_{j'} card(S_{j'f_\alpha}^j(Tp) \cap S_{j'.r}^j) = 0 \Leftrightarrow \bigcup_{j'} S_{j'f_\alpha}^j(Tp) \cap S_{j'.r}^j = \Phi \Leftrightarrow \forall \sigma_{j'}^j \in TW_{j'}^{j*}, \forall Tp_{j'.i}^j$$
$$\in S_{j'.r}^j : P_o(Tp_{j'.i}^j) = Tp \Rightarrow P_u(Tp_{j'.i}^j) \neq f_\alpha$$

Par conséquent, Tp est diagnosticable par rapport à f_α et il ne contient certainement pas l'événement de faute f_α , cependant, il peut contenir un autre événement de faute. Dans ce contexte, nous pouvons dire qu'une paire

temporelle Tp est diagnosticable par rapport à tous les événements de faute ou à l'état normale si et seulement si :

$$\sum_{\alpha} P_r^j(f_\alpha|Tp) = 0$$

Car :

$$\sum_{\alpha} P_r^j(f_\alpha|Tp) = 0 \Leftrightarrow \forall\, f_\alpha \in \sum_f : P_r^j(f_\alpha|Tp) = 0$$

C'est-à-dire qu'aucun événement $f_\alpha \in \sum_f$ ne s'est produit et que le SED est dans son comportement normal. En outre, si toutes les paires temporelles sont diagnosticables, le cycle de fonctionnement σ^j est dit diagnosticable, et le SED est diagnosticable si $\forall \sigma^j \in TW^*$: σ^j est diagnosticable. Sinon, l'occurrence d'un événement de faute f_α est incertaine et la probabilité que le SED dévie vers un mode anormal α est expectative.

Exemple 5: Considérons σ_1^1 (comme dans l'exemple précédent), σ_2^1 et σ_3^1 trois mots temporisés collectés pendant le cycle de fonctionnement σ^1, tel que: $\sigma_2^1 = b^1 f_2^{1,3} a^{4,2} c^{4,8} b^5 a^{5,5} d^{6,1} f_1^{6,2} c^{6,9} b^{7,9} a^{8,3} b^{10}$ et $\sigma_3^1 = b^1 a^{1,5} c^{2,1} b^{2,3} f_1^{2,5} a^{2,8} d^3 c^4 r_2^{4,1} b^5 a^{5,4} b^7$.

Soit $Tp = ba$ la première paire temporelle générée par le SED suivant le cycle de fonctionnement σ^1, en formant les différents ensembles requis pour chaque mot temporisé on obtient : $P_1^1(f_1|ba) = 0\%$; $P_1^1(f_2|ba) = 33{,}33\%$; $P_1^1(f_3|ba) = 0\%$; $\sum_\alpha P_r^j(f_\alpha|Tp) \neq 0$. Par conséquent, la paire temporelle "ba" ne peut être diagnostiquée avec certitude et la probabilité que le SED dévie vers l'état de faute f_2 est $33{,}33\%$.

Cette définition est similaire à celle présentée dans [35], cependant dans [35], les auteurs ont utilisé des fenêtres temporelles au lieu de paires temporelles, de plus, ils ne font pas de distinction entre les fenêtres temporelles similaires et ils les diagnostiquent de la même manière même si elles appartiennent à des phases différentes du cycle de fonctionnement, cependant, dans notre approche et grâce à la définition proposée nous atténuons en même temps les inconvénients présentés dans la dernière section, et grâce aux paramètres "r", et "j" ajoutés nous avons résolu le problème de similarité entre les paires temporelles, ce qui nous a permis de diagnostiquer chacune d'entre elles indépendamment des autres et cela sera prouvé dans l'étude de cas, qui sera présenté dans la dernière partie de ce chapitre.

4-2- Pronosticabilité

Dans le domaine des SEDs, la pronosticabilité est la capacité à être pronostiquée, c'est-à-dire la possibilité de prédire si un événement de faute $f_\alpha \in \sum_f$ se produira dans les prochains états de fonctionnement du SED. A cet égard, nous définissons un autre indice (comme nous l'avons déjà fait pour la diagnosticabilité), qui est un paramètre statistique appelé pronosticabilité noté par $Pg_r^j(f_\alpha/Tp)$, qui calcule la probabilité qu'un événement de faute $f_\alpha \in \sum_f$ se produira dans l'avenir proche du fonctionnement du SED.

La probabilité d'occurrence d'un événement de faute $f_\alpha \in \sum_f$ dans les paires temporelles suivantes étant donné une paire temporelle Tp est équivalente à la probabilité d'occurrence de cette faute dans la première paire temporelle suffixe de Tp et contenant l'événement de faute f_α. Cette définition est similaire à celle proposée pour la diagnosticabilité, qui calcule la probabilité d'occurrence de chaque événement de faute par rapport à chaque paire temporelle Tp générée par le SED. Ainsi, cette notion peut être utilisée pour calculer la valeur de la pronosticabilité, de telle sorte que la pronosticabilité d'un événement de faute $f_\alpha \in \sum_f$ donnant Tp, sera égale à la valeur de la diagnosticabilité par rapport au premier suffixe du pair temporelle de Tp et contiendra la faute f_α. Par conséquent, nous allons définir un nouvel ensemble appelé $Suf_r^j(Tp/f_\alpha)$ regroupement tous les suffixes de la première paire temporelle de Tp appartenant à un sous-mot temporel "r" pour tous les ensembles d'apprentissage collectés pour un cycle de fonctionnement σ^j, donc, nous définissons $Suf_r^j(Tp/f_\alpha)$ comme suit :

$$Suf_r^j(Tp|f_\alpha) = \{next(Tp')/P_u(Tp') = f_\alpha \text{ et } T(Tp') \succ T(Tp)\}$$

Par conséquent, à partir de l'ensemble des données collectées pour un cycle de fonctionnement σ^j, nous calculons la pronosticabilité comme suit :

$$Pg_r^j(f_\alpha|Tp) = Maximum\ des\ indéxes\ de\ diagnosticabilité\ des\ Tp\ regroupé\ dans\ Suf_r^j(Tp|f_\alpha) \quad 37$$

Plus formellement :

$$Pg_r^j(f_\alpha|Tp) = Max\left(P_{r'}^j\left(Suf_r^j(f_\alpha|Tp)\right)\right) \quad 38$$

Tel que $P_{r'}^j(Suf_r^j(f_\alpha/Tp))$ sont les valeurs de diagnosticabilité des premières paires temporelles suffixe collectées pour un Tp appartenant à un sous-mot temporisé, de sorte que $r' \geq r$. En plus, la fonction $Max(P_{r'}^j(Suf_r^j(f_\alpha/Tp)))$ prend le maximum de ces valeurs, afin de fournir la probabilité qu'un événement de faute f_α se produira au cours des prochains états. Considérons f_α un événement de faute ; une paire temporelle Tp est pronosticable par rapport à f_α si :

$$card\left(Suf_r^j(Tp/f_\alpha)\right) = card\left(TW_{j'}^{j*}\right) \text{ ou } card\left(Suf_r^j(Tp/f_\alpha)\right) = 0$$

Preuve 2 :

- Pour le premier cas :

  $card(Suf_r^j(Tp/f_\alpha)) = card(TW_{j'}^{j*}) \Leftrightarrow \forall \sigma_{j'}^j \in TW_{j'}^{j*}, \forall Tp_{j',i}^j \in S_{j',r'}^j, \exists Tp' \in S_{j',r'}^j : Suf_r^j(Tp/f_\alpha) = Tp'$ tel que r' ≥ r.

C'est-à-dire qu'il existe une paire temporelle Tp' suffixe de Tp , qui contient certainement l'événement de faute f_α c'est-à-dire qu'il y a une très forte probabilité que f_α se produise dans les prochains états de fonctionnement, donc il faut être vigilant car le système peut dévier vers un mode de faute pendant les prochains états de fonctionnement à tout moment.

- Pour le deuxième cas :

$card(Suf_r^j(Tp/f_\alpha)) = 0 \Leftrightarrow \forall \sigma_{j'}^j \in TW^{j*}, \forall Tp_{j'.i}^j \in S_{j'.r}^j, \nexists Tp' \in S_{j'.r'}^j : Suf_r^j(Tp/f_\alpha) = \phi$ tel que r' $\geq$ r.

C'est-à-dire qu'il n'existe pas de suffixe de la paire temporelle Tp' qui contient l'événement de faute f_α, cependant, elle peut en contenir un autre.

Si $\bigcup_\alpha Suf_r^j(Tp/f_\alpha) = \phi$, c'est à dire $\forall \sigma_{j'}^j \in TW^{j*}, \forall Tp_{j'.i}^j \in S_{j'.r}^j, \forall f_\alpha \in \sum_f, \nexists Tp' \in S_{j'.r}^j : Suf_r^j(Tp/f\alpha) = \phi$.

C'est-à-dire qu'il n'y a pas de pair temporelle Tp' suffixe de la paire temporelle Tp, qui contient un quelconque événement de faute f_α et le SED sera dans son comportement normal. De plus, si toutes les paires temporelles sont pronosticables, le cycle de fonctionnement σ^j est dit pronostiquable, et le SED est pronosticable si et seulement si $\forall \sigma^j \in TW^* : \sigma^j$ est pronosticable.

Exemple 6 : Considérons à nouveau σ_1^1, σ_2^1 et σ_3^1les trois mots temporisés de l'exemple 5 et prenons $Tp = ba$ la première paire temporelle générée par le SED suivant le cycle de fonctionnement σ^1, et on considère l'événement de faute f_1 comme exemple :

$$Suf_1^1(b^1a^{1,5}|f_1) = \{Tp_{1.6}^1 = d^{9,1}f_1^{9,3}c^{11}, Tp_{2.6}^1 = d^{6,1}f_1^{6,2}c^{6,9}, Tp_{3.4}^1 = b^{2,3}f_1^{2,5}a^{2,8}\}$$

En calculant la pronosticabilité de chaque paire temporelle, on obtient : $Pg_1^1(f_1/ba) = 66{,}66\%$, c'est-à-dire qu'il y a une probabilité qui peut même aller jusqu'à 66,66% que l'événement de faute f_1 se produise dans les prochains états.

4-3- Temps Restant avant une Faute

En calculant le temps restant avant une faute, les utilisateurs du SED pourront avoir une estimation sur le temps qui reste avant qu'un événement de faute f_α se produise probablement dans les prochains états de fonctionnement du SED. Par conséquent, et grâce à la présentation par mots temporelles, qui identifie chaque événement par sa date d'occurrence, nous pouvons facilement extraire leurs dates d'occurrence en utilisant la projection $T(Tp)$. Afin d'accomplir cette fonction, nous allons définir un nouvel indice $t_r^j(f_\alpha/Tp)$, qui fournira l'intervalle de temps, où un événement de faute f_α peut probablement se produire dans les étapes de fonctionnement à venir de sorte que :

$$t_r^j(f_\alpha/Tp) = \begin{cases} -1 \text{ si } Pg_r^j(f_\alpha/Tp) = 0 \\ [min\{T_{f_\alpha}(Suf_r^j(Tp/f_\alpha))\} - T(Tp); \\ max\{T_{f_\alpha}(Suf_r^j(Tp/f_\alpha))\} - T(Tp)]; \text{ si } Pg_r^j(f_\alpha/Tp) \succ 0 \end{cases} \quad (39)$$

où : $T(Tp)$ est la date d'occurrence de l'événement courant (deuxième événement de Tp) généré par le SED et $T_{f_\alpha}(Suf_r^j(Tp/f_\alpha))$ est l'ensemble de toutes les valeurs probables qui représentent le temps restant estimé de l'occurrence de l'événement de faute f_α, au sein des paires temporelles, qui composent $Suf_r^j(Tp/f_\alpha)$. Et la fonction $t_r^j(f_\alpha/Tp)$ prend le maximum et le minimum de cet ensemble (Dans le cas où l'occurrence de cet événement est probable c'est-à-dire $Pg_r^j(f_\alpha/Tp) \succ 0$)).

Remarque 4 : Si $min\{T_{f_\alpha}(Suf_r^j(Tp/f_\alpha))\} - T(Tp) \prec 0$ on pose : $min\{T_{f_\alpha}(Suf_r^j(Tp/f_\alpha))\} - T(Tp) = 0$.

Exemple 7: Considérons à nouveau les mêmes mots temporelles que ceux utilisés pour calculer la diagnosticabilité et la pronosticabilité pour les exemples 5 et 6 et considérons que $Tp = ba$ la première paire temporelle générée par le SED suivant le cycle de fonctionnement σ^1 nous supposons $T(ba) = 1{,}5$ et prenons l'événement de faute f_1 comme exemple: on obtient $T_{f_1}(Suf_1^1(ba/f_1)) = \{9{,}3; 6{,}2; 2{,}5\}$. Par conséquent, $t_1^1(f_1/ba) = [2{,}5 - 1{,}5; 9{,}3 - 1.5] = [1; 7{,}8]$ c'est-à-dire que l'événement de faute f_1 peut probablement se produire après 1 jusqu'à 7,8 unités de temps.

5- Construction des Réseaux de Neurones et Etude de Cas

L'objectif de cette section est de présenter les éléments constitutifs des modèles de RdNs afin de réaliser les opérations présentées ci-dessus, et cela inclut l'architecture de RdN choisie et la construction de la base de données pour les entrainés. Pour notre connaissance, trois RdNs doivent être développés, chacun assurant l'une des opérations présentées : le diagnostic, le pronostic et le calcul du temps restant avant la faute.

5-1- L'Architectures du Réseau de Neurone

Selon [3], une étude comparative entre plusieurs architectures de RdNs de type 'feed-forward' a été présentée. Le résultat de cette étude montre que le réseau de neurones à fonction de base radiale (RBFNN) offre certains avantages, qu'une autre architecture ne peut pas prendre en charge, notamment en ce qui concerne la précision. Le RBFNN [123], [124] est un RdN de type feed-forward avec une seule couche cachée en plus des couches d'entrée et de sortie ; chacune d'entre elles est entièrement connectée à la suivante. La couche cachée est formée de nœuds (nœuds RBFNN) et pour chaque nœud, une fonction d'activation gaussienne est associée. Deux paramètres importants sont associés μ (centre ou prototype) et Ω le rayon d'influence, qui confère au RBFNN une capacité à chevaucher des régions localisées. La figure 22 illustre l'architecture du RBFNN. Ainsi, le RBFNN sera utilisé pour calculer les différents paramètres présentés ci-dessus.

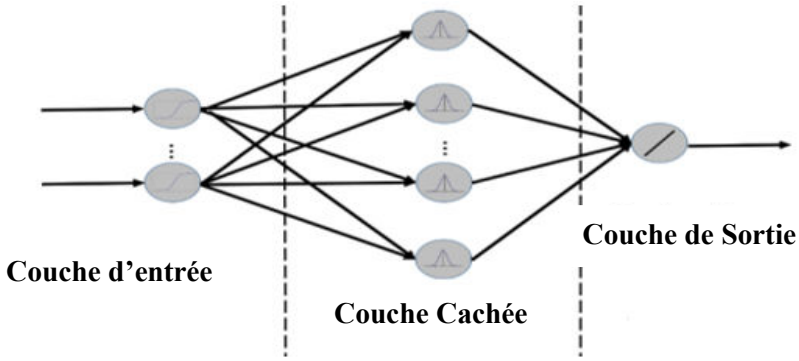


Figure 22: L'architecture du RBFNN

5-2- Apprentissage des Réseaux de Neurones

Comme nous avons déjà vu, la phase d'apprentissage des RdNs est la phase la plus importante de la construction d'un RdN, qui consiste à établir une relation entre les entrées et leurs sorties correspondantes. Dans notre cas, le jeu de données sera construit par un ensemble de la séquence d'événements générés par le SED pour les différents cycles de fonctionnement, qui doivent être organisés en mots temporelles puis en paires temporelles, des différents comportements du SED souhaité : Normal et Fautif. Les entrées (événements) sont considérées comme des données catégorielles, cependant, la plupart des algorithmes d'apprentissage profond ne peuvent pas opérer directement sur des données catégorielles, de sorte qu'ils exigent que toutes les variables d'entrée soient numériques, c'est-à-dire que les événements présentés au RdN doivent être convertis sous forme numérique, à cette fin, nous suggérons d'utiliser l'encodage à un coup de sorte que chaque événement sera encodé pour le transformer en vecteurs de 0 et de 1, de sorte qu'il change l'événement en $1 \times N(N = card(\sum_o))$ vecteur dimensionnel qui serait tous des 0 sauf pour l'index de l'événement, qui serait 1. Plus d'informations sur le codage à un coup est présenté dans [125].

5-3- Ensemble de Données du Diagnostic

En utilisant les RdNs pour traiter l'opération du diagnostic, le modèle est développé pour fournir une approximation de la valeur de diagnostic pour chaque paire temporelle générée par le SED en temps réel, à cette fin, chaque exemple d'entraînement utilisé pour entraîner le RBFNN dédié au diagnostic doit être comme suit :

$$TE_r^j = \begin{bmatrix} P_o(Tp) \\ j \\ r \end{bmatrix} \quad P_r^j(Tp)$$

Tel que : $P_o(Tp) \in \{0;1\}^{1\times 2N}$ est la matrice de lignes formées par l'utilisation d'un codage à un coup composé de l'événement observable de Tp et $(r,j) \in (\mathbb{R}^+)^2$ représente le sous-mot temporisé et le cycle de fonctionnement où appartient Tp, et cette matrice représente le vecteur d'entrée. En outre, $P_r^j(Tp) = [\cdots\cdots P_r^j(f_\alpha/Tp)\cdots\cdots] \in [0,1]^{1\times p}$ qui contiennent les probabilités d'occurrence ou les valeurs de diagnostic de tous les événements de faute appartenant à $\sum_f$, représentent le vecteur de sortie. Ainsi, le RBFNN avec $(2N+2)$ entrées et p sorties représentent le diagnostiqueur.

Remarque 5 : En pratique, le cycle de fonctionnement du SED est mentionné soit par les utilisateurs du SED, soit l'occurrence d'un événement prédéterminé permet de déterminer à quel cycle de fonctionnement le SED opère.

5-4- Ensemble de Données du Pronostic

En utilisant les RdNs pour assurer le pronostic, le modèle est développé pour fournir une approximation de la valeur de pronostic pour chaque paire temporelle générée par le SED en temps réel, à cette fin, chaque exemple de formation utilisé pour former le RBFNN dédié au pronostic doit être le suivant :

$$TE_r^j = \begin{bmatrix} P_o(Tp) \\ j \\ r \end{bmatrix} \quad Pg_r^j(Tp)$$

Tel que : $P_o(Tp), j,$ et r sont les mêmes que ceux que nous avons déjà définis dans la base de données du diagnostic et $Pg_r^j(Tp) = [\cdots\cdots Pg_r^j(f_\alpha/Tp)\cdots\cdots] \in [0,1]^{1\times p}$ qui contiennent les valeurs de pronosticabilité de tous les événements de faute appartenant à $\sum_f$, représentent le vecteur de sortie. Par conséquent, le RBFNN avec $(2N+2)$ entrées et p sorties représente le pronostiqueur.

5-5- Ensemble de Données du Temps Restant avant une Faute

Comme nous l'avons déjà fait pour le diagnostiqueur et le pronostiqueur, nous allons faire la même chose pour le modèle dédié au calcul du temps restant avant l'occurrence d'un événement de faute f_α. Telle que la matrice d'entrée sera la même que celle du diagnostiqueur et du pronostiqueur. Néanmoins, $t_r^j(Tp) = [\cdots\cdots t_r^j(f_\alpha/Tp)\cdots\cdots] \in (\Re^+)^{1\times(2p)}$ qui contiennent le temps restant avant l'apparition d'un événement de faute f_α est le vecteur de sortie. Par conséquent, le RBFNN avec $2N$+2 en entrée et $2p$ en sortie représentera le calculateur du temps restant avant une faute.

5-6- Algorithme du Diagnostic et du Pronostic des SEDs par paire temporelle

Après la phase de prétraitement des données d'apprentissage (Modèle statistique du SED formaté en paires temporelles) ainsi que l'entrainement des RdNs. La mise en place d'un module de diagnostic et de pronostic en ligne nécessite la programmation d'une application informatique que nous pouvons décrire par l'algorithme 1 qui sera capable d'assurer d'une façon automatique les différentes opérations de diagnostic et de pronostic de n'importe quel SED.

Algorithme 1

Input : Tp, j, r

Output : $P_r^j(Tp/f_\alpha), Pg_r^j(Tp/f_\alpha), t_r^j(Tp/f_\alpha)$

Step 1 : $j = 0; r = 1; i = 1; Tp_i^r = []; S_r^j = \{\}$

Step 2 : Update j

Step 3 : if event.available () == True

e_i= event.collect()

Tp_i^r. add(e_i)

if lenght (Tp_i^r)<2

Padding(Tp_i^r, 1)

else

Tp_i^r.delete_first_element ()

Tp_i^r. Concatenat (e_i)

Compare (Tp_i^r) with S_r^j.elements ()

if $Tp_i^r \in S_r^j$

$Tp_i^r = Tp_i^{r+1}$

r++

Tp_i^r=Tp_i^r.apply_one_hot_encodding()

else

S_r^j.add(Tp_i^r)

continue

else : go to Step 1

Step 4 : EV= $[Tp_i^r; j; r]$

$P_r^j(Tp/f_\alpha) = diag_NN(EV)$

$Pg_r^j(Tp/f_\alpha) = diag_NN(EV)$

$t_r^j(Tp/f_\alpha) = diag_NN(EV)$

Go to Step 2

Figure 23: Algorithme du Diagnostic et du Pronostic des SED par paire temporelle

Notre algorithme prend en entrée une paire temporelle, le cycle de fonctionnement et le rang de fonctionnement et fournit comme sorti les différents indices du diagnostic et de pronostic. Cet algorithme se compose de quatre étapes principales :

- **Etape 1 :** L'initialisation du cycle de fonctionnement par 0 car nous ne connaissons pas encore suivant quel cycle de fonctionnement le SED va opérer, l'initialisation de l'indice r et i par 1 et de Tp_i^r, S_r^j respectivement par une liste vide et un ensemble vide.
- **Etape 2 :** Le cycle j est soit réglé manuellement, soit un événement prédéterminé peut nous donner la possibilité de savoir dans quel cycle le SED fonctionne. En général, la mise à jour de j sera assurée par un algorithme indépendant qui s'appuie directement sur le SED étudié.
- **Etape 3 :** Cette étape consiste en différentes sous étapes, au premier temps le module du diagnostic et de pronostic vérifie si le SED a généré un événement observable ou non. Dans le cas de l'occurrence d'un événement observable il sera récupéré directement dans e_i après il sera ajouté a Tp_i^r. Dans le cas ou Tp_i^r ne dispose pas de deux événements, la fonction padding, qui permet d'ajouter un événement

vide aTp_i^r, sera appliqué. Dans le cas contraire, l'ancien événement de Tp_i^r c'est-à-dire e_{i-1}sera enlevé et le nouveau sera ajouté. Après, une phase importante de comparaison entre la praire temporelle actuelle et les paires temporelles de S_r^j afin d'incrémenter l'indice de rang r ou non.

- **Etape 4 :** C'est la phase finale de notre algorithme, qui consiste au calcul des indices : Diagnosticabilité, Pronosticabilité, et le temps restant avant une faute**.**

5-7- Etude de Cas

Dans cette section et afin d'illustrer la pertinence de l'approche développée, nous allons présenter une étude de cas, où nous essayerons de mettre en pratique le cadre théorique présenté dans les sections précédentes.

5-6-1- Présentation du SED

Considérons un SED opérationnel adopté de [35], ce système concerne un modèle de chariot suspendu (Figure 24) disponible au laboratoire de technique industrielle, faculté des sciences et techniques, université sidi Mohammed Ben Abdullah, Fès, Maroc. La fonction principale de ce système est de transférer chacun des cinq types de produits (A, B, C, D, et E) vers leurs propres stocks (S_A, S_B, S_C, S_D et S_E). Ce système dispose d'un lecteur de codes-barres qui distingue les différents produits. Nous supposons que $\sum_o = \{e_o, e_A, e_B, e_C, e_D, e_E, e_{PA}, e_{PB}e_{PC}, e_{PD}, e_{PE}, e_{SA}, e_{SB}, e_{SC}, e_{SD}, e_{SE}, e_{CA}, e_{CB}, e_{CC}, e_{CD}, e_{CE}, e_H\}$, $\sum_f = \{F_0, F_1, F_2, F_3\}$ et $\sum_r = \{r_1, r_2, r_3, r_4, r_5\}$ sont respectivement les ensembles d'événements observables, de fautes et d'événements réguliers. Le tableau 14 présente tous les événements et leur désignation.

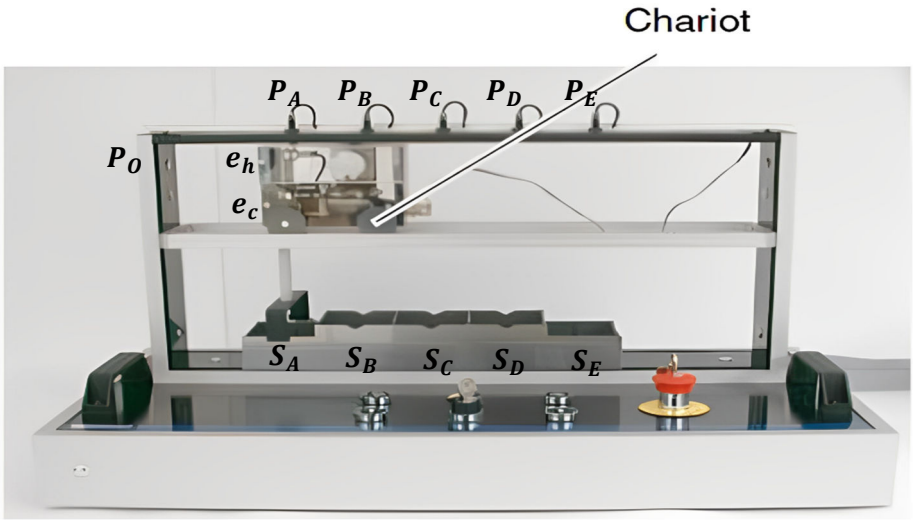


Figure 24: Le Chariot suspendu

Le chariot suspendu, et selon le type de produit détecté, peut exécuter un des cycles de fonctionnement suivants (Ces cycles représentent le fonctionnement normal du SED) :

- Pour le Produit A : $\sigma^1 = e_A^{t_1} e_{P_A}^{t_2} e_{C_A}^{t_3} e_{S_A}^{t_4} e_H^{t_5} e_{P_o}^{t_6}$
- Pour le Produit B : $\sigma^2 = e_B^{t_1} e_{P_B}^{t_2} e_{C_B}^{t_3} e_{S_B}^{t_4} e_H^{t_5} e_{P_o}^{t_6}$

- Pour le Produit C : $\sigma^3 = e_C^{t_1} e_{P_C}^{t_2} e_{C_C}^{t_3} e_{S_C}^{t_4} e_H^{t_5} e_{P_o}^{t_6}$
- Pour le Produit D : $\sigma^4 = e_D^{t_1} e_{P_D}^{t_2} e_{C_D}^{t_3} e_{S_D}^{t_4} e_H^{t_5} e_{P_o}^{t_6}$
- Pour le Produit E : $\sigma^5 = e_E^{t_1} e_{P_E}^{t_2} e_{C_E}^{t_3} e_{S_E}^{t_4} e_H^{t_5} e_{P_o}^{t_6}$

Événements générés par le chariot suspendu

Type d'événement /Ensemble	**Événement**	**Désignation**
Événements Observable /$\sum_o$	e_{P_o}	Le chariot est en position de chargement
	e_A, e_B, e_C, e_D, e_E	Chargement du produit sur le chariot 'A' / 'B' / 'C'/ 'D'/ 'E'
	$e_{P_A}, e_{P_B}, e_{P_C}, e_{P_D}, e_{P_E}$	Le chariot atteint la position 'A' / 'B' / 'C'/ 'D'/ 'E'
	$e_{S_A}, e_{S_B}, e_{S_c}, e_{S_D}, e_{S_E}$	Le produit atteint le stock 'A' / 'B' / 'C'/ 'D'/ 'E'
	$e_{C_A}, e_{C_B}, e_{C_C}, e_{C_D}, e_{C_E}$	Le bras du chariot atteint la position basse dans la Position 'A' / 'B' / 'C'/ 'D'/ 'E'.
	e_h	Le bras du chariot atteint la position haute
Événements de faute/$\sum_f$	F_0	Défaut du capteur
	F_1	Le chariot s'est arrêté à la mauvaise place
	F_2	Dépôt du produit dans le mauvais stock
	F_3	Mauvaise action
Événements régulière/ $\sum_r$	r_1	Fin de l'opération entre les positions 'O' et 'A'
	r_2	Fin de l'opération entre les positions 'A' et 'B'
	r_3	Fin de l'opération entre les positions 'B' et 'C'
	r_4	Fin de l'opération entre les positions 'C'et 'D'
	r_5	Fin de l'opération entre les positions 'D'et 'E'

Ce SED est présenté par une base de données contenant tous ses comportements possibles durant chaque cycle de fonctionnement. Cette base de données est divisée en paires temporelles et tous les paramètres $P_r^j(f_\alpha|Tp)$, $Pg_r^j\left(Suf_r^j(Tp|f_\alpha)\right)$ et $t_r^j(f_\alpha|Tp)$ sont calculés afin de les adapter à notre approche. Après ces étapes de prétraitement des données, nous obtenons un ensemble de données de près de 300 points de données et l'ensemble de données résultant a été divisé par le ratio 4,5 :1 de telle sorte que près de 80% est utilisé comme données d'apprentissage et 20% comme données de test.

5-6-2- Analyses des Modèles RBFNNs

Afin d'effectuer les opérations de diagnostic et de pronostic du SED présentées dans la section précédente, l'implémentation en Matlab 2019b du code RBFNN a été employée à travers une fonction intégrée "$newrb$". L'algorithme de cette fonction commence au début de l'apprentissage à partir d'une couche cachée vide, après les données d'entrée sont injectée dans le RBFNN et pour chaque vecteur d'entrée (La paire temporelle

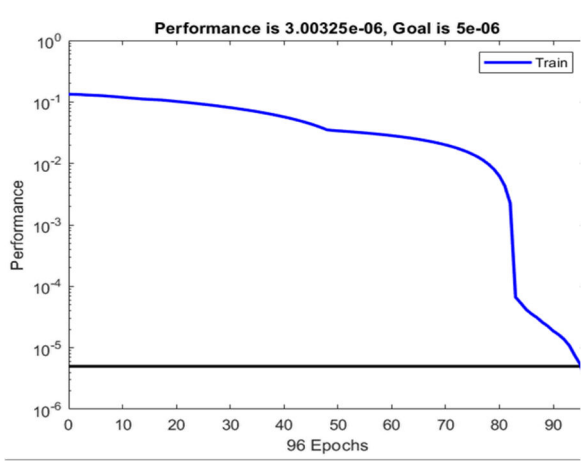
transformée en vecteur en utilisant l'encodage à un coup, $'r'$et $'j'$), le RBFNN essaie de distinguer correctement l'exemple de l'entrée à leurs cibles associées (Les valeurs de diagnosticabilité, de pronosticabilité et le temps restant avant une faute). Bien sûr, au début, les sorties du RBFNN sont très éloignées de ce qu'elles devraient être. Ainsi, la fonction de perte (l'EQM présentée dans la formule (40)) prend les prédictions obtenues par le réseau $(\hat{y})$ et la vraie cible (y) et calcule un score de distance, et mesure la façon dont le réseau s'est comporté sur cet exemple, puis il ajoute un par un des neurones à étalement fixe de la fonction de base radiale à la couche cachée jusqu'à ce que l'erreur quadratique moyenne (EQM) prédéterminée ou le nombre maximum de neurones soient atteints [126]. Dans ce contexte, trois RBFNNs à fonction de base radiale ont été entraînés sur un ensemble de données de près de 240 paires temporelles pour différents cycles et phase de fonctionnement. Les paramètres des réseaux fournis par le processus d'optimisation sont résumés dans le tableau 15 : ainsi, IL, HL et OL représentent respectivement le nombre de neurones dans les couches d'entrée (Input Layer), cachée (Hidden Layer) et de sortie (Output Layer), AF : la fonction d'activation ou de transfert des neurones des couches de sortie et cachée, MNN : Nombre maximal de neurones dans la couche cachée, SP : largeur de l'étalement des neurones (différentes valeurs ont été testées afin de trouver celle qui fournit les meilleures performances), Goal : valeur cible d'EQM (différentes valeurs ont été testées afin de découvrir celle qui atténue en même temps le sur-apprentissage et le sous-apprentissage des modèles). Pour une évaluation détaillée des performances et de la qualité des RBFNNs développés, les graphiques d'EQM du modèle ont été utilisés. A notre connaissance, la valeur de l'EQM à chaque époque est calculée selon la formule suivante :

$$EQM = \frac{1}{X} \times \sum_{i'=1}^{X} (Y_{i'} - \hat{Y}_{i'})^2 \qquad 40$$

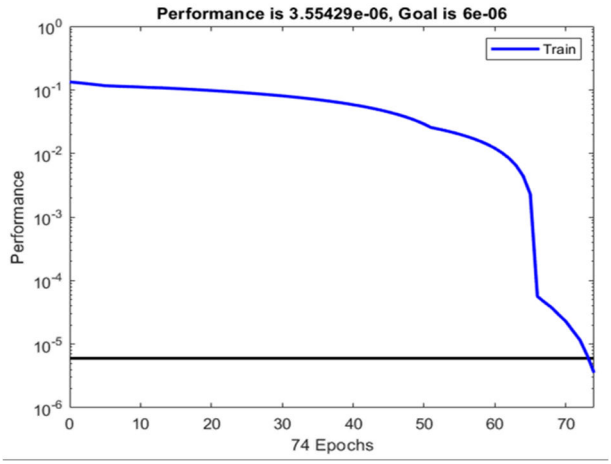
Tel que, " X " est le nombre d'échantillons de l'ensemble de données d'apprentissage, $Y_{i'}$: la valeur attendue de la sortie de l'échantillon i' et $\hat{Y}_{i'}$: la sortie réelle du RBFNN selon l'échantillon i'.

Tableau 14 : Les Paramètres de RBFNNs

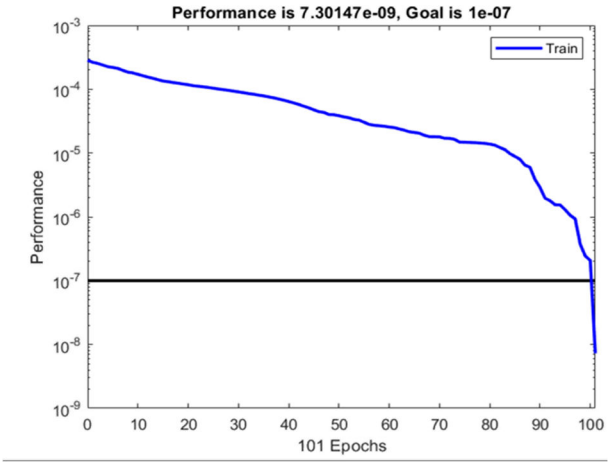
	Diagnostic	**Pronostic**	**Temps restant avant une Faute**
IL	48	48	48
HL	96	74	101
OL	4	4	8
AF	Gaussian RBF/Linear	Gaussian RBF/Linear	Gaussian RBF/Linear
MNN	100	100	120
SP	0.01	0.01	0.1
Goal	$5x10^{-6}$	$6x10^{-6}$	$1x10^{-7}$



a) Diagnostic



b) Pronostic

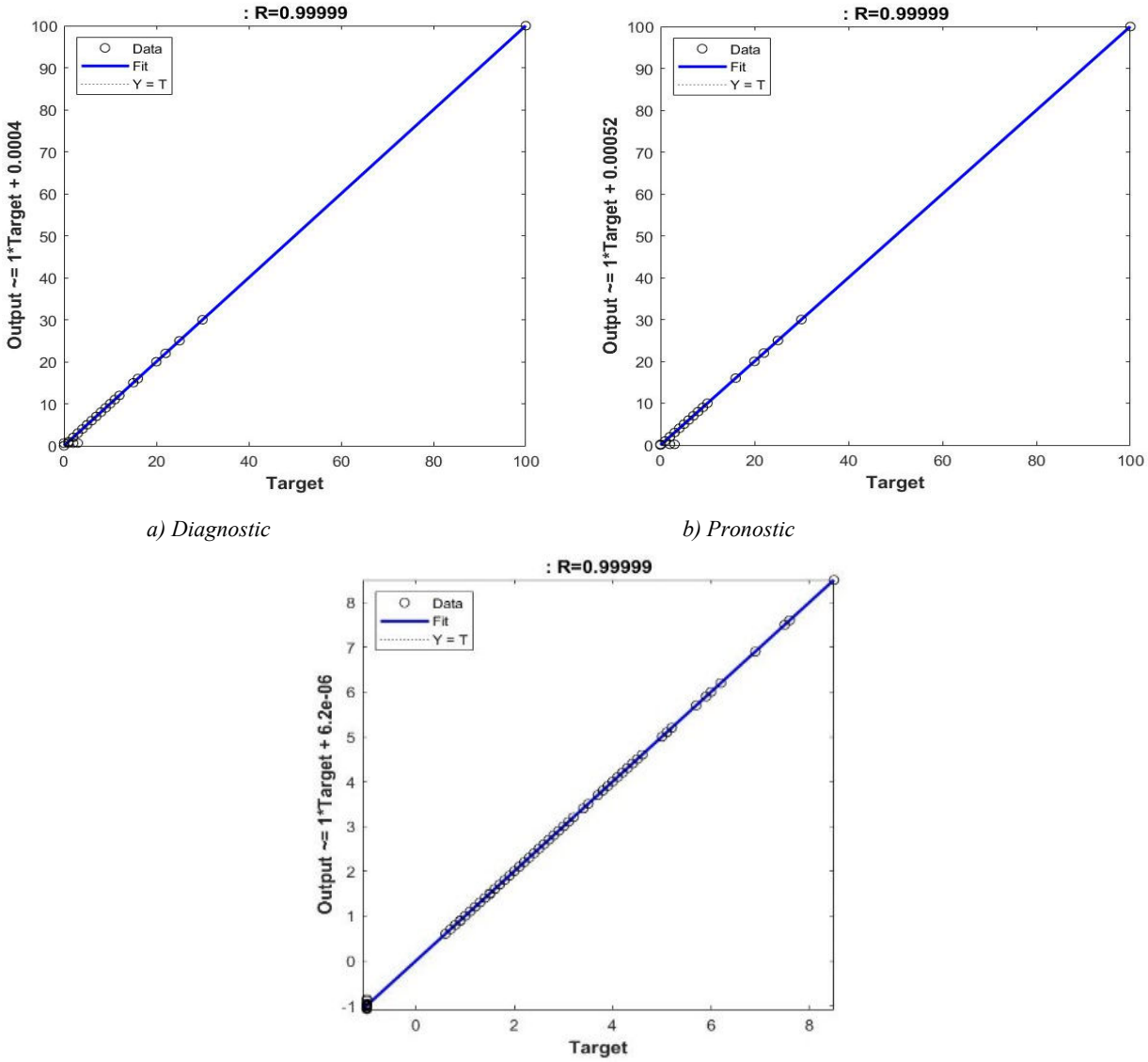


c) Temps Restant avant une Faute

Figure 25 : Performances d'apprentissage des RBFNNs du Diagnostic, du Pronostic et du Temps Restant avant une Faute

Les figures 25-a, 25-b et 25-c représentent les graphiques de performances des RBFNNs utilisés respectivement pour le diagnostic, le pronostic, et le temps restant avant une faute, ces figures montrent comment l'EQM est minimisé pendant l'optimisation des RBFNNs sans aucune indication et aucun signe de sous-apprentissage ou de sur-apprentissage des modèles. Nous pouvons clairement observer que les meilleures valeurs d'EQM se produisent dans les 96 dernières époques du cycle d'apprentissage, avec une EQM associée d'environ $3x10^{-6}$ pour le RBFNN consacré au diagnostic, 74 époques, avec une EQM de $3.55x10^{-6}$, pour le RBFNN dédié au pronostic et 101 époques, avec EQM de $7{,}3\ x10^{-7}$ pour le RBFNN associé au temps restant avant une faute, il est évident que les valeurs des EQMs sont très proches des valeurs des EQMs cibles, ce qui indique que les cibles d'apprentissage sont parfaitement estimées. Après l'opération d'entraînement, les RBFNNs sont testés sur des données composées de près de 70 paires temporelles. Afin de mesurer les performances des RBFNNs sur les données testées, nous allons utiliser les graphiques de régression linéaire (valeur "R"), qui mesurent la corrélation entre les cibles et les sorties des RBFNNs, de sorte que plus la valeur

"*R*" est proche de 1, plus les capacités de prédiction du modèle créé sont excellentes. Les résultats du processus de test sont représentés respectivement dans la Figure 26.

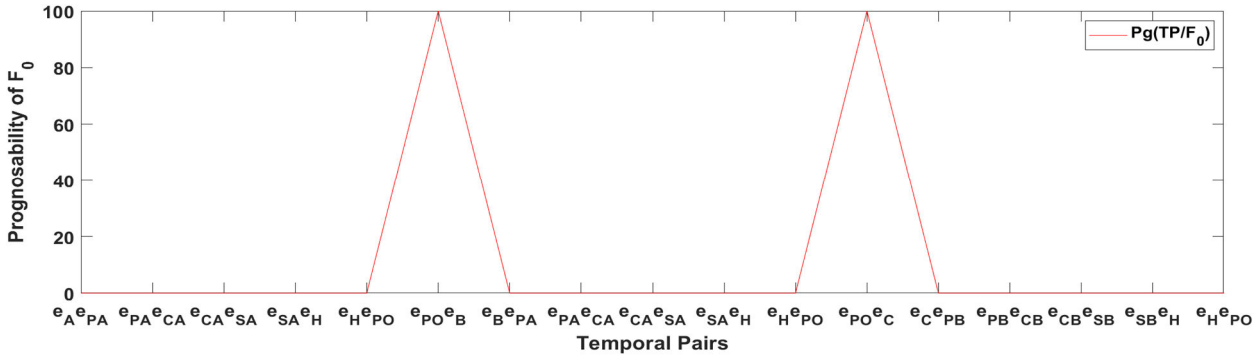


a) Diagnostic *b) Pronostic*

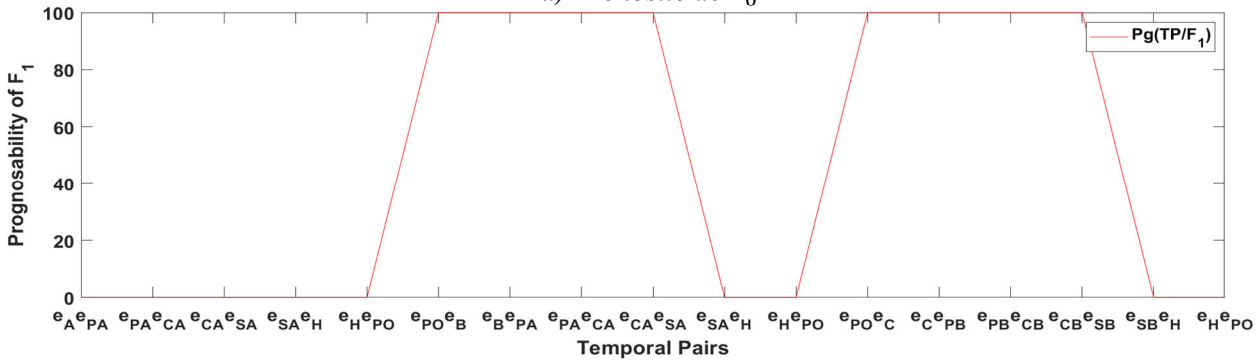
c) Temps Restant avant une Faute

Figure 26: Performance de Test des RBFs du Diagnostic, Pronostic et de du Temps Restant avant une Faute

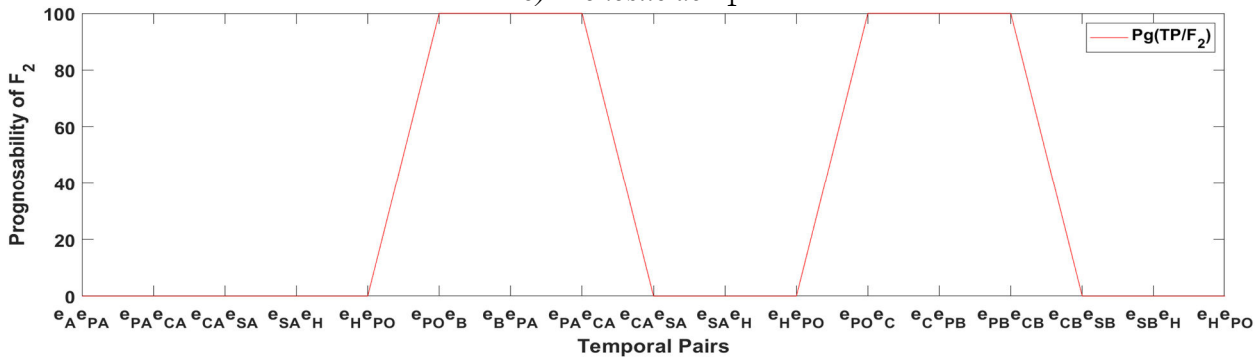
D'après cette figure, nous pouvons clairement observer que les valeurs de régression sont très proches de 1 (plus de 0,999), c'est-à-dire que pratiquement tous les points de données (cercles) se situent sur la ligne de 45°, ce qui indique que les RBFNNs disposent d'une grande capacité de prédiction et peuvent parfaitement traiter les opérations souhaitées. Après la construction des RBFNNs appropriés, nous lançons le chariot afin de transférer le produit 'A', 'B', et 'C'et nous créons exprès des événements de fautes afin de visualiser en pratique la réponse du pronostiquer, le calculateur du temps restant avant une faute et le diagnostiqueur. Les résultats obtenus sont présentés respectivement dans les Figures 27-29.



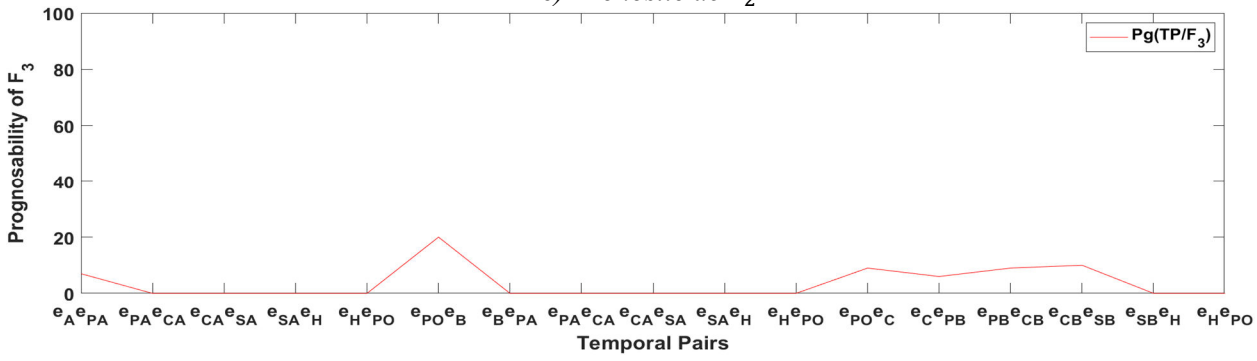
a) Pronostic de F_0



b) Pronostic de F_1



c) Pronostic de F_2



d) Pronostic de F_3

Figure 27: Résultats du Pronostic

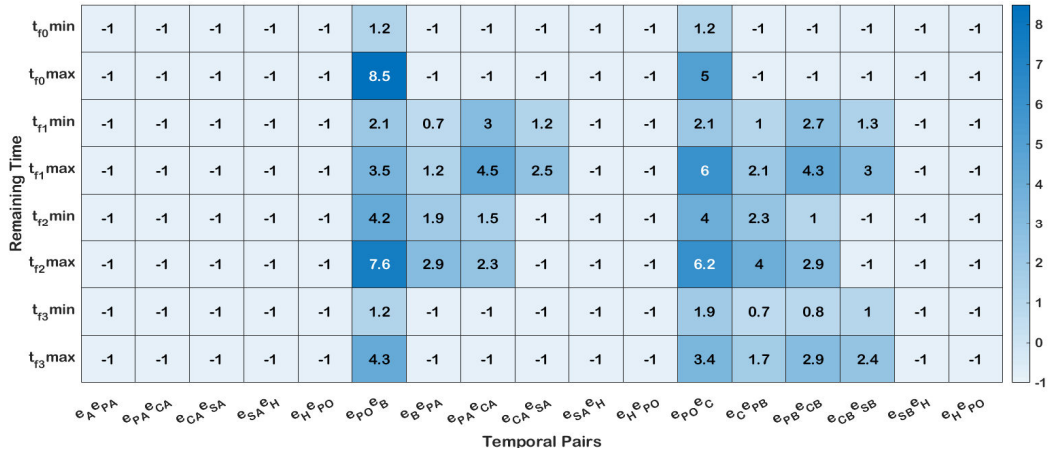
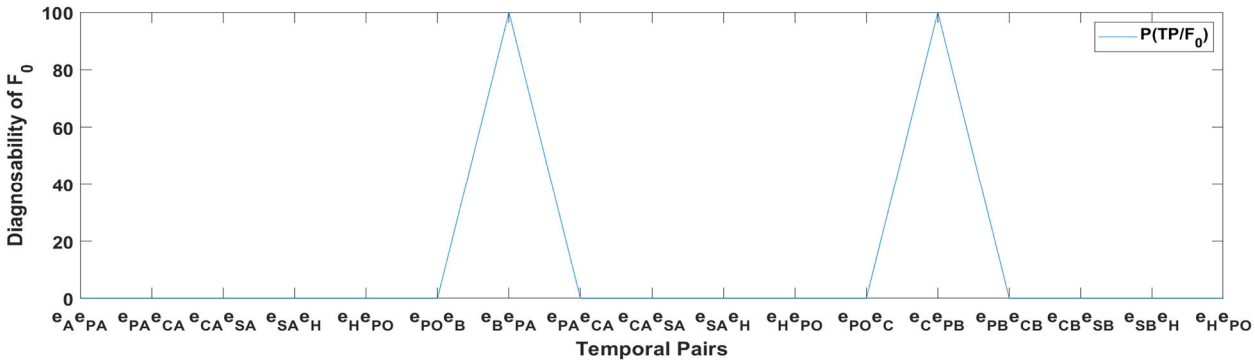
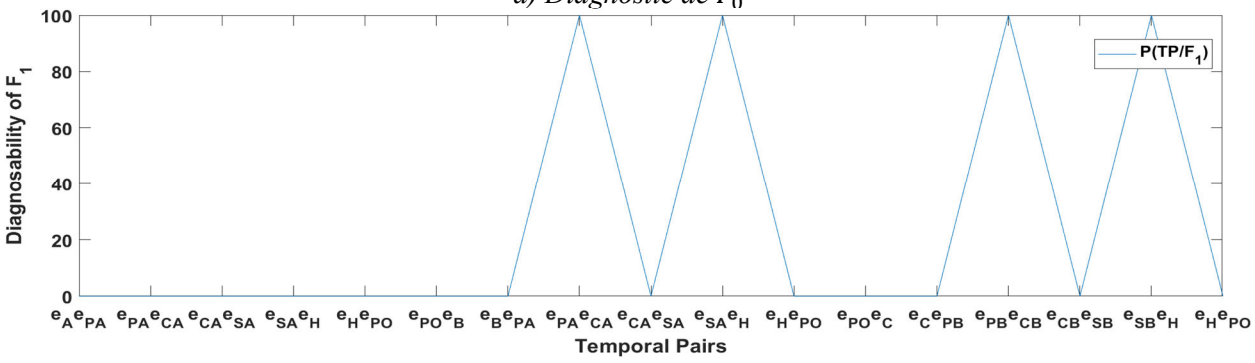


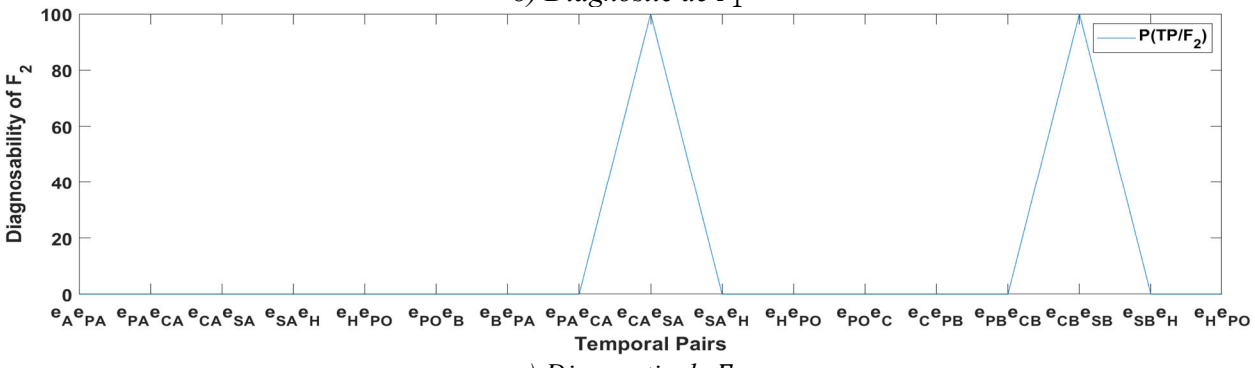
Figure 28: Résultat de Temps restant avant une faute



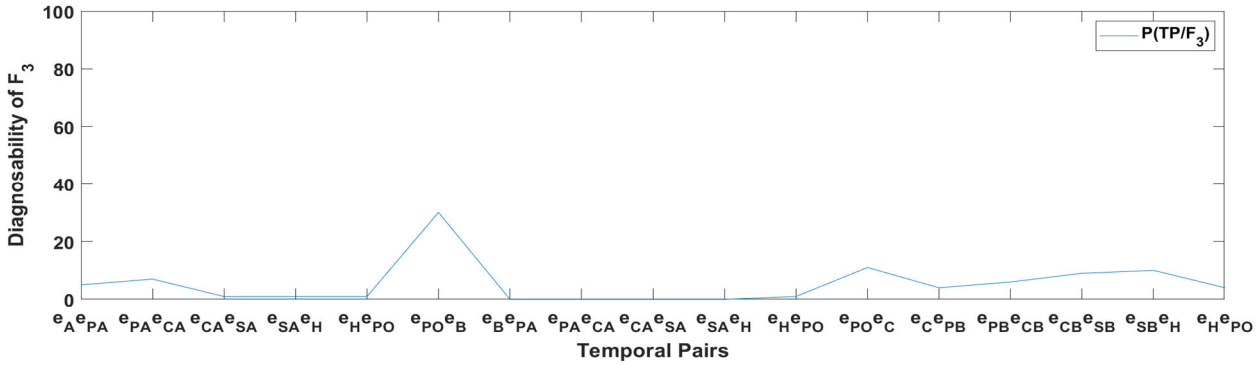
a) Diagnostic de F_0



b) Diagnostic de F_1



c) Diagnostic de F_2



d) Diagnostic de F_3

Figure 29: Résultats du Diagnostic

D'après ces résultats, il est clair que les diagnostiqueurs ont détecté avec précision l'apparition des événements de faute F_0 (Dans les paires temporelles $e_B\, e_{PA}$ et $e_C\, e_{PB}$), F_1 dans les pairs temporelles $e_{PA}e_{CA}, e_{SA}e_H, e_{PB}e_{CB}$ et $e_{SB}e_H$) et F_2 dans les pairs temporelles $e_{CA}e_{SA}, e_{CB}e_{SB}$ en générant une probabilité d'occurrence de 100%, et qui sont déjà prédites par le pronostiqueur dans les paires temporelles précédentes, c'est-à-dire avant même leur occurrence. De même, pour l'événement de faute F_3, mais dans ce cas, les probabilités sont très faibles, ce qui indique que l'apparition de cet événement est improbable. De plus, grâce au RBFNN dédié au calcul du temps restant, il nous a donné une estimation de l'intervalle de temps où les événements de faute peuvent apparaître.

6-Conclusion

Dans ce chapitre, nous avons présenté le cadre théorique pour construire un diagnostiqueur et pronostiqueur basé sur les réseaux de neurones du type 'Feed-Forward', qui permet l'analyse des grandes données extraites du modèle statistique du système à événements discrets pour traiter le diagnostic et le pronostic en ligne. À cette fin, nous avons proposé trois RdNs à action directe, qui garantissent le calcul de trois indices principaux: Diagnosticabilité, Pronosticabilité, et le Temps restant avant une faute de manière à déterminer l'état actuel et futur probable du système à événements discrets ainsi que l'intervalle de temps, où un événement de faute peut probablement se produire dans les étapes de fonctionnement à venir.

Dans le prochain chapitre nous allons couvrir la deuxième méthode, qui est le diagnostic et le pronostic des SEDs à l'aide des réseaux de neurones récurrents.

Chapitre 4 : Diagnostic et Pronostic des SEDs par Réseaux de Neurones Récurrent

Table des Matières

1- Introduction 86

2- Diagnostic et Pronostic des Fautes par les Réseaux de Neurones Récurrents 86

2-1- Diagnosticabilité 88

2-2- Pronosticabilité 90

2-3- Temps restant avant une Faute 92

2-4- Les Dimensions Effectives du Graph Partiel 92

3- Construction des Réseaux de Neurones Récurrent et Etude de cas 93

3-1- Ensemble de Données du Diagnostic 94

3-2- Ensemble de Données de Pronostic 94

3-3- Ensemble de Données du Temps Restant avant une Faute 94

3-4- Algorithme du Diagnostic et du Pronostic des SED par paire temporelle 94

3-4- Etude de Cas 95

3-4-1- La Construction et l'Evaluation des Réseaux de Neurones Récurrents 96

4- Conclusion 100

1- Introduction

L'objectif de ce chapitre c'est l'exploitation d'un type particulier des RdNs artificiels, qui sont les réseaux de neurones récurrents, ces types des RdNs sont dédiés directement au traitement des données séquentiel c'est-à-dire des données qui varient dans le temps ce qui les rend les plus adaptés aux problèmes de surveillance dynamique. Dans la littérature, les RNNs sont généralement réservés aux systèmes continus, et l'utilisation de tels outils au profit des SEDs est très limitée. Dans ce chapitre, au lieu de représenter les cycles de fonctionnement des SEDs par des mots temporisés, ils seront représentés par des graphes de transition d'états ou simplement des graphes, tel qu'un graphe de transition d'états est composé d'une séquence d'états qui peut être normaux ou de fautes et d'une séquence d'événements, qui peuvent être observables ou non. Soit TG^* l'ensemble de tous les graphs qui peuvent être formés dans un SED, de sorte que chaque événement est indexé par sa date d'occurrence, par exemple $\sigma = x_1 \xrightarrow{e_1^{t1}} x_2 \xrightarrow{e_2^{t2}} \dots x_{\mathrm{i}} \xrightarrow{e_i^{ti}} \dots \xrightarrow{e_n^{tn}} x_{\mathrm{n+1}}$ est un graphe de transition d'états, de sorte que e_i est un événement appartenant à $\sum$ et $t_i(t_i > 0)$ est sa date d'occurrence et x_{i} un état appartenant à X de sorte que x_1 est l'état initial du SED. Donc en se basant sur cette représentation, une nouvelle approche du diagnostic et de pronostic des SEDs par l'exploitation des réseaux de neurones récurrent sera présentée.

2- Diagnostic et Pronostic des Fautes par les Réseaux de Neurones Récurrents

En utilisant les RNNs pour assurer les opérations de Diagnostic et de Pronostic, chaque événement généré par le SED sera immédiatement présenté au RNN en plus des "K" derniers événements observables qui se sont déjà produits, pour déterminer l'état actuel probable (Diagnostic) en détectant si le système a dévié vers un état de faute F_α ou non et si cette déviation peut être dans les prochains états de fonctionnement. Le diagnostic et le pronostic du SED en utilisant les RNNs peuvent être assuré comme illustré dans la figure suivante (Figure 30) :

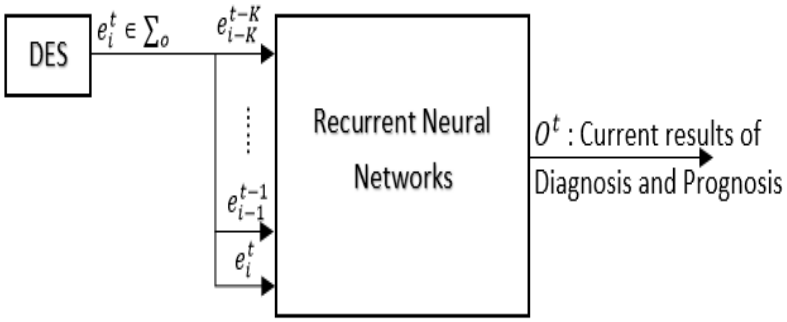


Figure 30: Diagnostic et Pronostic des Fautes par RNNs

Comme mentionné dans le chapitre précédent, la construction d'un modèle statistique pour entraîner les RdNs est indispensable. Pour cela, pour chaque graphe d'états de transition possible, une base de données est élaborée. La base de données construite pour assurer l'apprentissage des RNNs doit disposer en entrée, en plus de l'événement actuel, les "K" derniers événements observables, et en sortie, le résultat du diagnostic et du

pronostic. Par conséquent, nous proposons de diviser chaque graphe de transition σ^j à des graphs partiels, de sorte que chaque graph partiel SP_i^j contient l'événement courant e_i et les "K" derniers événements observables.

Définition 6 : (Graphs partiels d'un graphe de transition) soit $\sigma^j \in TG^*$ un graphe de transition tel que : $\sigma^j = x_1 \xrightarrow{e_1^{t1}} x_2 \xrightarrow{e_2^{t2}} \dots x_i \xrightarrow{e_i^{ti}} \dots \xrightarrow{e_n^{tn}} x_{n+1}$ enregistré à partir du $j^{\text{ème}}$ cycle d'un SED et "q" est le nombre d'événements observables. Un Graphs partiels SP_i^j d'un graphe de transition σ^j est une suite d'événements, qui est formée par la concaténation de l'événement e_i et les "K" dernier événements observables préfixe de e_i ça veut dire que $SP_i^j = e_{i-K}^{t-K} \dots e_{i-1}^{t-1} e_i^{ti}$ tel que $e_0^0 = \varepsilon$ est l'événement vide. Étant donné SP un graph partiel, les notations suivantes sont considérées :

- $P_o(SP)$: $SP \longrightarrow \sum_o$: La projection qui élimine les états, le temps et les événements non observables d'un graph partiel SP s'ils existent.
- $\overline{SP}$: Le préfix de SP qui contient tous les événements générés au cours du cycle de fonctionnement, du premier événement jusqu'à l'événement qui s'est produit avant SP.
- $\delta : X \times SP \longrightarrow X$: La fonction de transition partielle, cette fonction détermine l'état x' du SED après la séquence d'événements qui forment SP à partir d'un état x c'est-à-dire, $\delta(x, SP) = x'$.
- $T(x)$ et $T(e_i)$: Les projections, qui donnent respectivement le temps de début d'un état x et la date d'occurrence d'un événement e_i.
- E^j: L'ensemble de tous les graphs partiels, qui sont dérivés d'un graph σ^j.
- $E_{F_\alpha}^j$: : L'ensemble de tous les graphs partiels, qui sont dérivés d'un graph σ^j qui conduisent à un état de faute F_α, plus formellement :

$$E_{F_\alpha}^j = \{ SP_i^j \in E^j / \delta(x, P_o(SP_i^j)) = F_\alpha \}$$

Tel que $x = \delta(x_1, P_o(\overline{SP_i^j}))$.

- $E(SP)$: L'ensemble qui contient de tous les éléments de TG^* , tous les graphs partiels qui ont la même projection P_o comme SP, plus formellement :

$$E(SP) = \{\sigma^j \in TG^*, SP_i^j \in E^j / \ P_o(SP_i^j) = SP\}$$

Exemple 8 : Par exemple, considérons un SED défini par leurs ensembles $\sum_o = \{a, b, c, d\}$, $\sum_f = \{f_1, f_2, f_3\}$, $\sum_r = \{r_1, r_2\}$, $X_N = \{x_1, x_2, x_3, x_4\}$ et $X_F = \{F_1, F_2, F_3\}$. Et σ_1 est le graph généré pendant le premier cycle du

fonctionnement du SED, de sorte que : $\sigma^1 = x_1 \overset{b^1}{\rightarrow} x_2 \overset{a^{1.5}}{\longrightarrow} x_1 \overset{f_1^{2.3}}{\longrightarrow} F_1 \overset{b^5}{\rightarrow} x_2 \overset{f_2^{5.6}}{\longrightarrow} F_2 \overset{a^7}{\rightarrow} x_1 \overset{r_1^{7.3}}{\longrightarrow} x_1 \overset{d^5}{\rightarrow} x_3 \overset{c^{11}}{\longrightarrow} x_2 \overset{a^{12.4}}{\longrightarrow} x_1 \overset{f_3^{12.9}}{\longrightarrow} F_3 \overset{c^{13.1}}{\longrightarrow} F_3$. Pour $K = 3$, on obtient l'ensemble suivant :

$$E^1 = \{SP_1^1 = \varepsilon\,\varepsilon\,\varepsilon b^1, SP_2^1 = \varepsilon\,\varepsilon b^1 a^{1.5}, SP_3^1 = \varepsilon b^1 a^{1.5}\, f_1^{2.3} b^5, SP_4^1 = b^1 a^{1.5}\, f_1^{2.3} b^5 f_2^{5.6} a^7, SP_5^1 = a^{1.5}\, f_1^{2.3} b^5 f_2^{5.6} a^7 r_1^{7.3} d^5, SP_6^1 = b^5 f_2^{5.6} a^7 r_1^{7.3} d^5 c^{11}, SP_7^1 = a^7 r_1^{7.3} d^5 c^{11}\, a^{12,4}, SP_8^1 = d^5 c^{11}\, a^{12.4} f_3^{12.9} c^{13.1}\}$$

En outre, les ensembles suivants sont obtenus : $E_{F_1}^1 = \{SP_2^1\}, E_{F_2}^1 = \{SP_3^1\}$ et $E_{F_3}^1 = \{SP_8^1, SP_9^1\}$. Et concernant $E(SP)$, nous prenons $SP = adcb$ comme exemple, donc $E(adca) = \{SP_7^1\}$.

2-1- Diagnosticabilité

Dans le contexte des graphs partiel, la diagnosticabilité est définie comme la capacité de détecter une déviation du SED de son comportement normal à un comportement de faute F_α, due à la présence de l'événement de faute f_α, après l'observation du dernier événement observable, tout en tenant compte des "K" derniers événements survenus par le SED, c'est-à-dire que le diagnostiqueur va jouer le rôle d'un estimateur d'état qui permet, à partir des mesures, de déterminer l'état actuel du système. Un graph partiel SP est diagnosticable si et seulement si on peut détecter sans ambiguïté cette déviation. A cette fin, sur la base du modèle statistique et des données historiques collectées pour un SED, un paramètre ou indice statistique appelé indice de diagnosticabilité (similaire à celui défini dans le chapitre 3) est défini, qui sera noté par $P(F_\alpha|Sp)$, qui permet de calculer la probabilité que le SED dévie vers un état de faute F_α, en observant le graph partiel Sp, tel que :

$$P(F_\alpha|Sp) = \frac{Nombre\ des\ SP_i^j\ similaire\ \text{à}\ SP\ qui\ mène\ à\ un\ etat\ de\ faute\ F_\alpha}{Nombre\ des\ SP_i^j\ similaire\ \text{à}\ SP} \qquad 41$$

Plus formellement :

$$P(F_\alpha|SP) = \sum_j \frac{card\left(E(SP) \cap E_{F_\alpha}^j\right)}{card\left(E(SP)\right)} \qquad 42$$

Tel que : $E(SP) \cap E_{F_\alpha}^j$ fournis les graphs partiels similaires à Sp, ce qui conduit à l'état de faute F_α, Et le graph partiel Sp est diagnosticable par rapport à un état de faute F_α, si est seulement si $P(F_\alpha|Sp) = 1$ où $P(F_\alpha|Sp) = 0$.

Preuve 3 :

- Dans le cas ou $P(F_\alpha|Sp) = 1$:

  $P(F_\alpha|SP) = 1 \Leftrightarrow \sum_j card\left(E(SP) \cap E_{F_\alpha}^j\right) = card\left(E(SP)\right) \Leftrightarrow \forall \sigma^j \in TG^*, \forall Sp_i^j \in E^j$:
  $P_o\left(SP_i^j\right) = SP \Rightarrow \delta\left(x,\ P_o(SP_i^j)\right) = F_\alpha$, tel que : $x = \delta(x_1, P_o(\overline{SP_i^j}))$

  Dans ce cas, Sp est diagnosticable par rapport à F_α et la déviation du SED vers le mode de faute F_α est certaine.

- Dans le cas ou $P(F_\alpha|Tp) = 0$

$P(F_\alpha|Sp) = 0 \Leftrightarrow \sum_j card\left(E(Sp) \cap E^j_{F_\alpha}\right) = 0 \Leftrightarrow \forall \sigma^j \in TG^*, \forall SP_i^j \in E^j : P_o\left(SP_i^j\right) = SP \Rightarrow$

$\delta\left(x, P_o(SP_i^j)\right) \neq F_\alpha$ Tel que $x = \delta(x_1, P_o(\overline{\left(SP_i^j\right)}))$.

Dans ce cas, SP est diagnosticable par rapport à F_α et la déviation du SED vers le mode de faute F_α n'est pas attendue. Cependant, elle peut conduire à un autre mode de faute. Par conséquent, SP est diagnosticable par rapport à l'état normal si et seulement si $\sum_\alpha P(F_\alpha|SP) = 0$

Preuve 4 :

$\sum_\alpha P(F_\alpha|SP) = 0 \Leftrightarrow \forall \sigma^j \in TG^*, \forall SP_i^j \in E^j, \forall F_\alpha \in \mathrm{X}_F : P_o\left(SP_i^j\right) = SP \Rightarrow \delta\left(x, P_o(SP_i^j)\right) \neq F_\alpha$; Tel que $x = \delta(x_1, P_o(\overline{\left(SP_i^j\right)}))$.

C'est-à-dire que tous les graphs partiels SP, ne mènent à aucun état de faute F_α, donc le graph partiel est diagnosticable et le SED est dans son comportement normal. Donc, un SED est diagnosticable si et seulement si : $\forall \sigma^j \in TG^*, \forall SP_i^j \in E^j, SP_i^j$ est diagnosticable. Sinon, le graph partiel n'est pas diagnosticable. De plus, la probabilité que le SED dévie vers un mode de faute est prévue.

Cette définition est similaire au diagnostiqueur basé sur les automates finis proposé par Sampath et al. dans [18]. Cependant, cette définition renvoie des résultats binaires (N : Etat Normal, F : Etat de Faute) c'est-à-dire la déviation ou non du SED en plus d'un état incertain. Dans ce cas, aucune indication n'est fournie ou disponible sur l'état actuel probable. Mais grâce à la définition de la diagnosticabilité proposée dans cette approche, l'indice $P(F_\alpha|SP)$ permet de fournir un résultat nuancé, c'est-à-dire, dans tous les cas il fournit une probabilité qui donne un aperçu global de l'état courant probable du fonctionnement du système.

Exemple 9 : Nous considérons σ^1 (le même comme dans l'exemple précédent), σ^2 et σ^3 sont respectivement les graphs générés pendant le premier, le deuxième et le troisième cycle de fonctionnement d'un SED de sorte que : $\sigma^2 = x_1 \xrightarrow{b^1} x_2 \xrightarrow{a^{1.5}} x_1 \xrightarrow{f_2^{5.6}} F_2 \xrightarrow{a^7} x_1 \xrightarrow{r_1^{7.3}} x_1 \xrightarrow{d^9} x_3 \xrightarrow{c^{11}} x_2 \xrightarrow{a^{12.4}} x_1$ et $\sigma^3 = x_1 \xrightarrow{b^1} x_2 \xrightarrow{a^{1.5}} x_1 \xrightarrow{f_2^{2.6}} F_2 \xrightarrow{d^5} x_3 \xrightarrow{c^{11}} x_2 \xrightarrow{f_2^{12.4}} F_2 \xrightarrow{a^{12.4}} x_1$. Après avoir formé les ensembles requis pour chaque graph ($K = 3$) et soit $SP = \varepsilon\, \varepsilon$ ba, on obtient $card\left(E(SP)\right) = 3$, $\sum_{j=1}^3 card\left(E(SP) \cap E^j_{F_1}\right) = 1$ donc $P(F_1/SP) = 33\%$, $\sum_{j=1}^3 card\left(E(SP) \cap E^j_{F_2}\right) = 2$ donc $P(F_2/SP) = 66\,\%$ et $\sum_{j=1}^3 card\left(E(SP) \cap E^j_{F_3}\right) = 0$ donc nous obtenons $P(F_3/SP) = 0\%$.

Par conséquent, SP est diagnosticable par rapport à l'état de faute F_3 et non diagnosticable par rapport aux états de faute F_1 et F_2, et la probabilité que le SED dévie vers l'état F_1 est de 33% et de 66% pour F_2.

2-2- Pronosticabilité

Dans ce contexte, la pronosticabilité peut être définie comme la probabilité que le SED dévie de son comportement normal vers un mode de faute F_α lors de son prochain état de fonctionnement. A cet égard, nous allons définir un autre indice statistique appelé pronosticabilité, qui sera noté par $Pg(F_\alpha|SP)$ qui fournit la probabilité que le SED dévie vers un mode de faute F_α, après l'apparition de chacun des événements "K" immédiatement postérieurs à SP, et il est nommé par la continuation de SP et on le note par SP^{-1}. A cette fin, toutes les continuations de chaque graph partiel doivent être analysées. Pour cette raison, nous définissons l'ensemble $Suf^j(F_\alpha)$ qui regroupe tous les graphs partiels dérivés de σ^j, qui ont une continuation, qui conduit à l'état de faute F_α, tel que :

$$Suf^j(F_\alpha) = \{SP_i^j \in E^j / \delta\left(x,\ P_o(SP_i^j SP_{i,k}^{j\ -1})\right) = F_\alpha\}$$

Tel que : $x = \delta(x_1, P_o(\overline{SP_i^j}))$ et $SP_{i,k}^{j\ -1} = \begin{cases} e_{i+1} \\ e_{i+1}e_{i+2} \\ \vdots \\ e_{i+1} \dots e_{i+K} \end{cases}$

Et $SP_i^j SP_{i,k}^{j\ -1}$ est la concaténation du graph partiel SP_i^j et sa continuation $SP_{i,k}^{j\ -1}$, tel que $E^{-1}(SP)$ l'ensemble de toutes les continuations possibles de SP. Ensuite, à partir du modèle statistique associé au SED étudié, l'indice de pronostic $Pg(F_\alpha|SP)$ est calculé comme suit :

$$Pg(F_\alpha|SP) = \frac{le\ nombre\ des\ chemins\ partiels\ similaire\ \text{à}\ SP\ qui\ ont\ une\ continuation\ qui\ mène\ \text{à}\ F_\alpha}{le\ nombre\ des\ chemins\ partiels\ similaire\ \text{à}\ SP} \quad 43$$

Plus Formellement :

$$Pg(F_\alpha|SP) = \frac{card\ (\cup_j(Suf^j(F_\alpha) \cap E^{-1}(SP)))}{card\big(E(SP)\big)} \quad 44$$

Ou $\cup_j(Suf^j(F_\alpha) \cap E^{-1}(SP))$ et l'ensemble de tous les graphs partiels similaires à SP, qui ont une continuation SP^{-1} qui mène à l'état de faute F_α. Et le graph partiel SP est pronosticable par rapport à un état de faute F_α si et seulement si $Pg(F_\alpha|SP) = 0$ ou $Pg(F_\alpha|SP) = 1$.

Preuve 5 :

- Dans le cas où : $Pg(F_\alpha|SP) = 1$:

$$Pg(F_\alpha|SP) = 1 \Leftrightarrow card(\cup_j(Suf^j(F_\alpha) \cap E^{-1}(SP))) = card(E^{-1}(SP))$$

$$\Leftrightarrow \forall \sigma^j \in TG^*, \forall SP_i^j \in E^j, \exists SP^{-1} \in E^{-1}(SP) : P_o\big(SP_i^j\big) = SP$$

$$\Rightarrow \delta\left(x,\ P_o(SP_i^j SP_{i,k}^{j\ -1})\right) = F_\alpha.$$

Tel que : $x = \delta(x_1, P_o(\overline{SP_i^j}))$.

Dans ce cas, SP est pronosticable par rapport au mode F_α, de sorte qu'à partir de la séquence d'événements SP observés, il est possible de prédire la déviation vers le mode F_α, et le système va inévitablement vers l'état de faute F_α.

- Dans le cas où : $Pg(F_\alpha|SP) = 0$:

$$Pg(F_\alpha|SP) = 0 \Leftrightarrow card(\cup_j(Suf^j(F_\alpha) \cap E^{-1}(SP))) = 0$$
$$\Leftrightarrow \cup_j(Suf^j(F_\alpha) \cap E^{-1}(SP)) = \emptyset$$
$$\Leftrightarrow \forall \sigma^j \in TG^*,\ \forall SP_i^j \in E^j, \nexists SP^{-1} \in E^{-1}(SP) : P_o\left(SP_i^j\right) = SP$$
$$\Rightarrow \delta\left(x,\ P_o(SP_i^j {SP_{i,k}^j}^{-1})\right) = F_\alpha$$

Dans ce cas, il n'y a pas de continuation de SP, qui mène à l'état de faute F_α, et SP est pronosticable par rapport au mode F_α, cependant, il peut disposer d'une certaine continuation qui mène à différents états de faute. Pour cela, SP_i^j est pronosticable par rapport à l'état normal si et seulement si : $\sum_\alpha Pg(F_\alpha|SP) = 0$.

$$\sum_\alpha Pg(F_\alpha|SP) = 0 \Leftrightarrow \forall \sigma^j \in TG^*, \forall SP_i^j \in E^j, \forall F_\alpha \in \mathrm{X}_F, \nexists SP^{-1} \in E^{-1}(SP): P_o\left(SP_i^j\right)$$
$$= SP: \delta\left(x,\ P_o(SP_i^j {SP_{i,k}^j}^{-1})\right) = F_\alpha$$

C'est-à-dire que SP ne dispose d'aucune continuation qui mène vers un mode de faute et le SED sera dans son comportement normal. De plus, un SED est pronosticable si et seulement si :$\forall \sigma^j \in TG^*, \forall SP_i^j \in E^j: SP_i^j$ est pronosticable.

La définition proposée est similaire à celle proposée par Genc et al. [52], qui s'inspire de l'approche du diagnostiqueur développée dans [18], mais dans ce cas, une certaine décision sur le futur est fournie uniquement lorsque la continuation mène à l'état de faute ou non (décision binaire). Cependant, dans notre cas, une probabilité de chaque état futur probable est fournie grâce à l'indice de pronostic $Pg(F_\alpha|SP)$.

Remarque 6 : Dans le cas où il y a plus d'une continuation pour un graph partiel, qui mène au même état de faute, nous prenons celle qui détecte la déviation la plus proche possible.

Exemple 10 : Nous considérons une autre fois σ^1 , σ^2 et σ^3 les trois graphs de l'exemple 9 et soit $K = 3$ et $SP = \varepsilon\ \varepsilon\ ba$: $card(E(SP))$=3, $card\ (\cup_j(Suf^j(F_1) \cap E^{-1}(SP)))$=0 donc $Pg(F_1|SP) = 0\%$, $card\ (\cup_j(Suf^j(F_2) \cap E^{-1}(SP)))$=2 donc $Pg(F_2|SP) = 66\%$ et $card\ (\cup_j(Suf^j(F_3) \cap E^{-1}(SP)))$=0 alors $Pg(F_3|SP) = 0\%$. En conséquence, SP est pronosticable par rapport aux états de faute F_1 et F_3 et non pronosticable par rapport à F_2, de sorte que le SED peut dévier dans le futur avec une probabilité de 66% vers l'état F_2

2-3- Temps restant avant une Faute

Pour calculer le temps restant avant une défaillance concernant les graphs partiaux, nous définissons un nouvel indice $t(F_\alpha|SP)$ qui fournira un intervalle de temps, où F_α peut apparaître dans les étapes de fonctionnement à venir lorsque SP est observé, tel que :

$$t(F_\alpha|SP) = \begin{cases} +\infty; if\ Pg(F_\alpha|SP) = 0 \\ [Min(T(F_\alpha/SP)) - T(e_i);\ \text{Sinon} \\ Max(T(F_\alpha/SP)) - T(e_i)] \end{cases} \quad 45$$

Où $T(F_\alpha|SP) = \{T(F_\alpha)|\ P_o(SP_i^j) = SP\ et\ \ \delta\left(x,\ P_o(SP_i^j SP_i^{j^{-1}})\right) = F_\alpha\}$ est l'ensemble de toutes les valeurs probables du temps restant estimé avant l'apparition du mode de faute F_α dans la suite possible de SP et $T(e_i)$ est la date d'occurrence de l'événement observable actuel généré par le SED.

Remarque 7 : Si $Min(T(F_\alpha/SP)) - T(e_i) < 0$, on pose $Min(T(F_\alpha/SP)) - T(e_i) = 0$.

Exemple 11 : On considère une autre fois les mêmes graphs présentés dans l'exemple 9 et 10 et soit $SP = \varepsilon\ \varepsilon\ ba$ et supposons que $T(ba) = 1{,}5$ et prenons F_2 comme exemple (car $Pg(F_1|SP) = 0\%$ et $Pg(F_3|SP) = 0\%$) donc nous obtenons $t(F_2/SP) = [5{,}6 - 1{,}5\ ;\ 12{,}4 - 1{,}5\ ] = [4{,}1; 10{,}9]$ C'est-à-dire que le SED peut probablement dévier vers l'état de faute F_2 presque après 4,1 à 10,9 unités de temps.

2-4- Les Dimensions Effectives du Graph Partiel

Comme il a été déjà présenté, chaque graphe de transition doit être divisé en plusieurs graphs partiels. Cependant, sa dimension est directement liée au méta-paramètre "K" de sorte qu'il définit le nombre d'événements observables à prendre en compte, en plus de l'événement courant, mais la valeur de ce paramètre modifie directement l'efficacité du diagnostic et du pronostic. Nous proposons donc deux autres paramètres $K_{d,\alpha}$ et $K_{p,\alpha}$ qui seront les dimensions du graph partiel, qui prennent en considération l'efficacité de la diagnosticabilité et de la pronosticabilité. Dans ce sens, un algorithme récursif (Algorithme 2) est proposé, prenant comme entrées les graphes de transition et deux autres paramètres K_{max} et K_{min}, qui représentent les dimensions limites, qui respectivement ne doivent pas être dépassés et ne doivent pas être inférieures. Et calcule les indices $P(F_\alpha|SP)$ et $Pg(F_\alpha|SP)$ et fournis comme sorties $K_{d,\alpha}$ et $K_{p,\alpha}$, qui maximisent respectivement une fonction de coût $F_d^{K_{d,\alpha}}$ et $F_p^{K_{p,\alpha}}$. Ces fonctions de coût calculent la moyenne de tous les indices P et Pg de tous les graphs partiels, appartenant à $\bigcup_j E_j$ pour chaque état de faute F_α.

$$F_{d,\alpha} = \underset{SP_i^j \in \bigcup_j E_j}{\text{Mean}} \left\{P\left(F_\alpha \middle| SP_i^j\right)\right\}$$

$$F_{p,\alpha} = \underset{SP_i^j \in \bigcup_j E_j}{\text{Mean}} \left\{Pg\left(F_\alpha \middle| SP_i^j\right)\right\}$$

3- Construction des Réseaux de Neurones Récurrent et Etude de cas

L'objectif de cette section est de présenter les éléments fondamentaux des modèles RNNs pour assurer les opérations du Diagnostic, Pronostic, et le calcul du temps restant avant une faute introduite dans le cadre théorique. Pour chaque état de faute, quatre RNN sont associés pour assurer respectivement le diagnostic, le pronostic, et le temps restant minimum et maximum (car dans ce cas on a choisi de traiter l'opération du diagnostic et de pronostic des SEDs comme un problème de régression, donc chaque paramètre pour chaque état de faute doit être calculé indépendamment). En pratique, presque tous les SEDs génèrent des séquences très longues, ce qui fait que le LSTM pourrait parfaitement gérer le diagnostic et le pronostic des SEDs, ce qu'une autre couche de RNNs ne peut pas faire.

Remarque 8 : Dans ce chapitre les évènements seront représentés aussi en utilisant le codage à un coup comme présenté dans la section 5-2 du chapitre 3.

Algorithme 2

Inputs : $TG^*, K_{min,d}, K_{min,p}, K_{max,d}, K_{max,p}$
Outputs : $K_{d,\alpha}, K_{p,\alpha}$

Step 1 : $K_d = K_{min,d,}$, $K_p = K_{min,p}$
Step 2 : For each $\sigma_j \in TG^*$ form SP_i^j
Step 3 : While ($K_d \leq K_{max,d}$) do :
For $\alpha = 1$ a $\alpha = card(\mathrm{X}_F)$
For each SP_i^j
Compute $P\left(F_\alpha \middle| SP_i^j\right)$
End for
End for
$K_d <= K_d + 1$
Go to **Etape 2**
End While
Step 4 : While ($K_p \leq K_{max,p}$) do :
For $\alpha = 1$ to $\alpha = card(\mathrm{X}_F)$
For SP_i^j
Compute $Pg\left(F_\alpha \middle| SP_i^j\right)$
End for
End for
$K_p <= K_p + 1$
Go to **Etape 2**
End while
Step 5 : Compute F_d
Compute F_p
Step 6 : Compute $K_{d,\alpha}$; $K_d <= \max_{K_d}\{F_{d,\alpha}\}$
Compute $K_{p,\alpha}$; $K_p <= \max_{K_p}\{F_{p,\alpha}\}$

Figure 31: Algorithme d'Augmentation de l'Efficacité du Diagnosticabilité et Pronosticabilité

3-1- Ensemble de Données du Diagnostic

Les RNNs construits pour assurer l'opération de diagnostic sont développés pour fournir une approximation du paramètre de diagnosticabilité $P(F_\alpha|SP)$ pour chaque graph partiel généré par le SED. Par conséquent, chaque exemple d'entraînement utilisé pour entraîner ce RNN dédié à F_α doit être comme suit :

$$TE_d^n(F_\alpha|SP) = [P_o(SP) \qquad P(F_\alpha|Sp)]$$

De telle sorte que : $P_o(SP) \in \{0,1\}^{1\times(K_d+1)(N+1)}$ est le vecteur formé en utilisant le codage à un coup, qui représente les événements observables du graph partiel SP, et il représente également le vecteur d'entrée du RNN. De plus, $P(F_\alpha|SP) \in [0,1]$ est le vecteur de sortie, qui contient la valeur de diagnostic de l'état de faute F_α. Ainsi, les RNNs avec $(K_d + 1) * (N + 1)$ entrées et 1 sortie représentent les diagnostiqueurs.

3-2- Ensemble de Données de Pronostic

Les RNNs construits pour assurer l'opération du pronostic sont développés pour fournir une approximation du paramètre $Pg(F_\alpha|SP)$ pour chaque graph partiel généré par le SED. Par conséquent, chaque exemple d'apprentissage utilisé pour former ce RNN doit être comme suit :

$$TE_p^n(F_\alpha|SP) = [P_o(SP) \qquad Pg(F_\alpha|SP)]$$

Tel que $P_o(SP)$ est le même vecteur utilisé pour le diagnostiqueur et $Pg(F_\alpha|SP)] \in [0,1]$ représente le vecteur de sortie, qui contient le pronostic ou la probabilité que le SED dévie vers un état de faute pour le mode de faute F_α. Ainsi, les RNNs avec $(K_p + 1) * (N + 1)$ et 1 sortie représentent les pronostiqueurs.

3-3- Ensemble de Données du Temps Restant avant une Faute

Pour le temps restant avant un état de faute, à l'instar de ce qui a été déjà fait pour le diagnostiqueur et le pronostiqueur, on fera de même pour les modèles dédiés à la prévision du temps restant avant une faute. Ainsi, le vecteur d'entrée est le même que précédemment, néanmoins $t_{min}(F_\alpha|SP)$ et $t_{max}(F_\alpha|SP)$ sont respectivement les vecteurs de sortie des RNNs dédiés au calcul du temps minimum et maximum restant pour la déviation vers l'état de faute F_α . Par conséquent, les deux RNNs avec $(K_p + 1) * (N + 1)$ entrées et 1 sortie représentent le calculateur du temps restant avant une faute.

3-4- Algorithme du Diagnostic et du Pronostic des SED par paire temporelle

Comme déjà entamé dans la section 5-6 du chapitre précédent. On va entamer la même chose dans cette section. La mise en place de l'application informatique qui va jouer le rôle d'un module de diagnostic et de pronostic en ligne est décrite par l'algorithme 3. Au meilleur de nos connaissances, cet algorithme est appliqué indépendamment pour chaque classe de faute α.

Notre algorithme prend en entrée les graphes partiels du diagnostic et de pronostique pour chaque classe de faute, et fournit comme sortie les différents indices du diagnostic et de pronostic. Cet algorithme se compose de trois étapes principales :

- **Etape 1 :** L'étape 1 c'est la phase d'initialisation, qui permet d'initialiser les graphes partiels et ses ensembles comme des listes et des ensembles vide respectivement.

- **Etape 2 :** Cette étape consiste en différent sous étapes, au premier temps les module du diagnostic et de pronostique vérifient si le SED a généré un événement observable ou non. Dans le cas de l'occurrence d'un événement observable il sera récupéré directement dans e_i après il sera ajouté a Sp_d_α et Sp_p_α. Dans le cas où Sp_d_α et Sp_p_α ne disposent pas respectivement de $K_{d,\alpha}$ et $K_{p,\alpha}$ événements, la fonction padding, qui permet d'ajouter des événements vides à Sp_d_α et Sp_p_α , sera appliquée. Dans le cas contraire, l'ancien événement de Sp_d_α et Sp_p_α c'est-à-dire $e_{i-K_{d,\alpha}}$et $e_{i-K_{p,\alpha}}$ seront enlevé et le nouveau sera ajouté.

Algorithme 3

Entrés : Sp_d_α ; Sp_p_α
Sorties : $P(Sp_d_\alpha/f_\alpha), Pg(Sp_p_\alpha/f_\alpha), t(Sp_p_\alpha/f_\alpha)$

Etape 1 : $SP_i^j_d_\alpha = []$; $SP_i^j_p_\alpha = []$.
Etape 2 : Si event.disponible () == True
e_i= event.recuperer()
Sp_d_α.append(e_i)
Sp_p_α append(e_i)
Si lenght(Sp_d_α)<$K_{d,\alpha}$
Padding($Sp_d_\alpha, K_{d,\alpha} - \text{lenght}(Sp_d_\alpha)$)
Sinon
Sp_d_α.Supprimer_Premier_element ()
Sp_d_α.Concatenate (e_i)
Si lenght(Sp_p_α)<$K_{p,\alpha}$
Padding($Sp_p_\alpha, K_{p,\alpha} - \text{lenght}(Sp_p_\alpha)$)
Sinon
Sp_p_α.Supprimer_Premier_element ()
Sp_p_α.Concatenate (e_i)
Sp_d_α= Sp_d_α.Apply_one_hot_Encoding ()
Sp_p_α= Sp_d_α.Apply_one_hot_Encoding ()
Sinon : Go to Etape 1
Etape 3 : EV_d_α= [Sp_d_α] ; EV_p_α= [Sp_p_α]
$P(Sp_d_\alpha/f_\alpha) = diag_RNN(Sp_d_\alpha)$
$Pg(Tp/f_\alpha) = diag_RNN(Sp_p_\alpha)$
$t_r^j(Tp/f_\alpha) = diag_RNN(Sp_p_\alpha)$
Go to **Etape 2**

Figure 32: Algorithme du Diagnostic et du Pronostic des SED par graph partiel

- **Etape 3 :** C'est la phase finale de notre algorithme, qui consiste au calcul des indices : Diagnosticabilité, Pronosticabilité, et le temps restant avant une faute pour la classe de faute α.

3-4- Etude de Cas

Dans cette section, nous essayons de mettre en œuvre et en pratique l'approche proposée dans la partie théorique présentée dans les sections précédentes. Le SED choisie est le même présenter dans l'étude de cas

élaboré dans le chapitre 3. Nous définissons $X_N = \{X_1, X_2, X_3, X_4, X_5\}$ et $X_F = \{F_1, F_2, F_3\}$ qui représentent respectivement les ensembles des états normaux et des états de faute. Le tableau ci-dessous (tableau 16) fournit tous les éléments de ces ensembles et leur désignation :

Le SED est présenté par une base de données contenant tous les différents graphes de transition d'état qui décrivent tous ses comportements possibles. L'application de l'algorithme 3 sur notre ensemble de données résulte aux résultats rassemblés dans le tableau ci-dessous (Tableau 17) (les experts du système suggèrent de prendre $K_{min,d} = 2, K_{max,d} = 5, K_{min,p} = 2, \text{et } K_{max,p} = 4$.

Tableau 15 : Désignation des Etat du chariot suspendu

Type de l'ensemble / l'ensemble	**Etat**	**Désignation**
Etat Normal / $\mathbf{X_N}$	X_0	Le chariot est en position de chargement
	X_1	Déplacement à droite
	X_2	Mouvement vers le bas du bras du chariot
	X_3	Mouvement vers le haut du bras du chariot
	X_4	Déplacement à gauche
	X_5	Le produit est dans le bon stock
Etat de Fautes / $\mathbf{F_\alpha}$	F_1	Le chariot rate la bonne position
	F_2	Le chariot est dans une fausse position
	F_3	Le produit est dans le mauvais stock

Tableau 16 : Résultats de l'Application de l'Algorithme 1 sur le Chariot Suspendue

	K=2		K=3		K=4		K=5
	Diagnostic	Pronostic	Diagnostic	Pronostic	Diagnostic	Pronostic	Diagnostic
$\boldsymbol{F_1}$	**27.1%**	**2.6%**	22.2%	2%	18.2%	1.6%	15.4%
$\boldsymbol{F_2}$	**28.6%**	**31.1%**	22.2%	24.2%	18.2%	19.8%	15.4%
$\boldsymbol{F_3}$	5.1%	59.7%	23.7%	**68.7%**	**36.4%**	56.2%	30.8%

Selon les résultats obtenus, on peut observer que l'utilisation de la dimension $K = 2$ augmente les indices de diagnostic et de pronostic pour les états de faute $\boldsymbol{F_1}$ et $\boldsymbol{F_2}$ avec respectivement un indice de diagnostic et de pronostic de 27.1% et 2.6% pour $\boldsymbol{F_1}$ et 28.6% et 31.1% pour $\boldsymbol{F_2}$. Cependant, pour l'état de faute $\boldsymbol{F_3}$, les dimensions $K = 3$ et $K = 4$ augmentent respectivement la diagnosticabilité et la pronosticabilité avec un indice de 36,4% et 68,7%.

3-4-1- La Construction et l'Evaluation des Réseaux de Neurones Récurrents

Après le prétraitement des données (la formation des graphes partiels et le calcul des différents indices), des RNNs ont été développés et différentes valeurs des hyperparamètres (nombre de couches, nombre de nœuds

de chaque couche, taille du lot, nombre d'époques de formation, etc.) et des fonctions d'activation pour chaque couche ont été testées pour trouver celles qui fournissent les meilleures performances, en plus, pour atténuer en même temps le sur- et sous-ajustement des modèles.

Pour une évaluation détaillée des performances et de la qualité des RNNs pendant la formation, la variation de l'EQM pendant la formation et les graphiques de validation ont été utilisés. Les graphiques ci-dessous résument les résultats finaux obtenus :

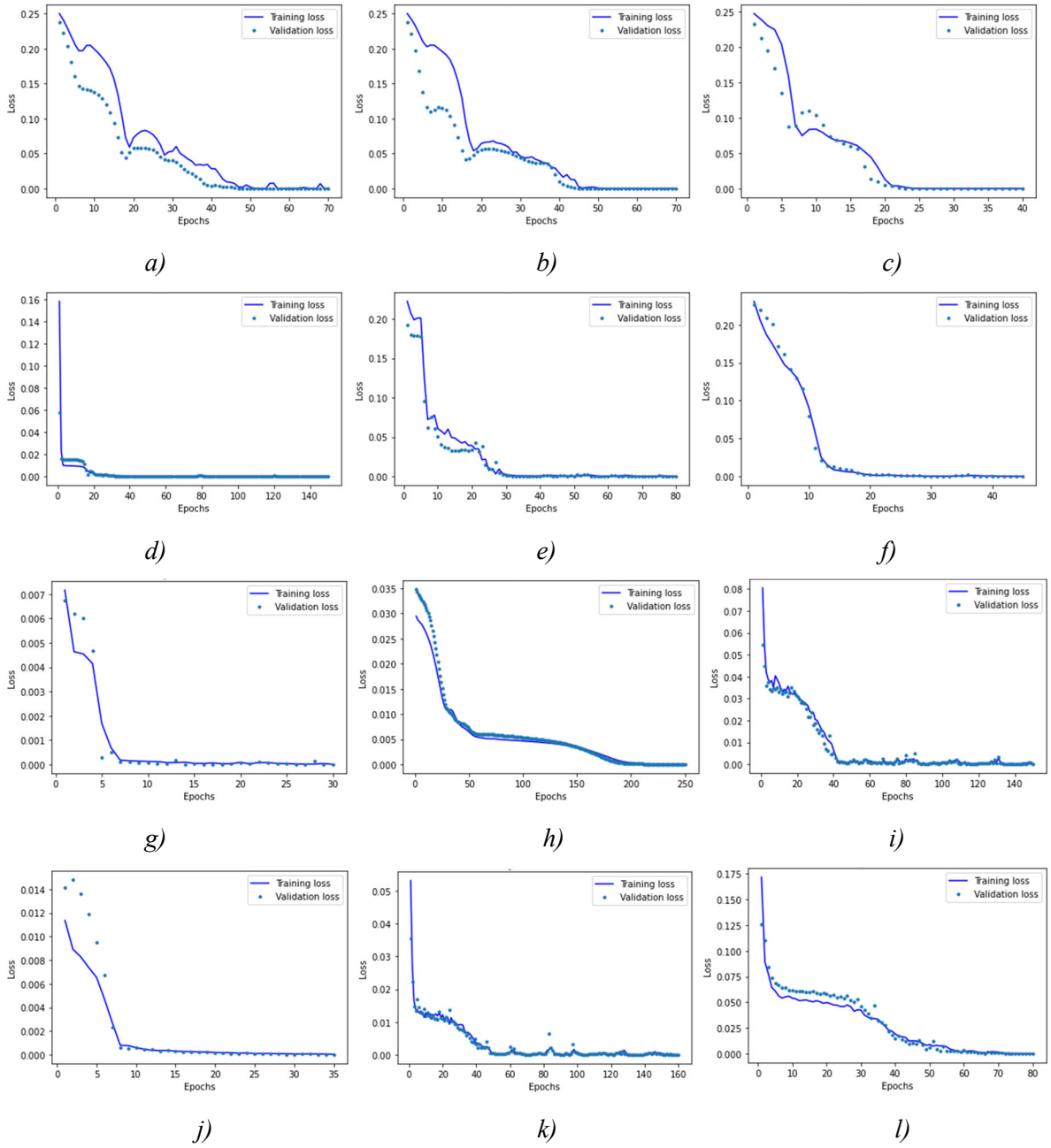


Figure 33: La variation du EQM d'apprentissage et de validation de modèle du diagnostic de F1(a), de F2(b), de F3(c), de pronostic de F1(d), de F2(e), de F3(f), de calcule du temps minimal restant avant F1(g), F2(h), F3(i) et de calcule du temps maximal restant avant F1(j), F2(k), F3(l)

Sur la base des figures suivantes, on peut observer comment l'EQM de la formation et de la validation est minimisée et diminuée pendant l'optimisation des RNNs sans aucune indication ou signe de sur- ou sous-apprentissage des modèles.

Cependant, pour les données de test, le RMSE et le nRMSE (le nRMSE indique la dispersion et la variabilité de la précision des prédictions du RNN) sont choisis pour mesurer la performance des modèles sur les données de test. Un bon modèle est celui pour lequel la RMSE et la nRMSE sont les plus faibles possible.

Selon [127], la précision du modèle basé sur le nRMSE peut être divisée en 4 classes comme le montre le tableau suivant (Tableau 18) :

Tableau 17 : Les Classes du nRMSE

Class	Valeur du nRMSE	Etat de précision
1	nRMSE<10%	Excellent
2	10%<nRMSE<20%	Bon
3	20%<nRMSE<30%	Juste
4	30%<nRMSE	Mauvais

Les résultats de la phase de test et les valeurs du RMSE et du nRMSE de nos modèles sont résumés dans le tableau 19.

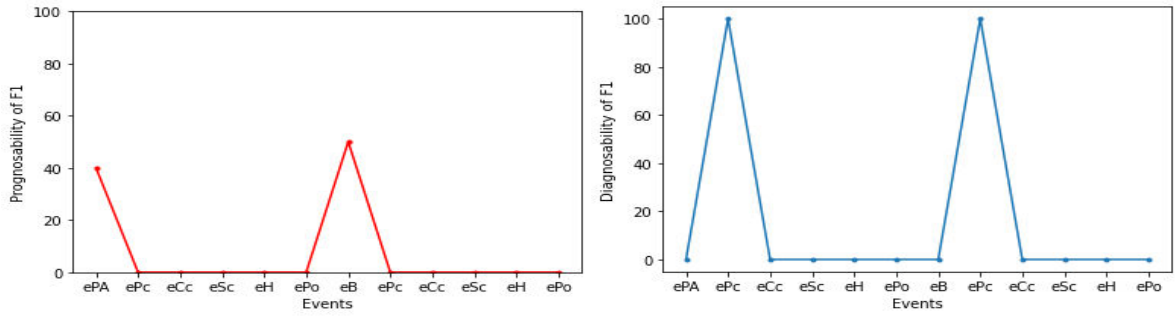
Tableau 18 : la RMSE et la nRMSE des RNNs sur les Données de Test.

Model	RMSE du Test	nRMSE de Test
Diagnostiqueur de F1	$\approx 2{,}66\times10^{-4}$	≈ 0,15%
Diagnostiqueur de F2	$\approx 1{,}16\times10^{-3}$	≈ 1%
Diagnostiqueur de F3	$\approx 6{,}42\times10^{-4}$	≈ 0,27%
Pronostiqueur de F1	$\approx 1{,}86\times10^{-3}$	≈ 4,49%
Pronostiqueur de F2	$\approx 7{,}57\times10^{-3}$	≈ 2,67%
Pronostiqueur de F3	$\approx 1{,}36\times10^{-2}$	≈ 2,48%
Temps restant Min de F1	$\approx 3{,}33\times10^{-2}$	≈ 4,53%
Temps restant Max de F1	$\approx 3{,}7\times10^{-3}$	≈ 6,2%
Temps restant Min de F2	$\approx 6{,}13\times10^{-4}$	≈ 1,98%

Temps restant Max de F2	$\approx 5{,}64\times10^{-4}$	$\approx 7{,}17\%$
Temps restant Min de F3	$\approx 1{,}51\times10^{-2}$	$\approx 7{,}47\%$
Temps restant Max de F3	$\approx 7{,}27\times10^{-3}$	$\approx 2{,}37\%$

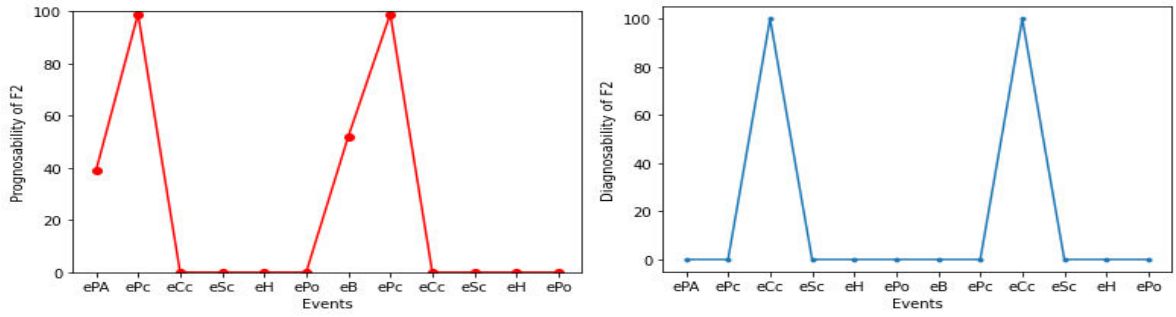
Dans notre cas, le RMSE obtenu pour tous les modèles est très proche de 0, de plus, la valeur nRMSE de nos modèles varie de 0,15% à 7,47% ce qui signifie que nos modèles appartiennent à la première classe, ce qui indique que nos modèles sont performants, ce qui signifie que tous les modèles disposent de grandes capacités prédictives, et peuvent traiter parfaitement les opérations souhaitées.

Enfin, le système est lancé selon les cycles de fonctionnement pour transposer successivement les produits 'A' et 'B', et des fautes sont créées exprès, pour visualiser en pratique, la réponse du diagnostiqueur, du pronostiqueur, et le calculateur du temps restant avant une faute. Les résultats obtenus sont présentés respectivement dans les figures suivantes :



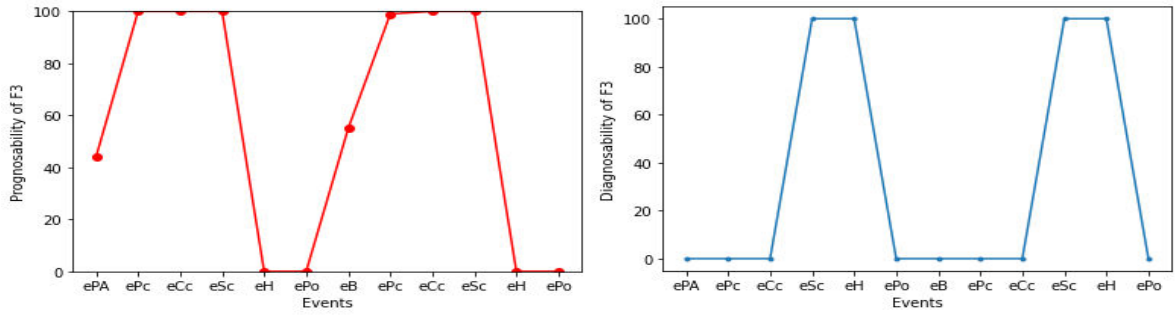
a) b)

Figure 34: Pronostic (a) et Diagnostic (b) de F1



a) b)

Figure 35: Pronostic (a) et Diagnostic (b) de F2



a) b)

Figure 36: Pronostic (a) et Diagnostic (b) de F3

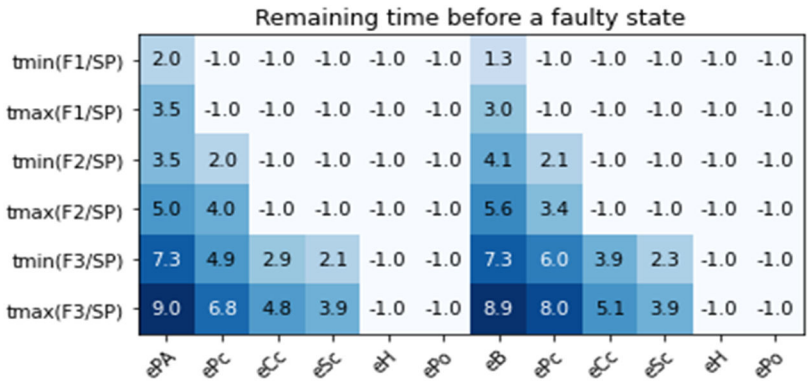


Figure 37: Temps restant avant les fautes

Sur la base de ces résultats, il est clair que les diagnostiqueurs ont détecté avec précision la déviation du système vers les états de faute, en fournissant une probabilité de déviation de 100%, qui sont déjà prédits par le pronostiqueur à un moment antérieur, c'est-à-dire avant même les déviations. De plus, grâce au RNN dédié au calcul du temps restant, il nous a donné une estimation de l'intervalle de temps où cette déviation peut apparaître sous forme de carte de chaleur, de sorte que plus la couleur est foncée, plus la déviation est loin.

4- Conclusion

Le diagnostic et le pronostic des fautes dans les systèmes industriels jouent un rôle vital dans les industries. Les travaux de recherche présentée dans ce chapitre, ont permis de développer une nouvelle approche pour le diagnostic et le pronostic des fautes dans les SEDs indépendamment des outils de modélisation. Dans ce contexte, le comportement du SED est décrit par des graphes partiels contenant les événements générés par le SED. Un algorithme pour déterminer les dimensions de ces graphes partiels, qui maximisent en même temps l'efficacité du diagnostic et du pronostic, en plus du cadre théorique pour construire un diagnostiqueur et pronostiqueur, qui permet l'analyse des big data extraites du modèle statistique du SED pour traiter un diagnostic et un pronostic en ligne. Dans ce but, des RNNs qui fournissent comme sorties respectivement trois indices principaux : Diagnosticabilité, Pronosticabilité, et le temps restant avant une faute, ont été proposé, de manière à identifier l'état actuel et futur probable du SED désiré ainsi que l'intervalle de temps, où le système peut probablement dévier vers un état de faute dans les étapes de fonctionnement à venir.

Dans ces deux derniers chapitres, les approches mentionnées prennent en considération uniquement la nature et l'ordre et les dates d'occurrence des événements générés, qui décrivent le comportement du SED en temps réel, pour effectuer le diagnostic et le pronostique des fautes dans le SED. Cependant, ils ignorent les contraintes temporelles lient les différents évènements générés par le SED. A cette fin, dans le chapitre suivant une approche de diagnostic des SEDs en se basant sur les contraintes temporelles sera élaborée.

Chapitre 5 : Diagnostic des SEDs en se Basant sur les Contraintes Temporelles

Table des Matières

1- Introduction 104
2- Contraintes Temporelles 104
3-La Période comme Contrainte Temporelles 105
3-1- Signatures Temporelles Causales 106
3-2- Approche Proposé 106
3-3- Vérification des contraintes temporelles pour les SEDs 107
3-4- L'extraction des Contraintes Temporelles à Partir du Modèle Statistique 108
3-5- La Perte d'Observabilité et le Blocage du Système 109
3-6- Construction des Réseaux de Neurones 109
3-7- Base de Données des Contraintes Temporelles 110
3-4- Algorithme du Diagnostic des SED par Contraintes temporelles. 111
3-8- Etude de Cas 112
4- Conclusion 116

1- Introduction

Comme mentionné dans le chapitre 3, les systèmes à événements discrets peuvent être confrontés à de nombreux types de fautes. Les trois principaux types sont les fautes de capteurs, de processus et d'actionneur. Par ailleurs, de nombreuses défaillances de ces trois types peuvent représenter une violation des contraintes temporelles, c'est-à-dire des défaillances qui peuvent conduire à une violation des contraintes temporelles, c'est-à-dire la non-exécution de chaque état dans sa marge de temps prévu. Néanmoins, il n'y a que quelques recherches qui traitent cette problématique. Par conséquent, dans ce chapitre, seules les fautes pouvant conduire à une violation des contraintes temporelles, c'est-à-dire les fautes qui se traduisent par un retard, une avance d'exécution ou un blocage au niveau du système seront prises en considération, afin de développer une approche basée sur le modèle statistique et les RdNs « feed-forward » pour assurer un diagnostic en ligne pour la vérification des contraintes temporelles. Dans la suite de ce chapitre, nous supposons les hypothèses suivantes :

- **A1 :** Le SED ne peut pas générer plus d'un événement à un instant t_i, c'est-à-dire que plusieurs événements observables ne peuvent pas se produire simultanément, cette hypothèse permet qu'un seul événement soit généré à un instant t_i.
- **A2 :** Les événements observables se produisent dans leur séquence régulière, c'est-à-dire leur ordre normal, car l'ordre correct ou non de l'occurrence d'une séquence d'événements est vérifié par les approches présentées précédemment.

2- Contraintes Temporelles

Dans la littérature, dans le contexte des SEDs, une contrainte temporelle (nous désignons une contrainte temporelle par TC) est un temps relatif qui sépare un événement e_i d'un autre événement e_j, de sorte que e_i survienne après e_j. Les contraintes temporelles peuvent être de trois types [128] :

- **Date :** L'utilisation de la date comme TC entre l'événement e_i et l'événement e_j permets de représenter l'intervalle de temps exact entre l'apparition de ces deux événements, de sorte que :

$$TC(e_i; e_j) = t$$

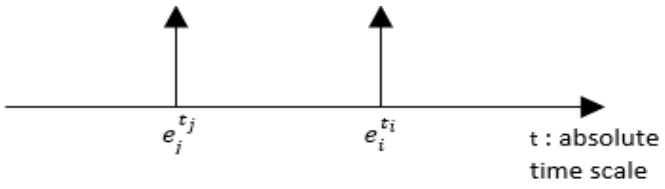


Figure 38: TC basée sur la Date

C'est-à-dire que l'occurrence de l'événement e_i doit se produire exactement à une date t_i de telle sorte que la contrainte temporelle est vérifiée si et seulement si $T(e_i) - T(e_j)$ sera exactement égal à $TC(e_i; e_j)$.

- **Période :** L'utilisation d'une période comme contrainte temporelle entre l'événement e_i et l'événement e_j permets de représenter non pas le temps exact mais le temps avec un degré d'incertitude, c'est-à-dire l'intervalle de temps entre l'occurrence de ces deux événements avec une marge de tolérance, c'est-à-dire que e_i devrait se produire à un intervalle de temps spécifique après e_j dans ce cas :

$$TC(e_i; e_j) = [t_{min}; t_{max}]$$

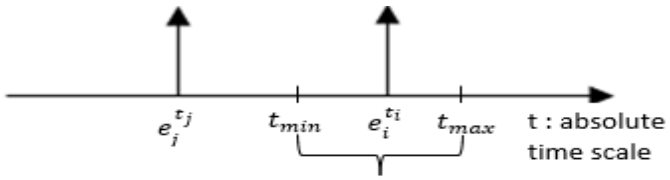


Figure 39: TC Basée sur la Période

Et la contrainte temporelle est vérifiée si et seulement si $T(e_i) - T(e_j) \in [t_{min}; t_{max}]$.

- **Durée :** En utilisant la durée comme contrainte temporelle, il s'agira de modéliser qu'une information doit être vraie ou vérifiée pendant une durée prédéterminée, c'est-à-dire qu'elle doit être vraie à partir d'une date et rester vraie pendant un intervalle de temps. Dans ce cas, on ne peut pas vraiment parler d'un événement mais plutôt de la durée "d" d'un état ou une tache (car en général la durée est dédiée aux évolutions continues), et la contrainte temporelle est vérifiée tant que :

$$d_{min} \leq d \leq d_{max}$$

Ainsi, d_{min} et d_{max} sont des durées minimales et maximales prédéterminées pour une tâche ou un état spécifique.

Les trois contraintes temporelles sont plus ou moins similaires et répondent au même objectif. Cependant, dans notre approche, nous prenons la période comme contraintes temporelles. parce qu'en utilisant la représentation temporelle des événements présentée dans la section 3 du premier chapitre, la période est le moyen le plus approprié pour la vérification des contraintes temporelles plutôt que la date, en plus que la contrainte temporelle basée sur la date est plus stricte et rigoureuse pour la vérification des contraintes temporelles en la comparant par la durée ou la période, ce qui posera un problème dans une vérification en ligne, parce que le temps d'occurrence des événements change selon l'état, le fonctionnement et les performances du SED, ainsi le diagnostiqueur peut générer une alarme même si le SED est dans son fonctionnement normal. Par conséquent, dans le reste de ce chapitre une nouvelle approche pour la vérification des TCs en se basant sur le modèle statistique sera présentées, en se basant sur la période, en vérifiant les TCs entre les évènements.

3-La Période comme Contrainte Temporelles

Avant d'entamer la présentation de l'approche proposée pour la vérification des TCs dans les SEDs basée sur la période, nous allons présenter un nouveau formalisme qui est les signatures temporelles causales.

3-1- Signatures Temporelles Causales

Les signatures temporelles causales ou STC sont des formalismes qui sont généralement utilisé pour décrire et reconnaître les comportements qui ont été utilisés pour le diagnostic des SEDs [128]. Selon Saddem et al, [129], [130] il est défini comme "un sous-ensemble d'événements observables partiellement ordonnés liés par une contrainte temporelle" et comme "la description d'un modèle temporel définissant un ordre partiel sur des événements déterminés par leur type et leur date d'occurrence". Le diagnostic basé sur les STCs vise la vérification des contraintes temporelles du STC de sorte que le STC est reconnu si et seulement si tous ses événements se sont produits dans leur intervalle de temps approprié, c'est-à-dire si et seulement si toutes les contraintes temporelles de tous ses événements sont vérifiées [129], c'est-à-dire qu'il n'y a pas de violation de contrainte temporelle pour aucun événement du STC. Par conséquent, chacun d'entre eux doit être doté de son intervalle de contrainte temporelle. Mais nous pouvons clairement observer qu'un événement e_i peut se produire plusieurs fois dans des phases différentes lorsqu'un mot temporel est généré par le SED. Par conséquent, nous suggérons de diviser chaque mot temporel en signatures temporelles causales (r), de telle sorte que chaque signature temporelle causale soit une séquence d'événements qui ne contiennent pas d'événements similaires. A cette fin, dans ce chapitre, nous allons établir une nouvelle définition de la signature temporelle causale, de sorte qu'à partir du mot temporisé et du modèle statistique, la STC et les contraintes temporelles sont formées pour chaque événement observable du SED.

3-2- Approche Proposé

Notre approche se compose de quatre étapes. La première, la deuxième et la troisième sont hors ligne et la quatrième est en ligne. La quatrième étape consiste à diagnostiquer les défaillances en ligne à l'aide de RdN basés sur le concept de la vérification des contraintes temporelles et du STC. Elle interprète en temps réel les événements générés par le SED étudié afin de détecter toute défaillance qui a provoqué une déviation de la contrainte temporelle. En gros, l'étude vise à répondre à deux questions : Le système a-t-il été confronté à une violation d'une contrainte temporelle ? Si oui, quelle faute est probablement responsable de cette violation ?

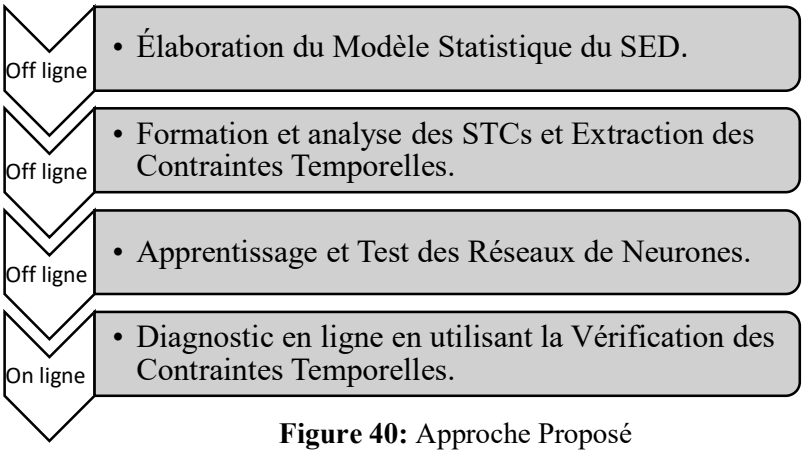


Figure 40: Approche Proposé

Définition 7 : (STCs à partir de mots temporisés) nous considérons un mot temporisé $\sigma_{j'}^{j} \in TW^{j}$ tel que $\sigma_{j'}^{j} = e_1^{t_1}\, e_2^{t_2} \dots e_i^{t_i} \dots e_k^{t_k}$ enregistré à partir du $j'^{ème}$ cycle d'un SED selon le $j^{ème}$ cycle de fonctionnement. Nous définissons une STC comme une séquence d'événements observables, de sorte que le STC ne contienne pas deux événements similaires et que chaque événement e_i soit doté d'un événement de référence $ref(e_i)$ de telle sorte que chaque événement soit directement dans une relation de cause/conséquence effet et lié par une contrainte temporelle avec son événement de référence, et nous définissons ε l'événement vide, qui sera la référence de tous les événements observables qui ne disposent pas d'un événement de cause/conséquence effet, tel que $T(\varepsilon) = 0$. La raison pour laquelle on choisit que les STCs ne contiennent pas d'événement observable similaire est de vérifier la contrainte temporelle pour chaque événement séparément, de manière à définir précisément la contrainte temporelle de chaque événement individuellement. Par conséquent, nous proposons de diviser chaque mot temporisé en STC, tel que chaque STC représente une phase de fonctionnement "r" par rapport à un $\sigma_{j'}^{j}$.

Exemple 12 : Nous considérons un SED défini par $\sum = \{a, b, c, d, f_1, f_2\}$ et nous considérons σ_1^1 un mot temporisé généré par le SED pendant le premier cycle selon le premier cycle de fonctionnement, de sorte que $\sigma_1^1 = b^1\, a^{1,5}\, f_1^{2,3}\, b^5\, f_2^{5,6}\, a^7\, d^{9,1} c^{11} b^{12} f_3^{12,5} a^{15}$ donc, l'ensemble des STCs dérivés de σ_1^1 est le suivant : $STC_{1.1}^1 = b^1\, a^{1,5}\, f_1^{2,3}$, $STC_{1.2}^1 = b^5\, f_2^{5,6}\, a^7\, d^{9,1} c^{11}$, $STC_{1.3}^1 = b^{12} f_3^{12,5} a^{15}$.

3-3- Vérification des contraintes temporelles pour les SEDs

Dans le contexte des SEDs, chaque tâche a un temps d'exécution prédéterminé qui représente une contrainte temporelle que le SED doit respecter pour garantir son bon fonctionnement et qui exige que les événements qui lui sont associés se produisent selon leurs dates prévues. A cette fin, nous associons à chaque événement e_i une contrainte temporelle, liée à son événement de référence $ref(e_i)$, qui est définie comme une période, où l'occurrence de l'événement est requise selon son événement de référence. Plus formellement :

$$\forall \sigma^{j} \in TW^{*},\ \forall e_i \in STC,\ \exists e'_i \in STC' / \ e'_i = ref(e_i) \Leftrightarrow TC_r^j(e_i \mid e'_i) = [P_{min} - \Delta P_i; P_{max} + \Delta P_i]$$

Ainsi, que ΔP_i est une marge de tolérance donnée par les experts, $TC_r^j(e_i \mid e'_i)$ est la période de contrainte temporelle, qui relie l'événement e_i du cycle j à la phase de fonctionnement r avec son événement de référence, P_{min} , P_{max} sont respectivement les périodes de limite de contrôle inférieur et supérieure de la contrainte temporelle, et la contrainte temporelle est vérifiée si et seulement si :

$$(T(e_i) - T(ref(e_i)) \in [P_{min} - \Delta P_i; P_{max} + \Delta P_i]$$

Dans le cas contraire, la contrainte temporelle est violée et le diagnostiqueur génère une alarme afin de vérifier et de corriger le problème.

Exemple 13 : Nous considérons le SED modélisé à l'aide de GRAFCET dans la figure ci-dessous, et défini par les ensembles suivants : $\sum_o = \{\varepsilon, a, b, c, d, e, g\}$, qui exécute 6 tâches T_1, T_2, T_3, T_4, T_5 et T_6, tel que $ref(b) = ref(d) = a$, $ref(c) = b$, $ref(e) = d$, $ref(g) = e$ et $ref(a) = \varepsilon$.

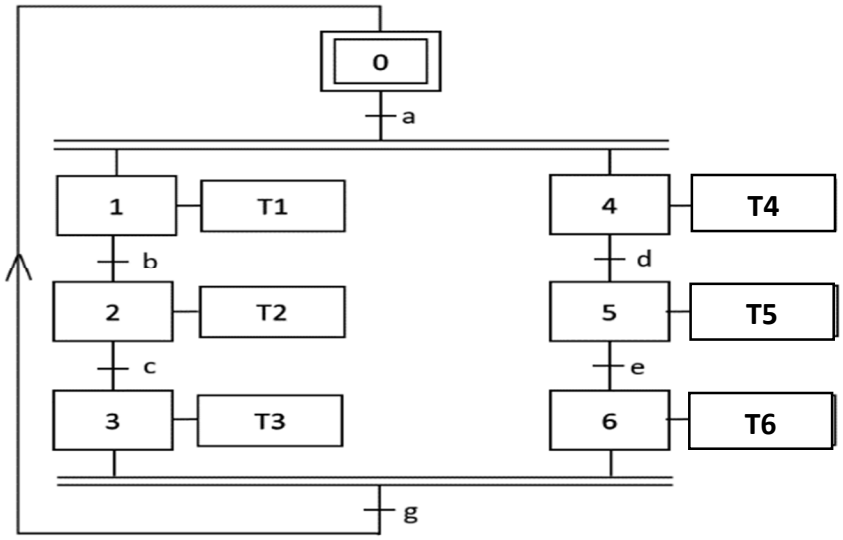


Figure 41: GRAFCET de Fonctionnement d'un SED

Et nous supposons que le mot temporel suivant est enregistré : $\sigma_1^1 = a^0\ b^{1,2}\ d^{3,5} e^6 c^{7,5} g^9$, après l'occurrence de $'a'$ le SED exécute plus d'une tâche en même temps, donc la bonne exécution de chaque tâche est directement liée au respect de la contrainte temporelle de chaque événement. Nous prenons la tâche T_2 comme exemple, la bonne exécution de cette tâche, qui commence par le franchissement de l'événement "b" et terminée par l'événement "c" dépend du temps d'exécution du "c" selon l'événement "b", ainsi tout problème lié au temps que le système peut rencontrer, pendant l'exécution de la tâche T_2, apparaîtra comme une violation de la contrainte temporelle de l'événement "c".

3-4- L'extraction des Contraintes Temporelles à Partir du Modèle Statistique

Comme nous l'avons déjà présenté, le modèle statistique est formé par un grand nombre de mots temporisés collectés pendant le fonctionnement du SED selon différents cycles, et chaque mot temporisé fournit des événements et leur temps d'occurrence. Au cours d'un cycle, la bonne exécution de chaque tâche dépend du temps d'occurrence des événements associés à cette tâche. En d'autres termes, le temps d'occurrence d'un événement e_i dépend de son événement de référence, par conséquent, le modèle statistique du SED doit être rassemblé et analyser les dépendances entre les événements et leur événement de référence associé, afin de corréler chacun d'eux avec sa propre contrainte temporelle.

Remarque 9 : l'élaboration des contraintes temporelles se fait uniquement en analysant les cycles de fonctionnement normal du modèle statistique, les cycles de fonctionnement anormal permettent de distinguer les fautes probables responsables de la violation des TCs.

Exemple 14 : On considère σ_1^1 (Le même présenté dans l'exemple 12) σ_2^1 et σ_3^1 trois mots temporisés collectés selon le cycle de fonctionnement σ^1, tel que $\sigma_2^1 = b^{0,9}\, a^{1,5}\, b^4\, a^{6,4}\, d^{7,5} c^9\, b^{10,5} a^{12}$ et $\sigma_3^1 = b^{1,1}\, a^{1,4}\, b^4\, a^{6,2}\, d^{7,8} c^{8,9} b^{10,4} a^{12}$. Nous prenons $e_i = a; j = 1; r = 1$ le premier évènement $a$généré par le SED selon le premier cycle de fonctionnement. Par conséquent, à partir du modèle statistique enregistré pour le cycle de fonctionnement $j = 1$, qui est composé de trois mots temporisés, nous extrayons la période suivante de l'événement " a " pour $r = 1$. On prend $ref(a) = b$ donc nous obtenons 1,4-1,1=0,3 ; 1,5-0,9=0,6 ; 1,5-1,1=0,4 et posons $\Delta P = 0{,}05$, On obtient $TC_1^1(a) = [0{,}3 - 0{,}05; 0{,}6 + 0{,}05]$. Par conséquent, lorsque le SED fonctionne selon le premier cycle au premier STC, l'événement "a" doit se produire après l'événement " b " entre 0,35 et 0,65 unité de temps, et l'apparition de "a" en dehors de cet intervalle est considérée comme une violation de la contrainte temporelle.

3-5- La Perte d'Observabilité et le Blocage du Système

La majorité des approches proposées dans la littérature sont principalement basées sur la vivacité du système (c'est-à-dire que le système est vivant), ce qui signifie que le SED ne peut pas atteindre un état où l'occurrence d'un événement observable n'est pas possible. Cependant, ce n'est pas le cas dans la pratique, car en pratique le système peut rencontrer différents types de problèmes dus à la perte d'observabilité causée par une faute de capteur, ou au blocage du système causé par des fautes de pré-actionneurs ou d'actionneurs. Ainsi, nous prenons comme exemple le SED présenté dans la figure 41. Nous supposons que le capteur associé et responsable de l'événement "b" est passé d'un événement observable à un événement non observable, ce qui entraîne une réponse erronée sur l'état actuel du SED, c'est-à-dire que le SED se trouve toujours à la place 1, alors qu'il peut se trouver à l'une des places postérieures à 1 ou bloquées. Cependant, selon l'approche proposée dans cette section, en calculant le temps écoulé à la place 1 (c'est-à-dire le temps écoulé après l'occurrence de l'événement "a") si celui-ci dépasse une valeur prédéterminée et spécifique (elle sera détaillée dans la section sur les vecteurs d'entrée), nous pouvons distinguer si le système est confronté à une perte d'observabilité causée par la défaillance d'un capteur ou si le système a été bloqué à une place à cause d'une faute dans l'actionneur ou le pré-actionneur.

3-6- Construction des Réseaux de Neurones

L'exploitation des RdNs pour le diagnostic des SEDs sous contraintes temporelles sera considérées comme une sorte ou un type de problème de classification tel que pour chaque vecteur d'entrée, qui contient l'événement courant "e_i" généré par le SED ainsi que l'indices "j" du cycle de fonctionnement, et les indices " r " de la signature temporelle causale où l'événement "e_i" est associé, en plus d'un indice " P " (il sera détaillé dans les sections suivantes), le RdN fournira si le SED est confronté à une violation de la contrainte temporelle ou non en fournissant la probabilité des fautes, qui peuvent être responsables de la violation de la contrainte temporelle, qui sont définies précédemment. L'architecture de réseau de neurones choisie pour traiter cette opération est le RBFNN.

Le RBFNN sera utilisé pour fournir trois types de classes :

- **Classe 1 :** la contrainte temporelle est vérifiée,
- **Classe 2 :** la violation de la contrainte temporelle en fournissant la faute responsable de cette violation,
- **Classe 3 :** le SED est bloqué.

La figure ci-dessous (figure 42) montre comment le RBFNN peut être utilisé pour la vérification des contraintes temporelles.

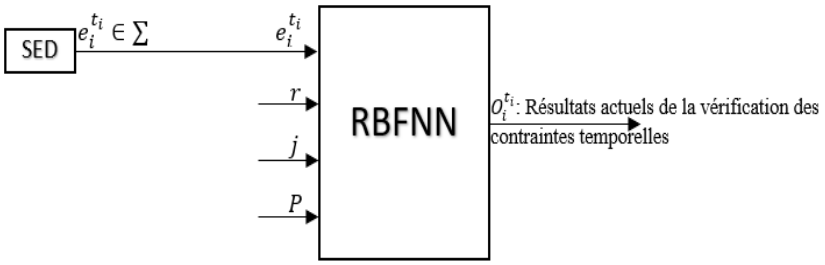


Figure 42: Vérification des TCs par RBFNN

3-7- Base de Données des Contraintes Temporelles

Le RBFNN a été développé pour fournir des approximations du degré de vérification des contraintes temporelles en se basant sur le période pour chaque événement généré par le SED en temps réel, afin de gérer l'opération de diagnostic du SED sous contrainte temporelle. Par conséquent, chaque exemple d'apprentissage pour chaque événement à un cycle spécifique "j" et CTS "r" utilisé pour entraîner le RBFNN devraient être le suivant :

$$TE_r^j(e_i) = \begin{bmatrix} e_i \\ j \\ r \\ P \end{bmatrix} \qquad TC_r^j(e_i)$$

Tel que e_i est un évènement observable, "j" est le cycle de fonctionnement et "r" est l'index de CTS où e_i appartient et "P" est le paramètre de la période de l'évènement e_i. $TC_r^j(e_i)$ est la sortie du RBFNN.

En cas de blocage du SED, l'événement considéré ne se produira pas et le vecteur d'entrée ne pourra pas être complété. Par conséquent, nous définissons une fonction de marquage qui permet de pouvoir construire le vecteur d'entrée dans n'importe quelle situation, telle que :

$$(e_i, P) = \begin{Bmatrix} V(e_i),\ P = 1\ si: ifoccur(e_i) = 1\ et\ P_{min} - \Delta P_i \le T(e_i) - T\big(ref(e_i)\big) \le P_{max} + \Delta P_i \\ V(e_i),\ P = 0\ si: ifoccur(e_i) = 1\ et\ (T(e_i) - T\big(ref(e_i)\big) > P_{max} + \Delta P_i\ ou\ T(e_i) - T\big(ref(e_i)\big) < P_{min} - \Delta P_i \\ V(\varepsilon), P = -1\ si: ifoccur(e_i) = 0\ et\ Cl(DES) - T(e_{i-1}) > P_{lim} \end{Bmatrix}$$

Tel que : $ifoccur(e_i)$ and $V(e_i)$ sont deux fonctions de marquage, de telle sorte que :

- $ifoccur(e_i)$: fournit deux résultats possibles (1 et 0) : 1 si l'événement e_i s'est produit (à sa période prédéterminée ou non) et 0 si e_i ne s'est pas produit (Par défaut $ifoccur(e_i) = 0$).

- $V(e_i)$: la fonction qui applique le codage à un coup pour convertir l'événement généré par le SED en un vecteur d'entrée.
- P_{lim} : Une période limite, qui sera utilisée pour reconnaître le blocage du SED.

3-4- Algorithme du Diagnostic des SED par Contraintes temporelles.

Comme déjà entamé dans la section 3-4 du chapitre 4 et 5-6 du chapitre 3. On va entamer la même chose dans cette section. La mise en place de l'application informatique qui va jouer le rôle d'un module de diagnostic et en ligne est décrit par l'algorithme 4. Au meilleur de nos connaissances, cet algorithme est appliqué indépendamment pour chaque classe de faute α.

Algorithme 4

Inputs: e_i ; r , j, p
Outputs: $TC_r^j(e_i)$

Step 1 : $j = 0; r = 1; i = 1; STC_r^j = []$
Step 2 : Update j
Step 3 : $Clk = SED.clock()$
if event.disponible () == True
e_i= event. collect ()
$t = T(e_i)$
if $e_i \notin STC_r^j$
STC_r^j.append(e_i)
Continue
else
r++
STC_r^j=[]
STC_r^j.append(e_i)
Continue
$ref(e_i)$= e_i.determine_ref()
$tr = T(ref(e_i))$
Determine $((e_i, P))$
else:
Go to step 1
Step 4 : EV= $[e_i; j; r; p]$
$TC = diag_NN_TC(EV)$
Go to step 2

Figure 43: Algorithme du Diagnostic et du Pronostic des SED par graph partiel

Notre algorithme prend en entrée l'événement en cours, les indices r , j et p du diagnostic pour chaque classe de faute, et fournit comme sortie les différents indices du diagnostic et de pronostic. Cet algorithme se compose de trois étapes principales :

- **Etape 1 :** L'initialisation du cycle de fonctionnement par 0 car nous ne connaissons pas encore suivant quel cycle de fonctionnement le SED va opérer, l'initialisation de l'indice r et i par 1 et de STC_r^j par une liste vide.

- **Etape 2 :** Le cycle j est soit réglé manuellement, soit un événement prédéterminé peut nous donner la possibilité de savoir dans quel cycle le SED fonctionne. En général, la mise à jour de j sera assurée par un algorithme indépendant qui s'appuie directement sur le SED étudié.
- **Etape 3 :** Cette étape consiste en différent sous étapes, au premier temps le module du diagnostic vérifie si le SED a généré un événement observable ou non. Dans le cas de l'occurrence d'un événement observable il sera récupéré directement dans e_i ainsi que récupérer ça date d'occurrence, une vérification s'il y a un événement observable similaire dans STC_r^j afin d'incrémenter l'indice r ou non est faite, finalement l'indice p est déterminer.
- **Etape 4 :** C'est la phase finale de notre algorithme, qui consiste à la vérification de la contrainte temporelle**.**

3-8- Etude de Cas

Pour mettre en évidence l'importance de l'approche décrite dans le cadre théorique présenté dans les sections précédentes, une étude de cas a été élaborée, dans laquelle nous tentons de mettre en pratique la méthode proposée. Le SED ou on va appliquer notre approche est le même utilisé dans les études de cas illustrées dans les chapitres précèdent, juste que dans cette partie juste les fautes menantes à des violations des contraintes temporelles sont pris en considération, dans ce cas nous posons $\sum_f$={f_1, f_2, f_3, f_4}, ainsi que chaque évènement est doté d'un événement de référence. Le tableau 20 représente les événements et leurs références ainsi que la désignation des événements de faute considérés dans cette section.

Tableau 19 : Désignation des Événements de Référence et des fautes relatives à des violation des contraintes temporelles

Ensemble	**Type d'évènement**	**Evènement (e_i)**	**Événements de Référence ($ref(e_i)$) / Désignation**
$\sum_o$	Événements Observables	e_{P_o}	$ref(P_o) = \varepsilon$
		e_A, e_B, e_C, e_D, e_E	$ref(e_A) = ref(e_B) = ref(e_c) = ref(e_D) = ref(e_E) = P_o$
		$e_{P_A}, e_{P_B}, e_{P_C}, e_{P_D}, e_{P_E}$	$ref(e_{P_A}) = ref(e_{P_B}) = ref(e_{P_C}) = ref(e_{P_D}) = ref(e_{P_E}) = P_o$
		$e_{S_A}, e_{S_B}, e_{S_c}, e_{S_D}, e_{S_E}$	$ref(e_{C_A}) = e_{P_A}$, $ref(e_{C_B}) = e_{P_B}$, $ref(e_{C_C}) = e_{P_C}$, $ref(e_{C_D}) = e_{P_D}$, $ref(e_{C_E}) = e_{P_E}$.
		$e_{C_A}, e_{C_B}, e_{C_C}, e_{C_D}, e_{C_E}$	$ref(e_{C_A}) = e_{P_A}$, $ref(e_{C_B}) = e_{P_B}$, $ref(e_{C_C}) = e_{P_C}$, $ref(e_{C_D}) = e_{P_D}$, $ref(e_{C_E}) = e_{P_E}$.
		e_h	Cycle 1 : $ref(e_h) = e_{S_A}$ Cycle 2 : $ref(e_h) = e_{S_B}$ Cycle 3 : $ref(e_h) = e_{S_C}$ Cycle 4 : $ref(e_h) = e_{S_D}$

			Cycle 5 : $ref(e_h)$=e_{S_E}
$\sum_f$	Événements Non-Observables	f_1	Défaillance du capteur
		f_2	Problème du pré-actionneur responsable du mouvement horizontal.
		f_3	Problème du pré-actionneur responsable du mouvement vertical.
		f_4	L'actionneur responsable du mouvement horizontal est rapide ou lent
		f_5	L'actionneur responsable du mouvement vertical est rapide ou lent
		f_6	Le SED est bloqué

Ce SED est présenté par une base de données qui contient tous les comportements potentiels du SED liés à la violation ou non des contraintes temporelles pour chaque cycle de fonctionnement. Cette base de données est divisée en STCs et toutes les contraintes temporelles ont été réglées afin de l'adapter à notre approche. En suivant ces étapes de prétraitement des données, nous obtenons un ensemble de données d'environ 300 points de données. Un RBFNN a été entraîné de la même manière que celle présentée dans l'étude de cas présentée dans le troisième chapitre. Les paramètres de ce RBFNN sont présentés dans le tableau ci-dessous (Tableau 21).

Tableau 20 : Paramètres du RBFNN pour la vérification des TCs

IL	HL	OL	AF	MNN	SP	Goal
26	250	7	Gaussian RBF/Linear	300	1	$2x10^{-6}$

La variation de l'EQM d'entrainement du modèle a été utilisé pour évaluer la performance et l'efficacité du RBFNN développé.

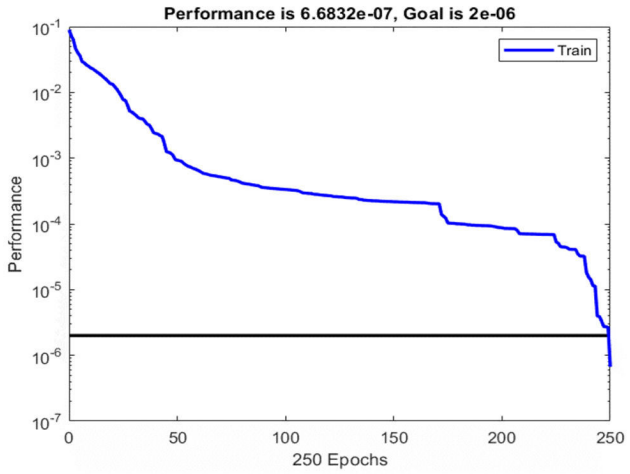


Figure 44: Performances d'apprentissage du RBF de la Vérifications des TCs

La figure suivante (figure 44) représente les performances du RBFNN utilisé, ces figures montrent comment le EQM est minimisé pendant l'apprentissage du RBFNN sans aucune indication ou signe de sous-apprentissage ou sur-apprentissage des modèles. Nous pouvons clairement observer que les meilleures valeurs de l'EQM se produisent dans les 250 dernières époques du cycle d'apprentissage, avec une EQM associée d'environ $6{,}68\text{x}10^{-7}$, il est évident que les valeurs de l'EQM sont plus faibles que les valeurs de l'EQM ciblées, ce qui indique que les cibles de formation sont parfaitement estimées. Après l'opération de formation, le RBFNN est testé sur des données composées de près de 40 points. Afin de mesurer les performances du RBFNN sur les données testées, nous avons utilisé les graphiques de régression linéaire. Les résultats du processus de test sont représentés dans la figure 45.

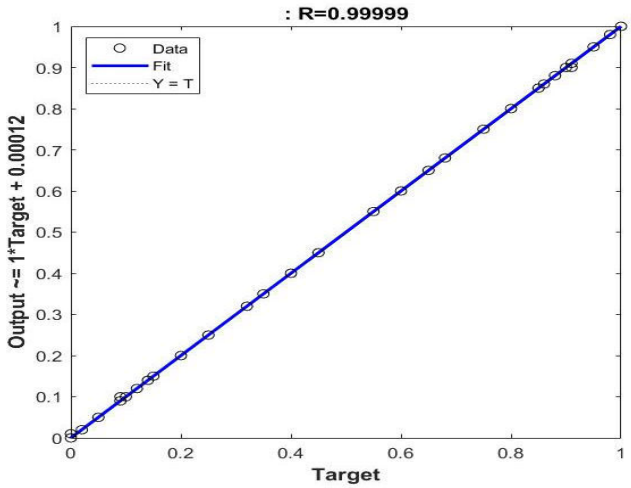
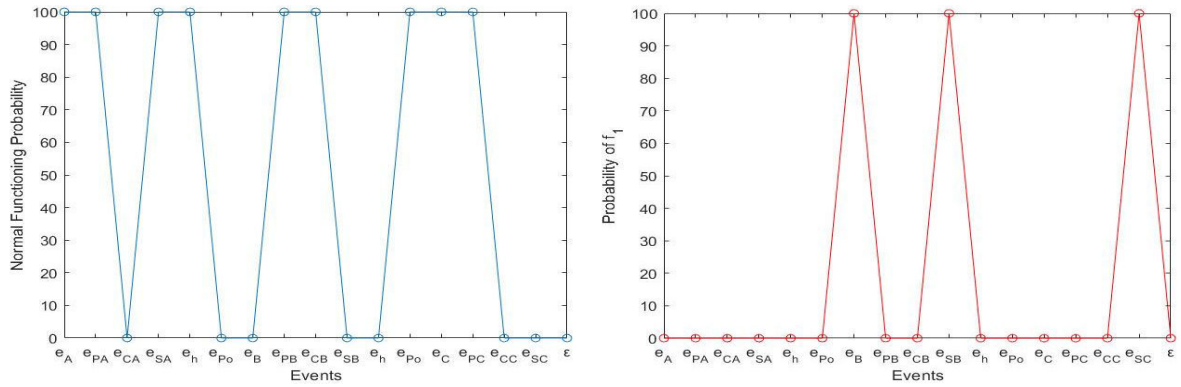


Figure 45: Performance de Test du RBF de la Vérification des TCs

Selon le graphe de régression présenté dans la figure 44, nous pouvons clairement observer que la valeur de régression est très proche de 1 (R=0,99999), c'est-à-dire que pratiquement tous les points de données (cercles) se situent sur la ligne de 45°, ce qui indique que le RBFNN fournit une excellente capacité de prédiction et peut parfaitement gérer l'opération de vérification des contraintes temporelles pour le SED étudié.

Après la construction du RBFNN approprié, nous lançons le chariot afin de transférer le produit 'A', 'B', et 'C'et nous créons exprès une violation des contraintes temporelles afin de visualiser en pratique la réponse du vérificateur de contraintes temporelles. Les résultats obtenus sont présentés respectivement dans la figure 46.



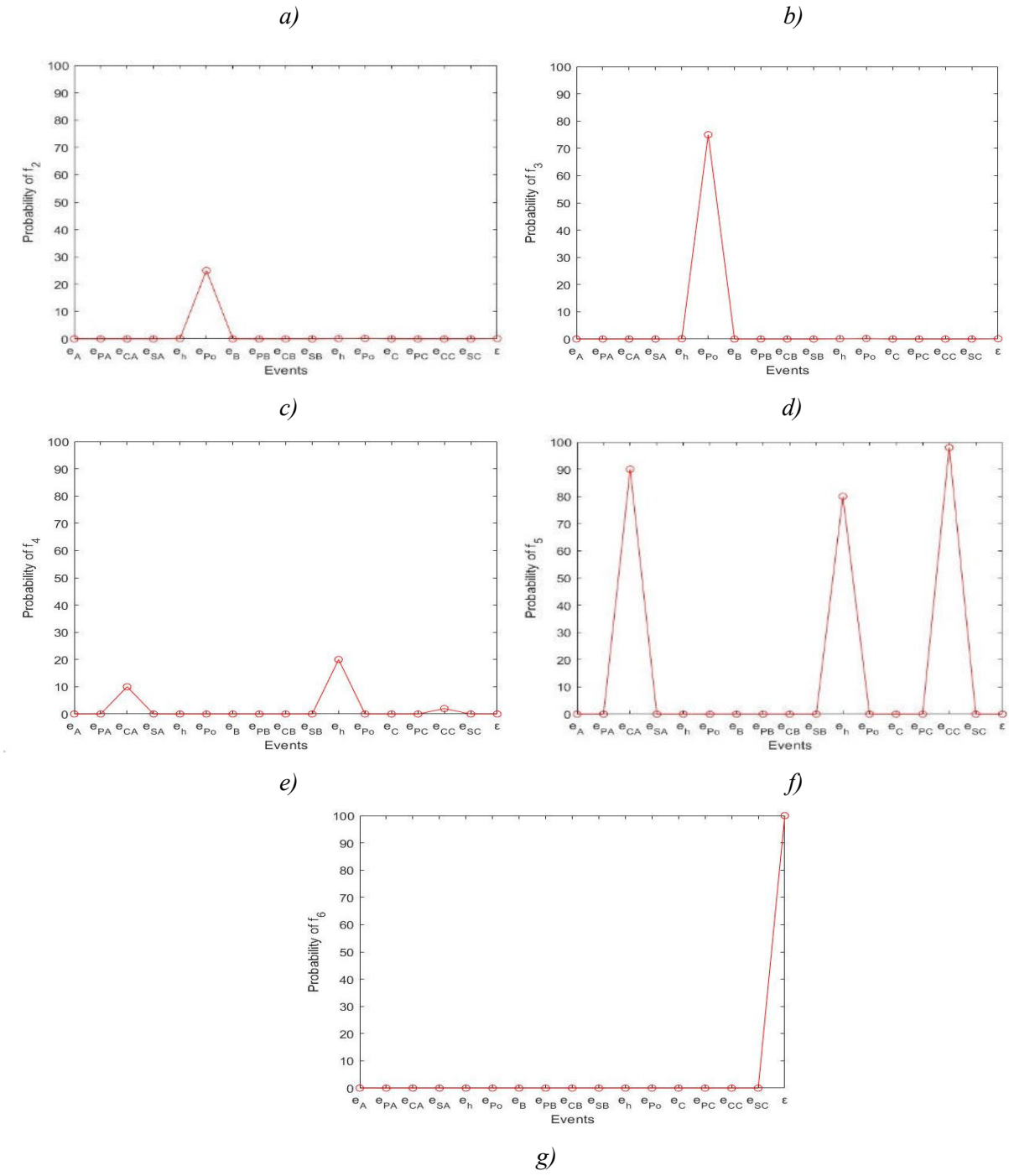


Figure 46: Résultats du Diagnostic du SED en se Basant sur la Contrainte Temporelle de la Période

Selon ces résultats, il est évident que le vérificateur a détecté avec précision la violation des contraintes temporelles relative à la période dans plusieurs phases du fonctionnement du SED en fournissant la probabilité des fautes responsables de la violation des TCs, en plus de la détection du blocage du système après le dernier événement observable généré par le SED.

4- Conclusion

Ce chapitre répond à une collaboration entre trois éléments principaux afin de proposer une nouvelle approche et un nouveau formalisme pour le diagnostic des systèmes à événements discrets sous contraintes temporelles. Dans un premier temps, nous avons proposé une nouvelle définition de la signature temporelle causale afin de résoudre le problème de la similarité entre les événements d'un cycle de fonctionnement, puis le modèle statistique du SED a été traduit en signatures temporelles causales afin d'extraire les dépendances entre les événements pour doter chaque événement de sa contrainte temporelle. Enfin, nous avons présenté le cadre théorique de la construction d'un vérificateur basé sur un réseau de neurones à fonction de base radiale, qui permet l'analyse des grandes données extraites du modèle statistique du SED pour procéder à une vérification en ligne des contraintes temporelles.

Conclusion Général et Perspectives

Le développement et l'élaboration d'approches et d'outils de surveillance pour les systèmes industriels sont l'une des préoccupations majeures des industries, ainsi que des chercheurs dans le domaine. Ce type d'outil permet de superviser et de surveiller l'état de fonctionnement du système ainsi que de détecter les éventuels comportements anormaux et déviations, afin de préserver ces systèmes de commettre des fautes dangereuses qui peuvent mettre en danger la sécurité des hommes est des équipements. Les travaux présentés dans cette thèse font partie de ce contexte. Nous nous sommes intéressés principalement au diagnostic et pronostic des systèmes à événement discret. Notre travail de recherche a permis de développer des nouvelles approches de diagnostic et de pronostic des fautes dans les SEDs en se basant sur les Réseaux de neurones, cet outil nous a permis d'assurer les opérations du diagnostic et de pronostic indépendamment des outils de modélisation qui sont généralement confrontés à divers écueils et difficultés, notamment lorsqu'il s'agit des systèmes industriels larges et complexes qui ont reconnu dernièrement un grand progrès technologique. Les principales contributions de notre travail sont les suivantes :

- **L'utilisation des Réseaux de neurones de types 'Feed-forward' pour le Diagnostic et le Pronostic des SEDs.**

Le comportement du SED est décrit par des paires temporelles contenant les deux évènements observables successifs générés par le SED couplé avec leurs dates d'occurrences. Dans cette approche, nous avons proposé de diviser chaque mot temporelle en sous-mots temporelles ou rangés (r), de sorte que chaque sous-mot temporelle représente une phase de fonctionnement par rapport à un σ_j^j, et cette partition sera faite de telle sorte qu'un sous-mot temporelle ne doit pas contenir des paires temporelles avec la même projection P_o, l'intégration de l'indice (r) a permis de donner au RdN la capacité de distinguer dans quelle phase de fonctionnement le SED fonctionne, ainsi de permettre au RdN de présenter un comportement dynamique. A la base de ça, la probabilité d'occurrence des fautes dans la dernière paire temporelle (Diagnostic), et leurs probabilités d'occurrence dans les prochains paires temporelles (Pronostic), ainsi que le temps restant avant leurs occurrences, sont calculés. Des réseaux de neurones à base de fonction radiale (RBFNN) sont utilisés pour apprendre les différentes relations existantes entre ces paires temporelles et les différents paramètres de diagnostic et de pronostic.

Notre approche est similaire à celle basée sur les fenêtres temporelles propose dans la littérature, dans cette approche, les auteurs ne font pas de distinction entre les fenêtres temporelles similaires et ils les diagnostiquent et les pronostiquent de la même manière même si elles appartiennent à des phases différentes du cycle de fonctionnement, cependant, dans notre approche et grâce à la définition proposée nous avons résolu le

problème de similarités entre les paires temporelles, ce qui nous a permis de diagnostiquer et pronostiquer chacune d'entre elles indépendamment des autres et cela a été prouvé par une étude de cas sur un SED réel.

- **L'utilisation des Réseaux de neurones de types 'Récurent ' pour le Diagnostic et le Pronostic des SEDs.**

Dans cette contribution, Le comportement des SEDs est décrit par des Graphs Partiels contenant les $'K+1'$ dernier événements généré par le SED, couplé avec leurs dates d'occurrences. Un algorithme qui permet de calculer les dimensions de ces Graphs Partiels qui permettent de maximiser l'efficacité des opérations de diagnostic et de pronostic est proposé. A la base de ça, la probabilité que le SED peut dévier de son comportement normal à un comportement de faute après l'observation du dernier événement observable, tout en tenant compte des derniers événements "K" survenus par le SED (Diagnostic), ainsi que la probabilité que le SED dévie vers un mode de faute, après l'apparition successive de chacun des événements "K" immédiatement postérieurs au Graph Partiel actuelle, sont calculées. Des couches récurrentes du types LSTM ont été entrainée pour apprendre les différentes relations existantes entres les Graph Partiel et les différents paramètres de diagnostic et de pronostic.

Cette approche est similaire aux approches de diagnostic et de pronostic basées sur les automates finis présenté dans la littérature (Approche diagnostiqueur), mais l'application de notre approche a permis de fournir un résultat nuancé, c'est-à-dire, dans tous les cas il fournit une probabilité qui donne un aperçu global de l'état courant et futur probable du fonctionnement du système, ce qui semblait clair et évident dans l'étude de cas présenté.

- **Diagnostic des SEDs en se Basant sur les Contraintes Temporelles**

Les résultats de diagnostic et présentés dans les deux contributions précédentes ont été améliorés en intégrant les contraintes temporelles pour chaque événement généré par le SED. La notion des contraintes temporelles a été abordée, ceci a constitué une notion très intéressante pour la détection de quelque type des fautes comme la perte d'observabilité des évènements observables, le blockage du système et les fautes qui peuvent être le résultat d'un retard ou d'une exécution en avance d'un évènement. La contrainte temporelle analysé est la contrainte de la période entre les événements, de tel sorte que chaque événement a été doté d'un événement de référence de telle sorte que chaque événement soit directement dans une relation de cause/conséquence effet et lié par une contrainte temporelle avec son événement de référence. Les dépendances entre chaque événement et son événement de référence ont été analyser et les contraintes temporelles ont été fixé en se basant sur les cycles de fonctionnement normal du modèle statistique du SED.

Au terme de cette thèse, plusieurs perspectives de recherche se dégagent. Elles visent à améliorer et à élargir les résultats produits. Parmi eux, six aspects fondamentaux ressortent :

- L'application de ces approches à un système industriel concret pour récupérer et extraire les résultats d'une manière empirique.
- Intégrer les lois de fiabilité, ce qui permettra à notre approche de prendre en considération, les degrés de fiabilité de chaque composant particulier, de sorte qu'en fonction des indices de diagnostic et de pronostic de chaque faute et la fiabilité de ses composants appropriés, un résultat amélioré du diagnostic et du pronostic sera fourni.
- L'application des Réseaux de neurones à convolution (CNN) pour le diagnostic et le pronostic des SEDs. Dans ce cas les Graphs Partiels seront représentés sous forme des images à deux dimensions et ses images seront les entrées du CNN afin d'assurer un diagnostic et un pronostic en ligne.
- L'intégration des phénomènes stochastiques pour améliorer les résultats du Diagnostic et du Pronostic présenter dans cette thèse. Dans ce cas, les mots temporisés seront considérés comme un processus stochastique et les événements de faute comme des variables aléatoires.
- Développement d'une approche d'estimation d'état des SEDs non-déterministe en utilisant le modèle statistique et les réseaux de neurones récurrents.
- Le développement d'une approche de vérification des contraintes temporelles en se basant sur la contrainte de la durée, dans ce cas, chaque état du SED aura une durée estimée pour l'exécution des tâches qui sera extraite à partir du modèle statistique du système, ainsi qu'une marge de tolérance qui peut être choisie par les experts du système et qui indique une contrainte temporelle que le SED doit respecter pour fonctionner normalement.

Références

[1] F. T. S. Chan, H. C. W. Lau, R. W. L. Ip, H. K. Chan, and S. Kong, "Implementation of total productive maintenance: A case study," *Int. J. Prod. Econ.*, vol. 95, no. 1, pp. 71–94, Jan. 2005, doi: 10.1016/j.ijpe.2003.10.021.

[2] "Qu'est ce que la maintenance 4.0 ?," *Flexio - créer vos applications en mode agile*, May 25, 2019. https://www.flexio.fr/maintenance-usine-4-0/ (accessed Aug. 08, 2022).

[3] Y. Peng, M. Dong, and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: a review," *Int. J. Adv. Manuf. Technol.*, vol. 50, no. 1, pp. 297–313, Sep. 2010, doi: 10.1007/s00170-009-2482-0.

[4] R. Ammour, "Contribution au diagnostic et pronostic des systèmes à évènements discrets temporisés par réseaux de Petri stochastiques," *Autom. Normandie Univ. Fr. FfNNT 2017NORMLH21ff Fftel-01720312ff*, p. 167, 2017.

[5] D. Racoceanu, "Contribution à la Surveillance des Systèmes de Production en Utilisant l'Intelligence Artificielle," *Autom. Robot. Univ. Franche-Comté Fftel- 00011708ff*, p. 246, 2006.

[6] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, Aug. 2015, pp. 2147–2152. doi: 10.1109/FSKD.2015.7382284.

[7] A. Marhaug and P. Schjølberg, "Smart Maintenance-Industry 4.0 and Smart Maintenance: from Manufacturing to Subsea Production Systems," presented at the 6th International Workshop of Advanced Manufacturing and Automation, Nov. 2016, pp. 47–54. doi: 10.2991/iwama-16.2016.9.

[8] R. Zemouri, "Contribution to the monitoring of production systems using dynamic neural networks: Application to e-maintenance," Nov. 2003.

[9] D. Racoceanu, "Contribution à la Surveillance des Systèmes de Production en Utilisant l'Intelligence Artificielle," thesis, Université de Franche-Comté, 2006. Accessed: Jul. 07, 2022. [Online]. Available: https://tel.archives-ouvertes.fr/tel-00011708

[10] G. W. Vogl, B. A. Weiss, and M. Helu, "A review of diagnostic and prognostic capabilities and best practices for manufacturing," *J. Intell. Manuf.*, vol. 30, no. 1, pp. 79–95, Jan. 2019, doi: 10.1007/s10845-016-1228-8.

[11] "5G and Its Impact on The Internet of Things." https://www2.stardust-testing.com/en/5g-and-impact-on-iots (accessed Aug. 08, 2022).

[12] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989, doi: 10.1109/5.21072.

[13] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, 2. ed., corr. At 2. print. New York, NY: Springer, 2010.

[14] N. Malki, "Contribution au diagnostic des Systèmes à Evénements Discrets par modèles temporels et distributions de probabilité," p. 160.

[15] F. Liu, “Predictability of Failure Event Occurrences in Decentralized Discrete-Event Systems and Polynomial-Time Verification,” *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 498–504, Jan. 2019, doi: 10.1109/TASE.2018.2868330.

[16] J. Zaytoon and S. Lafortune, “Overview of fault diagnosis methods for Discrete Event Systems,” *Annu. Rev. Control*, vol. 2, no. 37, pp. 308–320, 2013, doi: 10.1016/j.arcontrol.2013.09.009.

[17] A. Vignolles, E. Chanthery, and P. Ribot, “An overview on diagnosability and prognosability for system monitoring,” *Proc. Eur. Conf. PHM Soc.*, vol. 5, p. 12, 2020.

[18] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, “Diagnosability of discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 40, no. 9, pp. 1555–1575, Sep. 1995, doi: 10.1109/9.412626.

[19] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. C. Teneketzis, “Failure diagnosis using discrete-event models,” *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 2, pp. 105–124, Mar. 1996, doi: 10.1109/87.486338.

[20] K. Zhang, “A Unified Method to Decentralized State Detection and Fault Diagnosis/prediction of Discrete-event Systems,” *Fundam. Informaticae*, vol. 181, no. 4, pp. 339–371, Jan. 2021, doi: 10.3233/FI-2021-2062.

[21] M. Dotoli, M. Pia Fanti, A. M. Mangini, and W. Ukovich, “Identification of the unobservable behaviour of industrial automation systems by Petri nets,” *Control Eng. Pract.*, vol. 19, no. 9, pp. 958–966, Sep. 2011, doi: 10.1016/j.conengprac.2010.09.004.

[22] M. Manyari-Rivera, J. C. Basilio, and A. Bhaya, “Integrated Fault Diagnosis Based on Petri Net Models,” in *2007 IEEE International Conference on Control Applications*, Oct. 2007, pp. 958–963. doi: 10.1109/CCA.2007.4389357.

[23] J. Prock, “A new technique for fault detection using Petri nets,” *Automatica*, vol. 27, no. 2, pp. 239–245, Mar. 1991, doi: 10.1016/0005-1098(91)90074-C.

[24] A. Ramirez-Trevino, E. Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, “Online Fault Diagnosis of Discrete Event Systems. A Petri Net-Based Approach,” *Autom. Sci. Eng. IEEE Trans. On*, vol. 4, pp. 31–39, Feb. 2007, doi: 10.1109/TASE.2006.872120.

[25] J. Lunze and J. Schröder, “State Observation and Diagnosis of Discrete-Event Systems Described by Stochastic Automata,” *Discrete Event Dyn. Syst.*, vol. 11, no. 4, pp. 319–369, Oct. 2001, doi: 10.1023/A:1011273108731.

[26] P. Bouyer, F. Chevalier, and D. D’Souza, “Fault Diagnosis Using Timed Automata,” in *Foundations of Software Science and Computational Structures*, vol. 3441, V. Sassone, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 219–233. doi: 10.1007/978-3-540-31982-5_14.

[27] F. G. Cabral, M. V. Moreira, O. Diene, and J. C. Basilio, “A Petri Net Diagnoser for Discrete Event Systems Modeled by Finite State Automata,” *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 59–71, Jan. 2015, doi: 10.1109/TAC.2014.2332238.

[28] A. Giua and C. Seatzu, “Observability of place/transition nets,” *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1424–1437, Sep. 2002, doi: 10.1109/TAC.2002.802769.

[29] X. Zhao, G. Lamperti, D. Ouyang, and X. Tong, “Minimal Diagnosis and Diagnosability of Discrete-Event Systems Modeled by Automata,” *Complexity*, vol. 2020, p. e4306261, Feb. 2020, doi: 10.1155/2020/4306261.

[30] M. P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu, “A New Approach for Diagnosability Analysis of Petri Nets Using Verifier Nets,” *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3104–3117, Dec. 2012, doi: 10.1109/TAC.2012.2200372.

[31] S. Hu, J. Zhang, and Z. Li, “Diagnosability Enforcement in Labeled Petri Nets Based on Digital Twins,” in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, May 2022, vol. 1, pp. 1279–1284. doi: 10.1109/CoDIT55151.2022.9804165.

[32] K. Zhang and J. Raisch, “Diagnosability of labeled weighted automata over the monoid ($\mathbb{Q}\geqslant 0,+,0$),” in *2021 60th IEEE Conference on Decision and Control (CDC)*, Dec. 2021, pp. 6867–6872. doi: 10.1109/CDC45484.2021.9683278.

[33] B. Li, M. Khlif-Bouassida, and A. Toguyéni, “Reduction Rules for Diagnosability Analysis of Complex Systems Modeled by Labeled Petri Nets,” *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 1061–1069, Apr. 2020, doi: 10.1109/TASE.2019.2933230.

[34] S. Genc and S. Lafortune, “Predictability of event occurrences in partially-observed discrete-event systems,” *Automatica*, vol. 45, no. 2, pp. 301–311, Feb. 2009, doi: 10.1016/j.automatica.2008.06.022.

[35] M. Msaaf and F. Belmajdoub, “Fault diagnosis and prognosis in discrete event systems using statistical model and neural networks,” *Int J Mechatron. Autom.*, vol. 6, no. 4, p. 10, 2018.

[36] R. Alur and T. A. Henzinger, “Back to the future: towards a theory of timed regular languages,” in *Proceedings., 33rd Annual Symposium on Foundations of Computer Science*, Oct. 1992, pp. 177–186. doi: 10.1109/SFCS.1992.267774.

[37] F. Cassez and A. Grastien, “Predictability of Event Occurrences in Timed Systems,” in *Formal Modeling and Analysis of Timed Systems*, Berlin, Heidelberg, 2013, pp. 62–76. doi: 10.1007/978-3-642-40229-6_5.

[38] M. FRI, “Elaboration d’un Diagnostiqueur pour les Systèmes à Evénements Discrets et Contribution à la Prise de Décision.”

[39] A. Allahham, “Surveillance des systèmes à événements discrets commandés: Conception et implémentation en utilisant l’automate programmable industriel,” *Phys. Phys. Univ. JosephFourier - Grenoble Fr. Fftel-00347788*, p. 199, 2008.

[40] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 3e édition. Boston: Pearson, 2006.

[41] *Les réseaux de Petri : Un outils de modélisation Cours et exercices corrigés de Annie Choquet-Geniet*.

[42] A. T. Y. Watanabe, R. Sebem, A. B. Leal, and M. da S. Hounsell, “Fault prognosis of discrete event systems: An overview,” *Annu. Rev. Control*, vol. 51, pp. 100–110, 2021, doi: 10.1016/j.arcontrol.2021.03.005.

[43] T. Ekanayake, D. Dewasurendra, S. Abeyratne, L. Ma, and P. Yarlagadda, “Model-based fault diagnosis and prognosis of dynamic systems: a review,” *Procedia Manuf.*, vol. 30, pp. 435–442, Jan. 2019, doi: 10.1016/j.promfg.2019.02.060.

[44] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, “Fault diagnosis in discrete-event systems: framework and model reduction,” in *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*, Dec. 1998, vol. 4, pp. 3769–3774 vol.4. doi: 10.1109/CDC.1998.761808.

[45] Y. Wu and C. N. Hadjicostis, “Algebraic approaches for fault identification in discrete-event systems,” *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2048–2055, Dec. 2005, doi: 10.1109/TAC.2005.860249.

[46] D. Lefebvre and C. Delherm, “Diagnosis of DES With Petri Net Models,” *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 114–118, Jan. 2007, doi: 10.1109/TASE.2006.872122.

[47] P. M. Merlin, “A Study of the Recoverability of Computing Systems,” Nov. 1974, Accessed: Jul. 06, 2022. [Online]. Available: https://escholarship.org/uc/item/1p80c4fg

[48] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, “Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems,” *Control Eng. Pract.*, vol. 19, no. 9, pp. 989–1001, Sep. 2011, doi: 10.1016/j.conengprac.2010.12.010.

[49] M. P. Cabasino, A. Giua, and C. Seatzu, “Fault detection for discrete event systems using Petri nets with unobservable transitions,” *Automatica*, vol. 46, no. 9, pp. 1531–1539, Sep. 2010, doi: 10.1016/j.automatica.2010.06.013.

[50] A. Giua and C. Seatzu, “Fault detection for discrete event systems using Petri nets with unobservable transitions,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec. 2005, pp. 6323–6328. doi: 10.1109/CDC.2005.1583175.

[51] K. René Boel and G. Jiroveanu, “Distributed contextual diagnosis for very large systems,” *IFAC Proc. Vol.*, vol. 37, no. 18, pp. 333–338, Sep. 2004, doi: 10.1016/S1474-6670(17)30768-1.

[52] S. Genc and S. Lafortune, “Predictability of event occurrences in partially-observed discrete-event systems,” *Automatica*, vol. 45, no. 2, pp. 301–311, Feb. 2009, doi: 10.1016/j.automatica.2008.06.022.

[53] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, “Faults prognosis using partially observed stochastic Petri nets,” in *2016 13th International Workshop on Discrete Event Systems (WODES)*, May 2016, pp. 472–477. doi: 10.1109/WODES.2016.7497890.

[54] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, “A Comparative Study of Fault Prognosis Approaches for Timed Stochastic Discrete Event Systems,” *IFAC-Pap.*, vol. 50, no. 1, pp. 6807–6812, Jul. 2017, doi: 10.1016/j.ifacol.2017.08.1199.

[55] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, “Fault prognosis of timed stochastic discrete event systems with bounded estimation error,” *Automatica*, vol. 82, pp. 35–41, Aug. 2017, doi: 10.1016/j.automatica.2017.04.028.

[56] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre, “State estimation of discrete event systems for RUL prediction issue,” *Int. J. Prod. Res.*, vol. 55, no. 23, pp. 7040–7057, Dec. 2017, doi: 10.1080/00207543.2017.1346835.

[57] R. Zhao, F. Liu, and J. Tan, “Relative predictability of failure event occurrences and its opacity-based test algorithm,” *Int. J. Control*, vol. 92, no. 7, pp. 1600–1608, Jul. 2019, doi: 10.1080/00207179.2017.1403049.

[58] Villemeur Alain, *Sûreté de fonctionnement des systèmes industriels: fiabilité, facteurs humains, informatisation / Alain Villemeur,... ; préfaces de Paul Caseau, Arnould d’Harcourt*. Paris: Eyrolles, 1988.

[59] F. Chollet, *Deep Learning with Python*, 1st edition. Shelter Island, New York: Manning, 2017.

[60] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd edition. Beijing China ; Sebastopol, CA: O’Reilly Media, 2019.

[61] E. Minca, D. Racoceanu, Fl. Dragomir, and N. Zerhouni, “A FUZZY APPROACH FOR DISCRETE EVENT SYSTEMS RECOVERY,” *IFAC Proc. Vol.*, vol. 40, no. 18, pp. 585–590, Sep. 2007, doi: 10.3182/20070927-4-RO-3905.00097.

[62] C. V. I. Narvaez, “Diagnostic par techniques d’apprentissage floues: concept d’une méthode de validation et d’optimisation des partitions,” *Autom. Robot. INSA Toulouse Fr. Fftel-00190884f*, p. 141, 2007.

[63] Z. Li, Y. Wang, and K.-S. Wang, “Intelligent predictive maintenance for fault diagnosis and prognosis in machine centers: Industry 4.0 scenario,” *Adv. Manuf.*, vol. 5, no. 4, pp. 377–387, Dec. 2017, doi: 10.1007/s40436-017-0203-8.

[64] *Industrie 4.0. Les défis de la transformation numérique du modèle industriel allemand - Dorothée Kohler,Jean-Daniel Weisz*. Accessed: Jul. 06, 2022. [Online]. Available: https://www.decitre.fr/livres/industrie-4-0-9782110102102.html

[65] M. Blanchet, “Industrie 4.0 : nouvelle donne industrielle, nouveau modèle économique,” *Géoéconomie*, vol. 82, no. 5, pp. 37–53, 2016, doi: 10.3917/geoec.082.0037.

[66] A. Moeuf, R. Pellerin, S. Lamouri, S. Tamayo-Giraldo, and R. Barbaray, “The industrial management of SMEs in the era of Industry 4.0,” *Int. J. Prod. Res.*, vol. 56, no. 3, pp. 1118–1136, Feb. 2018, doi: 10.1080/00207543.2017.1372647.

[67] J.-R. Ruiz-Sarmiento, J. Monroy, F.-A. Moreno, C. Galindo, J.-M. Bonelo, and J. Gonzalez-Jimenez, “A predictive model for the maintenance of industrial machinery in the context of industry 4.0,” *Eng. Appl. Artif. Intell.*, vol. 87, no. C, Jan. 2020, doi: 10.1016/j.engappai.2019.103289.

[68] P. Alexandre, “CONTRIBUTION AU DIAGNOSTIC DECENTRALISE DES SYSTEMES A EVENEMENTS DISCRETS: APPLICATION AUX SYSTEMES MANUFACTURIERS,” p. 212.

[69] M. M. Savino, A. Brun, and C. Riccio, “Integrated system for maintenance and safety management through FMECA principles and fuzzy inference engine,” *Eur. J Ind. Eng.*, vol. 5, no. 2, p. 132, 2011.

[70] A. Bouti and D. A. Kadi, “A state-of-the-art review of fmea/fmeca,” *Int. J. Reliab. Qual. Saf. Eng.*, vol. 01, no. 04, pp. 515–543, Dec. 1994, doi: 10.1142/S0218539394000362.

[71] A. Jamshidi and R. Kazemzadeh, “A Fuzzy Cost-based FMEA Model,” *undefined*, 2010, Accessed: Jul. 06, 2022. [Online]. Available: https://www.semanticscholar.org/paper/A-Fuzzy-Cost-based-FMEA-Model-Jamshidi-Kazemzadeh/37a1616832dd251121095adff2de4f01adc7ffad

[72] N. Xiao, H.-Z. Huang, Y. Li, L. He, and T. Jin, “Multiple failure modes analysis and weighted risk priority number evaluation in FMEA,” *Eng. Fail. Anal.*, vol. 18, no. 4, pp. 1162–1170, Jun. 2011, doi: 10.1016/j.engfailanal.2011.02.004.

[73] K. Pickard, P. Muller, and B. Bertsche, “Multiple failure mode and effects analysis-an approach to risk assessment of multiple failures with FMEA,” in *Annual Reliability and Maintainability Symposium, 2005. Proceedings.*, Jan. 2005, pp. 457–462. doi: 10.1109/RAMS.2005.1408405.

[74] Y. Papadopoulos, “Model-based system monitoring and diagnosis of failures using statecharts and fault trees,” *Reliab. Eng. Syst. Saf.*, vol. 81, no. 3, pp. 325–341, Sep. 2003, doi: 10.1016/S0951-8320(03)00095-4.

[75] “NIOSHTIC-2 Publications Search - 20000060 - System safety and risk management: NIOSH instructional module.” https://www.cdc.gov/niosh/nioshtic-2/20000060.html (accessed Jul. 06, 2022).

[76] C. A. E. II, *Hazard Analysis Techniques for System Safety*, 1st edition. Hoboken, NJ: Wiley-Interscience, 2005.

[77] T. Berredjem and M. Benidir, “Bearing faults diagnosis using fuzzy expert system relying on an Improved Range Overlaps and Similarity method,” *Expert Syst. Appl.*, vol. 108, pp. 134–142, Oct. 2018, doi: 10.1016/j.eswa.2018.04.025.

[78] T. Satyanarayana, G. Subramanyam, and K. V. Rama Rao, “Implementing an expert system for fault diagnosis of electronic equipment,” *Eng. Appl. Artif. Intell.*, vol. 8, no. 3, pp. 355–364, Jun. 1995, doi: 10.1016/0952-1976(95)00012-P.

[79] Q. B. Chou, J. Mylopoulos, and J. Opala, “Experiences with an expert system technology for real-time monitoring and diagnosis of industrial processes,” Dec. 1996, Accessed: Jul. 07, 2022. [Online]. Available: https://www.osti.gov/etdeweb/biblio/454182

[80] P. Ribot, “Vers l’intégration diagnostic/pronostic pour la maintenance des systèmes complexes,” *Autom. Robot. Univ. Paul Sabatier - Toulouse III Fr. Fftel-00450835ff*, p. 171, 2009.

[81] J. Maciej Kościelny and M. Syfert, “Fuzzy Logic Application to Diagnostics of Industrial Processes,” *IFAC Proc. Vol.*, vol. 36, no. 5, pp. 711–716, Jun. 2003, doi: 10.1016/S1474-6670(17)36576-X.

[82] A. Rodríguez Ramos *et al.*, “An approach to multiple fault diagnosis using fuzzy logic,” *J. Intell. Manuf.*, vol. 30, no. 1, pp. 429–439, Jan. 2019, doi: 10.1007/s10845-016-1256-4.

[83] S. S. Godil, M. S. Shamim, S. A. Enam, and U. Qidwai, “Fuzzy logic: A ‘simple’ solution for complexities in neurosciences?,” *Surg. Neurol. Int.*, vol. 2, p. 24, Feb. 2011, doi: 10.4103/2152-7806.77177.

[84] R. Zemouri, D. Racoceanu, and N. Zerhouni, “Réseaux de neurones récurrents à fonctions de base radiales : RRFR Application au pronostic,” *Rev. Intell. Artif.*, vol. 16, no. 3, pp. 307–338, Jun. 2002, doi: 10.3166/ria.16.307-338.

[85] B. Li, M.-Y. Chow, Y. Tipsuwan, and J. C. Hung, “Neural-network-based motor rolling bearing fault diagnosis,” *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 1060–1069, Oct. 2000, doi: 10.1109/41.873214.

[86] A. Jose, Ruz-Hernandez, D. A. Suarez, and E. N. Sanchez, “FAULT DIAGNOSIS FOR A STEAM GENERATOR VIA RECURRENT NEURAL NETWORKS,” *IFAC Proc. Vol.*, vol. 39, no. 13, pp. 663–668, Jan. 2006, doi: 10.3182/20060829-4-CN-2909.00110.

[87] M. Mijwil, “Artificial Neural Networks Advantages and Disadvantages,” *Extr. LinkedIn Httpswww Linkedin Compulseartificial-Neuralnet Works-Advant.-Disadv.-Maad-M-Mijwel*, Jan. 2018.

[88] *Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processes*. Accessed: Jul. 07, 2022. [Online]. Available: https://link.springer.com/book/10.1007/978-3-540-79872-9

[89] A. Maier, O. Niggemann, R. Just, M. Jäger, and A. Vodencarevic, “Anomaly Detection in Production Plants using Timed Automata - Automated Learning of Models from Observations,” in *ICINCO*, 2011. doi: 10.5220/0003538903630369.

[90] H. Alaoui Ismaili and F. Belmajdoub, “Étude comparative des approches de construction d’un module de diagnostic des Systèmes à évènements discrets,” in *Xème Conférence Internationale : Conception et Production Intégrées*, Tanger, Morocco, Dec. 2015. Accessed: Jul. 07, 2022. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01260829

[91] N. PALLUAT, D. RACOCEANU, and N. ZERHOUNI, “Utilisation des réseaux de neurones temporels pour le pronostic et la surveillance dynamique : Etude comparative de trois réseaux de neurones récurrents,” *Util. Réseaux Neurones Temporels Pour Pronost. Surveill. Dyn. Etude Comp. Trois Réseaux Neurones Récurrents*, vol. 19, no. 6, pp. 913–950, 2005.

[92] *9781973343288 - L'intelligence artificielle expliquée : Comment les algorithmes et le Deep Learning dominent le monde ! (French Edition) - Thomas Cambrai*. Accessed: Jul. 13, 2022. [Online]. Available: https://www.eurolivre.fr/livre/isbn/9781973343288.html

[93] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[94] R. Sathya and A. Abraham, "Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification," *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 2, 2013, doi: 10.14569/IJARAI.2013.020206.

[95] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and Tensorflow: Concepts, Tools, and Techniques to Build Intelligent Systems*, $ {number}nd édition. Beijing China ; Sebastopol, CA: O'Reilly Media, Inc, USA, 2019.

[96] A. Gulli and S. Pal, *Deep Learning with Keras: Implement Neural Networks with Keras on Theano and TensorFlow*. Packt Publishing, 2017.

[97] "Hebb, D. O. Organization of behavior. New York: Wiley, 1949, pp. 335, $4.00," *J. Clin. Psychol.*, vol. 6, no. 3, pp. 307–307, 1950, doi: 10.1002/1097-4679(195007)6:3<307::AID-JCLP2270060338>3.0.CO;2-K.

[98] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, pp. 386–408, 1958, doi: 10.1037/h0042519.

[99] B. Widrow and M. E. Hoff, "Adaptive Switching Circuits," *IRE 1960 IRE WESCON Conv. Rec.*, pp. 96–104, 1960.

[100] Y. Lecun, "Une procedure d'apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks)," *Proc. Cogn. 85 Paris Fr.*, pp. 599–604, 1985.

[101] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990, doi: 10.1109/5.58337.

[102] Rummelhart D.E, Hinton G.E., and Williams R.J, "Learning Internal Representation by Error Propagation," *Parallel Distrib. Process. Explor. Microstruct. Cogn. MIT Press. Books*, vol. 1, pp. 318–362, 1986.

[103] S. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4, pp. 185–196, Jun. 1993, doi: 10.1016/0925-2312(93)90006-O.

[104] J. Konečný, J. Liu, P. Richtárik, and M. Takáč, "Mini-Batch Semi-Stochastic Gradient Descent in the Proximal Setting," *IEEE J. Sel. Top. Signal Process.*, vol. 10, no. 2, pp. 242–255, Mar. 2016, doi: 10.1109/JSTSP.2015.2505682.

[105] G. Nakerst, J. Brennan, and M. Haque, "Gradient descent with momentum --- to accelerate or to super-accelerate?" arXiv, Jan. 17, 2020. doi: 10.48550/arXiv.2001.06472.

[106] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o(1/k^2)," *Undefined 1983 Pap.-Method--Unconstrained-Convex-Minimization--Nesteroved910d96802212c9e45d956adaa27d915f5d7469 Accessed Nov 13 2020*.

[107] M. Assran and M. Rabbat, "On the Convergence of Nesterov's Accelerated Gradient Method in Stochastic Settings." arXiv, Jun. 27, 2020. doi: 10.48550/arXiv.2002.12414.

[108] M. Riedmiller and H. Braun, "RPROP - A Fast Adaptive Learning Algorithm," *Proc ISCIS VII*, 1992.

[109] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: the RPROP algorithm,” in *IEEE International Conference on Neural Networks*, Mar. 1993, pp. 586–591 vol.1. doi: 10.1109/ICNN.1993.298623.

[110] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[111] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, “The Marginal Value of Adaptive Gradient Methods in Machine Learning.” arXiv, May 21, 2018. doi: 10.48550/arXiv.1705.08292.

[112] M. C. Mukkamala and M. Hein, “Variants of RMSProp and Adagrad with Logarithmic Regret Bounds,” *Precup Teh W Eds Proc. 34th Int. Conf. Mach. Learn. Vol. 70 Proc. Mach. Learn. Res. Pp 2545–2553 Int. Conv. Cent. Syd. Aust. 06–11 Aug 2017 PMLR*.

[113] Hinton, G., Srivastava, N., and Swersky, K., “Lecture 6d - a separate, adaptive learning rate for each connection. 2012.,” *Slides Lect. Neural Netw. Mach. Learn.*, 2012.

[114] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization.” arXiv, Jan. 29, 2017. doi: 10.48550/arXiv.1412.6980.

[115] Y. You, I. Gitman, and B. Ginsburg, “Large Batch Training of Convolutional Networks.” arXiv, Sep. 13, 2017. doi: 10.48550/arXiv.1708.03888.

[116] Y. You *et al.*, “Large Batch Optimization for Deep Learning: Training BERT in 76 minutes.” arXiv, Jan. 03, 2020. doi: 10.48550/arXiv.1904.00962.

[117] M. B. Dahhou, M. K. Abderrahim, M. E. Busvelle, M. M. Chtourou, and M. A. Toumi, “Diagnostic, reconstruction et identification des défauts capteurs et actionneurs : appli- cation aux station d’épurations des eaux usées,” *Autre Univ. Toulon Éc. Natl. D’ingénieurs Sfax Tunis. Fr. FfNNT 2012TOUL0016ff Fftel-00843868f*, p. 201, 2012.

[118] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, “Phoneme recognition using time-delay neural networks,” *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 3, pp. 328–339, Mar. 1989, doi: 10.1109/29.21701.

[119] T. J. Sejnowski and C. R. Rosenberg, “NETtalk: A Parallel Network That Learns to Read Aloud,” *Complex Syst*, vol. 1, pp. 145–168, 1987.

[120] Faris, H., Aljarah, I., Mirjalili, S., “Evolving radial basis function networks using moth–flame optimizer.,” *Handb. Neural Comput.*, pp. 537–550, 2017.

[121] Zemouri, R., “Contribution à la surveillance des systèmes de production à l’aide des réseaux de neurones dynamiques: Application à la e-maintenance (Doctoral dissertation, Université de Franche-Comté).,” *Autom. Robot. Univ. Franche-Comté Fr. Fftel-00006003ff*, 2003.

[122] J. Chen and R. Kumar, “Stochastic Failure Prognosability of Discrete Event Systems,” *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1570–1581, Jun. 2015, doi: 10.1109/TAC.2014.2381437.

[123] J. Moody and C. J. Darken, “Fast Learning in Networks of Locally-Tuned Processing Units,” *Neural Comput.*, vol. 1, no. 2, pp. 281–294, Jun. 1989, doi: 10.1162/neco.1989.1.2.281.

[124] Broomhead, D.S., Lowe, D., “Radial basis functions, multi-variable functional interpolation and adaptive networks.,” *R. Signals Radar Establ. Malvern*, 1988.

[125] “https://machinelearningmastery.com/how-to-prepare-categorical-data-for-deep-learning-in-python.”

[126] Kopal, I., Harničárová, M., Valíček, J., Krmela, J., & Lukáč, O., "Radial basis function neural network-based modeling of the dynamic thermo-mechanical response and damping behavior of thermoplastic elastomer systems. Polymers," vol. 11, no. 6, p. 1074., 2019, doi: https://doi.org/10.3390/polym11061074.

[127] M.-F. Li, X.-P. Tang, W. Wu, and H.-B. Liu, "General models for estimating daily global solar radiation for different solar radiation zones in mainland China," *Energy Convers. Manag.*, vol. 70, pp. 139–148, Jun. 2013, doi: 10.1016/j.enconman.2013.03.004.

[128] N. Ben Rabah, R. Saddem, F. Ben Hmida, V. Carré-Ménétrier, and M. Tagina, "Automatic Acquisition and Update of a Causal Temporal Signatures Base- for Faults Diagnosis in Automated Production Systems," in *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, Madrid, Spain, 2017, pp. 262–269. doi: 10.5220/0006430102620269.

[129] R. Saddem and A. Philippot, "Causal Temporal Signature from diagnoser model for online diagnosis of Discrete Event Systems," in *2014 International Conference on Control, Decision and Information Technologies (CoDIT)*, Metz, France, Nov. 2014, pp. 551–556. doi: 10.1109/CoDIT.2014.6996954.

[130] R. Saddem, A. Toguyéni, et T. Moncef, "ALGORITHME D'INTERPRETATION D'UNE BASE DE SIGNATURES TEMPORELLES CAUSALES POUR LE DIAGNOSTIC EN LIGNE DES SYSTEMES A EVENEMENTS DISCRETS.," *9th Int. Conf. Model. Optim. Simul.*, Jun. 2012.

Résumé

Le développement et l'élaboration d'approches et d'outils de surveillance pour les systèmes industriels sont l'une des préoccupations majeures des industries, ainsi que des chercheurs dans le domaine. Ce type d'outil permet de superviser et de surveiller l'état de fonctionnement du système ainsi que de détecter les éventuels comportements anormaux et déviations. Les travaux présentés dans cette thèse s'inscrivent dans ce contexte. Nous nous intéressons à la surveillance (diagnostic et pronostic) des systèmes à événements discrets. Cependant, l'utilisation d'outils basés sur des modèles est devenue une tâche complexe et épuisante et parfois impossible, surtout lorsqu'il s'agit de grands systèmes industriels, alors que la complexité de ces systèmes n'a cessé d'augmenter et qu'ils ont récemment connu de grands progrès technologiques. Par conséquent, le développement de techniques plus sophistiquées est nécessaire. Dans cette thèse, nous nous intéressons principalement à l'utilisation et à l'exploitation des réseaux neuronaux récurrents, qui offrent un grand comportement dynamique. Il s'agit de mettre en œuvre des approches intelligentes, qui traitera du diagnostic et du pronostic des SEDs, de sorte que les événements générés seront présentés et analysés par les réseaux de neurones en temps réel, afin d'assurer un diagnostic et un pronostic en ligne.

Mots clés : Systèmes industriels, systèmes à événements discrets, surveillance, diagnostic, pronostic, réseaux de neurones.

Abstract

The development and elaboration of monitoring approaches and tools for industrial systems are one of the major concerns for industries, as well as researchers in the field. This type of tool allows to supervise and monitor the state of operation of the system as well as to detect the possible anomalous behaviors and deviations. The work presented in this thesis is part of this context. We are interested in the monitoring (Diagnosis and Prognosis) of Discrete Event Systems. However, the use of tools based on models has become a complex and exhausting task and sometimes impossible especially when it comes to large industrial systems, while the complexity of these systems has been increasing incessantly and have recently witnessed great technological progress. Therefore, the development of more sophisticated techniques is required. In this thesis, we are mainly interested in the use and the exploitation of recurrent neural networks, that provides a great dynamic behavior. To perform an intelligent approach, which will deal with the diagnosis and the prognosis of DES, so that the events generated, will be presented and analyzed by recurrent neural networks in real-time, to ensure an online diagnosis and prognosis.

Keywords: Industrial Systems, Discrete Event Systems, Monitoring, Diagnosis, Prognosis, Neural Networks.

Résumé de la thèse

Le développement et l'élaboration d'approches et d'outils de surveillance pour les systèmes industriels sont l'une des préoccupations majeures des industries, ainsi que des chercheurs dans le domaine. Ce type d'outil permet de superviser et de surveiller l'état de fonctionnement du système ainsi que de détecter les éventuels comportements anormaux et déviations. Les travaux présentés dans cette thèse s'inscrivent dans ce contexte. Nous nous intéressons à la surveillance (diagnostic et pronostic) des systèmes à événements discrets. Cependant, l'utilisation d'outils basés sur des modèles est devenue une tâche complexe et épuisante et parfois impossible, surtout lorsqu'il s'agit de grands systèmes industriels, alors que la complexité de ces systèmes n'a cessé d'augmenter et qu'ils ont récemment connu de grands progrès technologiques. Par conséquent, le développement de techniques plus sophistiquées est nécessaire. Dans cette thèse, nous nous intéressons principalement à l'utilisation et à l'exploitation des réseaux neuronaux récurrents, qui offrent un grand comportement dynamique. Il s'agit de mettre en œuvre des approches intelligentes, qui traitera du diagnostic et du pronostic des SEDs, de sorte que les événements générés seront présentés et analysés par les réseaux de neurones en temps réel, afin d'assurer un diagnostic et un pronostic en ligne.

Mots clés : Systèmes industriels, systèmes à événements discrets, surveillance, diagnostic, pronostic, réseaux de neurones.